STANDARDS RELATED DOCUMENT

ADatP-4778.2

PROFILES FOR BINDING METADATA TO A DATA OBJECT

Edition A Version 1

DECEMBER 2020

NORTH ATLANTIC TREATY ORGANIZATION

Published by the
NATO STANDARDIZATION OFFICE (NSO)
© NATO/OTAN

INTENTIONALLY BLANK

NORTH ATLANTIC TREATY ORGANIZATION (NATO)

NATO STANDARDIZATION OFFICE (NSO)

NATO LETTER OF PROMULGATION

2 December 2020

1. The enclosed Standards Related Document, ADatP-4778.2, Edition A, Version 1, PROFILES FOR BINDING METADATA TO A DATA OBJECT, which has been approved in conjunction with ADatP-4778 by the nations in the CONSULTATION, COMMAND AND CONTROL Board (C3B), is promulgated herewith.

2. ADatP-4778.2, Edition A, Version 1 is effective upon receipt.

3. This NATO standardization document is issued by NATO. In case of reproduction, NATO is to be acknowledged. NATO does not charge any fee for its standardization documents at any stage, which are not intended to be sold. They can be retrieved from the NATO Standardization Document Database ((https://nso.nato.int/nso/) or through your national standardization authorities.

4. This publication shall be handled in accordance with C-M(2002)60.

[illegible]

Gulyas
2 December 2020 at 13:51
Zoltán GULYAS
Brigadier General, HUNAF
Director, NATO Standardization Office

INTENTIONALLY BLANK

TABLE OF CONTENTS

INTENTIONALLY BLANK

CHAPTER 1 Introduction

1.1. Background

The Primary Directive on Information Management (PDIM) prescribes the application of metadata and markings in accordance with NATO policies and directives to facilitate sharing and control of NATO information.

The PDIM defines metadata as structured information that describes, explains, locates, and otherwise makes it easier to retrieve and use an information resource. The structure consists of ‘elements’, each of which will contain ‘values’. The values relate to the resource itself, there may be controls over what the actual values can be.

Metadata is a key enabler for the effective and efficient management of information. Modern automated information systems require information resources to be labelled with metadata.

1.2. Objective

The NATO Core Metadata Specification defines a set of core metadata elements to support information management in the Alliance.

This document recognizes the existence of communities of interest’s specific metadata standards and aims at steering their evolution in the mid to long term and at providing a single mediation standard in the short term to achieve sharing of information among different communities of interest.

1.3. Scope

NCMS applies to all NATO information and to any information resource handled or processed by NATO’s communications and information systems. NCMS describes information resource and supports its consistent and appropriate handling.

All NATO civil and military bodies are mandated to use NCMS.

Allies and Partners must also use NCMS when handling NATO information.

1.4. NATO Metadata Regulatory Standards

NATO has the following metadata standards:

- **ADatP-5636** NATO Core Metadata Specification defines the core set of metadata elements that must be used to support interoperable information exchange

- **ADatP-4774** Confidentiality Metadata Label Syntax provides support for the Security Layer metadata elements
- **ADatP-4778** Metadata Binding Mechanism describes how to consistently bind metadata (of any sort) to a finite data object

Standards-related Documents (SRDs) complement those three standards by providing implementation and other guidance.

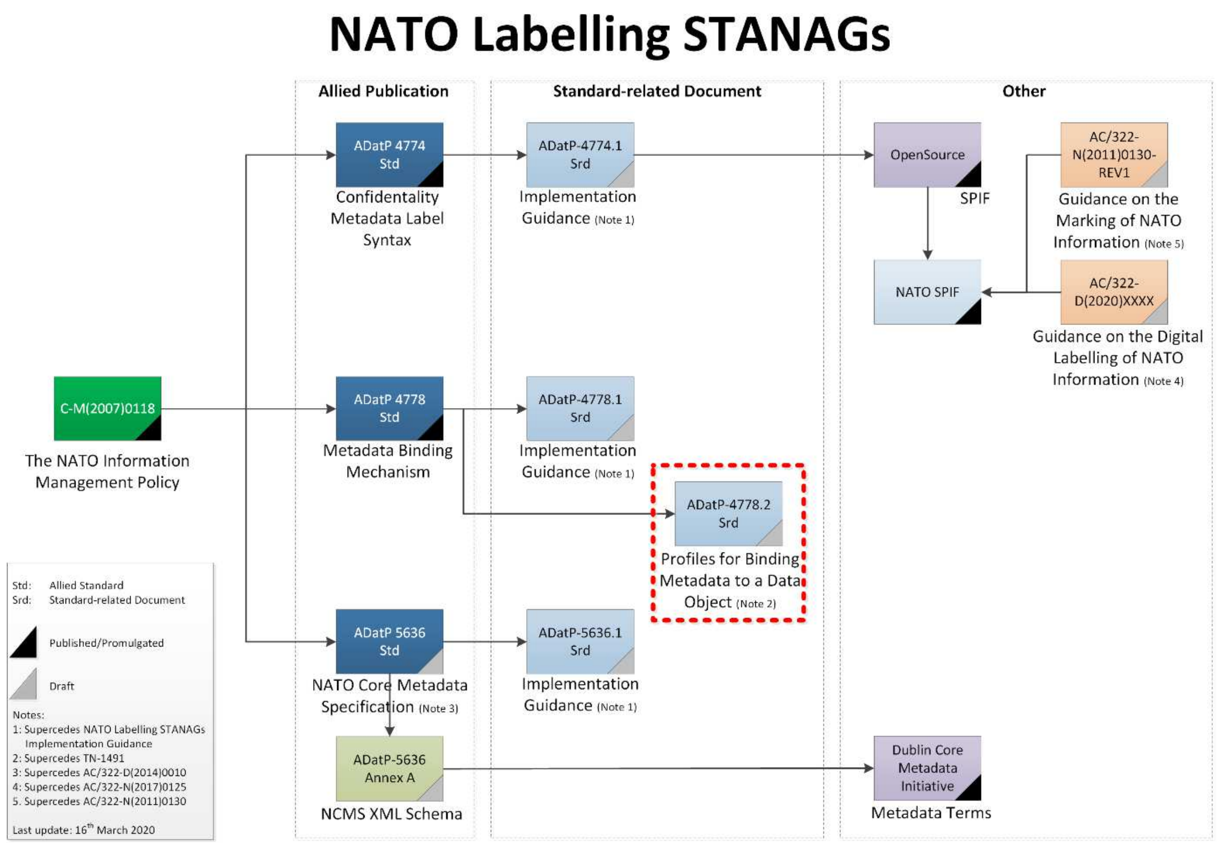

NATO Labelling STANAGs
Allied Publication
Standard-related Document
Other
ADatP 4774 Std
Confidentiality Metadata Label Syntax
ADatP-4774.1 Srd
Implementation Guidance (Note 1)
OpenSource
SPIF
NATO SPIF
AC/322-N(2011)0130-REV1
Guidance on the Marking of NATO Information (Note 5)
AC/322-D(2020)XXXX
Guidance on the Digital Labelling of NATO Information (Note 4)
C-M(2007)0118
The NATO Information Management Policy
ADatP 4778 Std
Metadata Binding Mechanism
ADatP-4778.1 Srd
Implementation Guidance (Note 1)
ADatP-4778.2 Srd
Profiles for Binding Metadata to a Data Object (Note 2)
ADatP 5636 Std
NATO Core Metadata Specification (Note 3)
ADatP-5636.1 Srd
Implementation Guidance (Note 1)
ADatP-5636 Annex A
NCMS XML Schema
Dublin Core Metadata Initiative
Metadata Terms
Std: Allied Standard
Srd: Standard-related Document
Published/Promulgated
Draft
Notes:
1: Supercedes NATO Labelling STANAGs Implementation Guidance
2: Supercedes TN-1491
3: Supercedes AC/322-D(2014)0010
4: Supercedes AC/322-N(2017)0125
5. Supercedes AC/322-N(2011)0130
Last update: 16th March 2020

Figure 1 NATO Labelling STANAGs

This document (SRD) is Profiles for Binding Metadata to a Data Object for the Metadata Binding Mechanism (highlighted in a red, dashed box in Figure 1).

1.5. Summary

ADatP-4778 - Metadata Binding Mechanism specifies a method for binding metadata information (including confidentiality metadata labels) to finite data objects.

There is a need for complementary Binding Profiles that define how metadata should be bound to specific data object types and where the resulting binding should be located with respect to the data object.

These Binding Profiles reduce the risks to capability procurement for common funded programmes in the NATO Enterprise by ensuring that all data objects of a given type and labelled in a consistent manner and that the metadata binding can be located.

This Standards related 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.

Additional Binding Profiles may be developed and supplement in future Editions and Versions of this Standards related document.

1.6. Overview

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 2) 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).

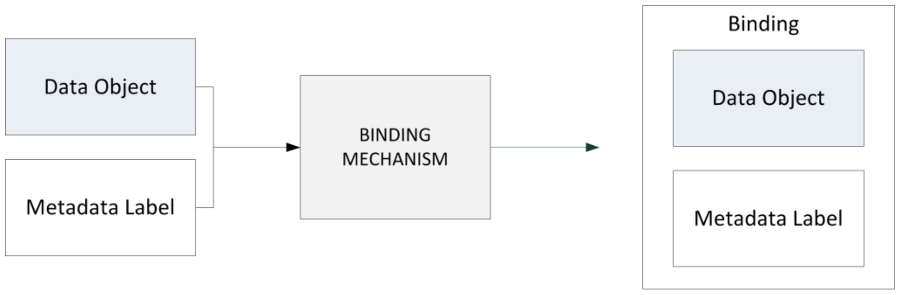


Figure 2 Creation of a binding

ADatP-4778 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 ADatP-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 ADatP-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 updates to this SRD, in accordance with AAP-03 (Reference [17]). 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, where applicable, 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. The examples provided to illustrate the semantics specified for each binding profile use Confidentiality Metadata Labels (Reference [1]) as example metadata with confidentiality metadata values specified for the AMOCO policy (Reference [22]).

1.7. Conformance And Interoperability

The profiles referenced in this document are methods of applying the binding mechanism stipulated in ADatP-4778. 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 ADatP-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 testing. Although a formal Reference facility for testing of external implementations of ADatP-4778 and these Binding Profiles is not yet established, a reference implementation for ADatP-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.

1.8 Configuration Management And Governance

Binding Profiles describe how to apply the binding mechanism specified in ADatP-4778 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) 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.

CHAPTER 2 Cryptographic Artefact Binding Profiles

2.1. Introduction

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]) and Cryptographic Message Syntax (CMS, Reference [18]) cryptographic protocols for generating a cryptographic artefact using digital signatures and / or key-hashed message authentication code (HMAC, Reference [7]) as the cryptographic techniques and mechanisms.

Table 2-1 below lists the supported cryptographic protocols and cryptographic mechanisms that are profiled for generating cryptographic artefacts.

Cryptographic Protocol	Cryptographic Mechanism	Reference
XML Signature (Reference [3[)	Digital Signature	ANNEX A and ANNEX B
	Keyed-Hash Message Authentication Code	ANNEX A and ANNEX C
CMS (Reference [18])	Digital Signature	ANNEX E

Table 2-1 Supported Cryptographic Protocols and Mechanisms Profiles

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

2.2. Identification

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

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:cryptoartefact
Version Identifier	urn:nato:stanag:4778:profile: cryptoartefact:1:2

Table 2-2 Profile Identifiers

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

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

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

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

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

2.3. Standards (Reference)

Reference [1] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
Reference [2] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium
Reference [3] W3C XMLDSIG-CORE, 2008, “XML- Signature Syntax and Processing (Second Edition)”, at http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/, W3C Recommendation, W3C, 10 June 2008
Reference [4] Web Services Security (WS-Security), SOAP Message Security 1 (WS-Security 2004), OASIS Standard Specification, 1 February 2006
Reference [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
Reference [6] W3C XPointer, 2002, “XML Pointer Language (XPointer)”, at http://www.w3.org/TR/xptr/, W3C Working Draft, W3C, 16 August 2002
Reference [7] IETF RFC 2104, “HMAC: Keyed-Hashing for Message Authentication”, at http://tools.ietf.org/html/rfc2104, February 1997

Reference [8] IETF RFC 8551, “Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification”, at http://tools.ietf.org/html/rfc8551, April 2019
Reference [9] IETF RFC 7515, “JSON Web Signature (JWS)”, at http://tools.ietf.org/html/rfc7515, May 2015
Reference [10] IETF RFC 6931, “Additional XML Security Uniform Resource Identifiers (URIs)”, at http://tools.ietf.org/html/rfc6931, April 2013
Reference [11] W3C XMLDSIG-2nd-Ed Errata, 2014, “Errata for XML Signature 2nd Edition”, at http://www.w3.org/2008/06/xmldsigcore-errata.html, W3C Recommendation, W3C, 01 October 2014
Reference [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.
Reference [13] W3C XMLDSIG-CORE1, 2013, “XML Signature Syntax and Processing Version 1.1”, at http://www.w3.org/TR/2013/REC-xmldsig-core1-20130411/, W3C Recommendation, W3C, 11 April 2013
Reference [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.
Reference [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.
Reference [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
Reference [17] AAP-03 “Directive for the Production, Maintenance and Management of NATO Standardization Documents”, Edition K, Version 1, February 2018.
Reference [18] IETF RFC 5652, “Cryptographic Message Syntax (CMS)”, at http://tools.ietf.org/html/rfc5652, September 2009
Reference [19] IETF RFC 8550, “Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Certificate Handling”, at http://tools.ietf.org/html/rfc8550, April 2019
Reference [20] IETF RFC 2634, “Enhanced Security Services for S/MIME”, at http://tools.ietf.org/html/rfc2634, June 1999
Reference [21] IETF RFC 3629, “UTF-8, a transformation format of ISO 10646”, at http://tools.ietf.org/html/rfc3629, November 2003
Reference [22] IETF RFC 3114, “Implementing Company Classification Policy with the S/MIME Security Label”, at http://tools.ietf.org/html/rfc3114, May 2002

ANNEX A Generic XML Signature Cryptographic Artefact Profile

A.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 Annex is read together with XMLDSIG.

Figure A-1 illustrates the structure of an XML Signature element including the primary sibling elements: SignedInfo; SignatureValue; KeyInfo; and, Object.

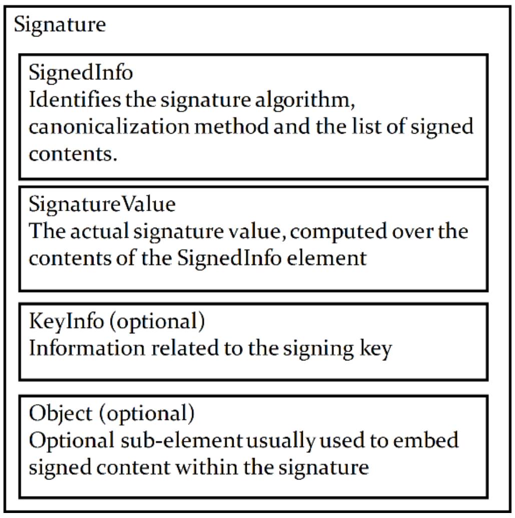

Signature
SignedInfo
Identifies the signature algorithm, canonicalization method and the list of signed contents.
SignatureValue
The actual signature value, computed over the contents of the SignedInfo element
KeyInfo (optional)
Information related to the signing key
Object (optional)
Optional sub-element usually used to embed signed content within the signature

Figure A-1 XML Signature Structure

This Annex 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 Annex will be divided into the following sub sections:

- General requirements for XMLDSIG including SignedInfo, SignatureValue and Object elements (refer to ANNEX A);
- Specific requirements for XMLDSIG SignedInfo and KeyInfo elements related to digital signatures (refer to ANNEX B); and,
- Specific requirements for XMLDSIG SignedInfo and KeyInfo elements related to keyed-hashed message authentication codes (refer to ANNEX C).

Example Binding Data Objects containing cryptographic artefacts conformant with this profile are illustrated in ANNEX D.

The notational conventions used for this Annex 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.

A.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 4.4.3.1, 4.4.3.2 and 4.4.3.3

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 4.4.3.3) the following requirements are REQUIRED to be followed:

- Originator MUST create a `Manifest` element for each *DataReference* or *MetadataReference* elements (that conforms to this use case) contained in the *bindingInformation* that includes a `Reference` element (as specified in `Manifest` section of ANNEX A);
- The `Manifest` element that the Originator creates MUST be stored as a child element of an Object element;
- Recipient MUST 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 ANNEX A).
        - 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.

XML Normalization

XML (de)serialization may result in a namespace prefix to be redefined within the XML document. XML documents that are provided as input to XML Signature Core Signature Generation and Verification may have differing information content, however, they are logically equivalent within a given application context. As a result the signature verification of the logically equivalent XML document will fail.

It is therefore RECOMMENDED that all XML documents prior to being provided as input to a XML Signature library for Core Signature Generation and Verification are passed through an XML normalization process that:

- Removes all namespace prefixes except “xml”;
- Visits each Element node in XML Document order;
- At each Element node all visibly utilised namespace URIs are considered;
- At each Element node duplicate namespace URIs are removed;
- At each Element node namespace URIs that have already been assigned are removed;
- At each Element node if an Attribute node of that Element node has a qualified name that is assigned a different namespace than the namespace of the Element node assign a prefix definition to the namespace of the Attribute node;
- Namespace declarations SHALL appear before attribute declarations;
- Attribute declarations are sorted lexicographically by namespaceURI as primary key and localName as secondary key;
- Attribute prefix definitions SHALL be written as “n0”, “n1”, “n2” ….etc. and,
- Preserve whitespace.

An XML Stylesheet (XSLT) 1.0 transform that performs XML normalization as described above is published in the NATO Metadata Registry and Repository (NMRR) at:

- https://nmrr.ncia.nato.int/rest/doc/NATO/Information%20Assurance/OLP/XML_Normalisation_1.0.xsl

It is RECOMMENDED that the XML Signature library when performing Core Signature Generation does not use a namespace prefix for the `<Signature/>` element and preserves whitespace when creating the XML Document containing the `<Signature/>` element.

URI Schemes

XML Signature Core Signature Generation and Verification may have differing levels of support to dereference specified URIs based on the URI scheme contained within a `Reference` element URI attribute value. In the use case whereby a URI scheme is used within a `Reference` element URI attribute value that may not be supported by XML Signature Core Signature Generation and Verification implementations the following requirements are REQUIRED to be followed:

- Originator MUST create a `Manifest` element for each *DataReference* or *MetadataReference* elements (that conforms to this use case) contained in the *bindingInformation* that includes a `Reference` element (as specified in `Manifest` section of ANNEX A);
- The `Manifest` element that the Originator creates MUST be stored as a child element of an `Object` element;
- Recipient MUST 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 scheme);
        - Digest the resulting data object using the `DigestMethod` (as specified in the Reference section in ANNEX A).
        - Compare the generated digest value against `DigestValue` in the Manifest Reference; if there is any mismatch, validation fails.

Core Signature Syntax

This section refers to XMLDSIG Section 4.

Signature

This section refers to XMLDSIG Section 4.2.

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 4.3.

SignedInfo

This section refers to XMLDSIG Section 4.4.

CanonicalizationMethod

This section refers to XMLDSIG Section 4.4.1.

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 4.4.2.

The `SignatureMethod` Algorithm attribute is REQUIRED.

The value of the `SignatureMethod` Algorithm is further specified depending on the cryptographic technique and mechanism being used (refer to ANNEX B for Digital Signatures or ANNEX C for HMAC).

Reference

This section refers to XMLDSIG Section 4.4.3.

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 4.4.3.1.

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 4.4.3.4.

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 4.4.3.5.

The `DigestMethod` Algorithm attribute MUST conform to the specifications detailed in Table 2-3.

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	Prohibited
http://www.w3.org/2001/04/xmlenc#sha256	Optional
http://www.w3.org/2001/04/xmldsig-more#sha384	Mandatory
http://www.w3.org/2001/04/xmlenc#sha512	Optional
http://www.w3.org/2001/04/xmlenc#ripemd160	Optional

Table 2-3 DigestMethod Algorithm Identifiers

DigestValue

This section refers to XMLDSIG Section 4.4.3.6.

KeyInfo

This section refers to XMLDSIG Section 4.5.

The `KeyInfo` element is REQUIRED.

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

Object

This section refers to XMLDSIG Section 4.6.

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 to support the use case for: Same-Document URI-References; and, URI Schemes not supported by XML Signature Core Signature Generation and Verification implementations.

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); or, the URI scheme.

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 the originator MUST 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.

ANNEX B XML Signature: Digital Signature Cryptographic Artefact

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

SignedInfo

This section refers to XMLDSIG Section 4.4.

SignatureMethod

This section refers to XMLDSIG Section 4.4.2.

The `SignatureMethod` Algorithm attribute MUST conform to the specifications detailed in Table 2-4.

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

Table 2-4 SignatureMethod (PKI) Algorithm Identifiers

KeyInfo

This section refers to XMLDSIG Section 4.5.

The `KeyInfo` element is REQUIRED.

KeyName

This section refers to XMLDSIG Section 4.5.1.

The `KeyName` element SHALL NOT be present.

KeyValue

This section refers to XMLDSIG Section 4.5.2.

The `KeyValue` MAY be present.

RetrievalMethod

This section refers to XMLDSIG Section 4.5.3.

The `RetrievalMethod` SHALL NOT be present.

X509Data

This section refers to XMLDSIG Section 4.5.4.

The `X509Data` element is REQUIRED.

In strategic systems with high throughput, certificates MUST be included.
X.509 version 3 certificates (Reference [16]) 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 4.5.5.

The `PGPData` element SHALL NOT be present.

SPKIData

This section refers to XMLDSIG Section 4.5.6.

The `SPKIData` element SHALL NOT be present.

MgmtData

This section refers to XMLDSIG Section 4.5.7.

The `MgmtData` element SHALL NOT be present.

ANNEX C XML Signature: 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 ANNEX A and this Annex.

SignedInfo

This section refers to XMLDSIG Section 4.4.

SignatureMethod

This section refers to XMLDSIG Section 4.4.2.

The `SignatureMethod` Algorithm attribute MUST conform to the specifications detailed in Table 2-5.

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

Table 2-5 SignatureMethod (HMAC) Algorithm Identifiers

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 4.5.

The `KeyInfo` element is REQUIRED.

KeyName

This section refers to XMLDSIG Section 4.5.1.

The `KeyName` element MAY be present.

KeyValue

This section refers to XMLDSIG Section 4.5.2.

The `KeyValue` SHALL NOT be present.

RetrievalMethod

This section refers to XMLDSIG Section 4.5.3.

The `RetrievalMethod` SHALL NOT be present.

X509Data

This section refers to XMLDSIG Section 4.5.4.

The `X509Data` SHALL NOT be present.

PGPData

This section refers to XMLDSIG Section 4.5.5.

The `PGPData` element SHALL NOT be present.

SPKIData

This section refers to XMLDSIG Section 4.5.6.

The `SPKIData` element SHALL NOT be present.

MgmtData

This section refers to XMLDSIG Section 4.5.7.

The `MgmtData` element SHALL NOT be present.

ANNEX D Example XML Signature Cryptographic Bindings

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

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

```
<mb:BindingInformation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
  <Signature Id="id-a99fac99-513d-4b08-8158-ef862e4d9f80"
xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#" />
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#hmac-sha256" />
      <Reference URI="#id-66bb29e1-9696-4ea0-be3c-f7d0096a0d81">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />

<DigestValue>9JBAVs2gUWUzFh8uUl1ubXW13VgQxli3NM+CF0vQG14=</DigestValue>
      </Reference>
      <Reference URI="#id-d55d0123-babc-467f-b309-62e95291a9e4">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />

<DigestValue>8G8AHBPiAJ+W6PUOq+W/Vua+iO7Zj6GzooPRmkqtqnY=</DigestValue>
      </Reference>
      <Reference URI="#id-b3eaf318-700f-4740-b43e-2def8d98db81">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />

<DigestValue>Kx02/WnFE/2MN7lEuWemAiDetsJZ+8lJt4nvg4GyRNc=</DigestValue>
      </Reference>
    </SignedInfo>

<SignatureValue>g3nzbBiu7msmVHfCjmVqqSiimlASoBSM/hxqFN7YxH0=</SignatureValue>
    <KeyInfo Id="id-b3eaf318-700f-4740-b43e-2def8d98db81">
      <KeyName>HMAC_SECRET_KEY</KeyName>
    </KeyInfo>
    <Object Id="id-17250b2d-f0f5-4457-9e21-23db31e3460d">
      <SignatureProperties Id="id-d55d0123-babc-467f-b309-62e95291a9e4">
        <SignatureProperty>
          <wsu:TimeStamp xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
            <wsu:Created>2015-11-13T15:58:44Z</wsu:Created>
          </wsu:TimeStamp>
        </SignatureProperty>
      </SignatureProperties>
    </Object>
  </Signature>
  <mb:MetadataBindingContainer>
    <mb:MetadataBinding 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>TEST Amoco</slab:PolicyIdentifier>
            <slab:Classification>GENERAL</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 2-2 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>fAXcjRa4zlLyB+lchyBK/9JzlsoZSbxNCmr/27nA9aI=</DigestValue>
      </Reference>
      <Reference URI="#id-82744679-a547-40aa-a683-cf97619054fe">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />

<DigestValue>j5AgAamc6cv54VDzl0kDlQ4wYZLLAU3761eFOUWvtX0=</DigestValue>
      </Reference>
      <Reference URI="#id-9920a48c-c3a1-45d0-a81c-1ce04d1d8de6">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />

<DigestValue>hWUoi0gFxnFsGnHJO/V2eNg/silda814PSP2/WlsqtU=</DigestValue>
      </Reference>
    </SignedInfo>

<SignatureValue>gItAuwdEykw5xDht50TOei1xfT0q7KLaUXm4w/2rnpTjoxiODTI3Wr8D4fmx/
404bVrX23StY6HHT/dxDPcgODa+K9YL/pl3y8RvIrfWGhiZReY5AUj1EF3mxI22ari/ao0shKe18a
PJ0J2RmGH3t30qrHfvUXcIcREIOT1S6GajpNCOJPYoa9yb400MOx0oRHXkFegnQ5eXeSBIh2u4Dhw
L0I4GSeuYA9FVt8qyvla9EnTTS6fG2+gLjd6YEQzfIBvVtrY5b9WnhqqiHy5tyepZgVtMSEXrukWr
NELpvwC467KR+MincgUA9RlsAEvCBaR4oQKTUOxBQ5tD+N/FzQ==</SignatureValue>
    <KeyInfo Id="id-9920a48c-c3a1-45d0-a81c-1ce04d1d8de6">
      <X509Data>

<X509Certificate>MIIDM……wIBAgIJAI29/+A/MN7RPAx5eOKQg==</X509Certificate>
      </X509Data>
    </KeyInfo>
    <Object Id="id-63fc02c0-10b6-49fd-9759-7bfb1d52ecf7">
      <SignatureProperties Id="id-82744679-a547-40aa-a683-cf97619054fe">
        <SignatureProperty>
          <wsu:TimeStamp xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
            <wsu:Created>2015-11-13T16:01:38Z</wsu:Created>
          </wsu:TimeStamp>
        </SignatureProperty>
        </SignatureProperties>
      </Object>
    </Signature>
    <mb:MetadataBindingContainer>
      <mb:MetadataBinding 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>TEST Amoco</slab:PolicyIdentifier>
              <slab:Classification>GENERAL</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 2-3 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" />
        <Reference URI="">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-
19991116">
              <XPath>not(ancestor-or-self::*[local-name() =
'BindingInformation' and namespace-uri() =
'http://www.nato.int/2014/06/nl/mb'])</XPath>
            </Transform>
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"
/>

<DigestValue>RYxJZ8BN/MR2D0BDxiCxGSDaQvGFKQ86udb0Ov5A2s4=</DigestValue>
        </Reference>
        <Reference URI="#id-c9e4f1c8-ad34-4d4d-9909-827570de41a2">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"
/>

<DigestValue>WKOWdda84YLuSqbaZsS8LQ6kqF6HR0dfC+iz/e+KPf0=</DigestValue>
        </Reference>
        <Reference URI="#id-1bd95780-277f-44b3-99fd-b2b69505ae5a">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"
/>

<DigestValue>UbMTebL9lKFARnG1qWOpQ1DiuCFPzs6W1hse9gPOxUk=</DigestValue>
        </Reference>
        <Reference URI="#id-001c1a07-74c4-4815-aecd-dd1bcba8bc9c">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"
/>
```

```
<DigestValue>z7+6QZiSTqYMHCiy9o3uxGfA8q5ScEeHlHZs3w9+8S4=</DigestValue>
        </Reference>
      </SignedInfo>

<SignatureValue>dk7Ds4Atik6yF/wKZjOIDVGGyv1rigTDLj6gRsqCTHY=</SignatureValue>
      <KeyInfo Id="id-001c1a07-74c4-4815-aecd-dd1bcba8bc9c">
        <KeyName>HMAC_SECRET_KEY</KeyName>
      </KeyInfo>
      <Object Id="id-4dcc6c48-6ed0-4cf0-b386-b85f7ee0c826">
        <SignatureProperties Id="id-1bd95780-277f-44b3-99fd-b2b69505ae5a">
          <SignatureProperty>
            <wsu:TimeStamp xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
              <wsu:Created>2015-11-13T16:07:37Z</wsu:Created>
            </wsu:TimeStamp>
          </SignatureProperty>
        </SignatureProperties>
      </Object>
    </Signature>
    <mb:MetadataBindingContainer>
      <mb:MetadataBinding 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>TEST Amoco</slab:PolicyIdentifier>
              <slab:Classification>GENERAL</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>TEST Amoco</slab:PolicyIdentifier>
              <slab:Classification>GENERAL</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>
  </mb:BindingInformation>
 </Document>
```

Figure 2-4 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>0JsT5SNKuCYoe91tl8n590Hcy/UivrId3Zf6kJy7pdg=</DigestValue>
        </Reference>
        <Reference URI="#id-b073db91-a8b3-4905-809d-82e92b0d0ecc">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"
/>

<DigestValue>a3yUgG8j0eIPI6ZSw7aw4JPHO1SBglS0+Fb7lwVmMeo=</DigestValue>
        </Reference>
        <Reference URI="#id-7d5d0648-59c1-48a9-a3bc-a07a24f0a67b">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"
/>

<DigestValue>zMHgHTwG+OtqPY8+T4cwYGby2UoSv71QJ2eU0peB5ds=</DigestValue>
        </Reference>
        <Reference URI="#id-45f67abd-5803-4933-acb8-5061adde54f4">
          <Transforms>
```

```
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"
/>

<DigestValue>g8jESHIgXr4bGZFwOzh2O4r8Vv0y6jfH7qKgQTGV9ww=</DigestValue>
        </Reference>
      </SignedInfo>

<SignatureValue>C1yPwzpU/ngO42sXo2HHZTtbXNTe2FAXf2RivMy5u6z/xoNlmi/mHm5ejZPFW
koGaUmWDadREcc5lI6XBYXeks2YVyMh05uDRCQLPYNkIAx3BpUFH7y9JUklj4WvlDBeZ2GwNhp463
QMvn8pF35cXw1f86Vc0M3CtAm5MNbnS6BqqswdygCF/HivjHcQSnYGRhI4vegelwfYyhFRHQQ1OE3
ytUDR8VLKZfgyK3M6mcQjvlHtL2qjRxMHrkQQtt8oBQk6iAWxYgbqeIzqw3cIYL5jb/ML2UOycGgw
UIqGFx95EouKuOMZSN8e2dnaVaHp26XlzpdJkyTkVr5/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>TEST Amoco</slab:PolicyIdentifier>
              <slab:Classification>GENERAL</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>TEST Amoco</slab:PolicyIdentifier>
              <slab:Classification>GENERAL</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 2-5 : Embedded Cryptographic BDO Containing a Detached Signature with a Digital Signature Cryptographic Artefact

ANNEX E Generic CMS Cryptographic Artefact Profile

E.1. Introduction

The S/MIME protocol Cryptographic Message Syntax (CMS) (Reference [18]) leaves the implementers with a number of options that need to be agreed on in order to achieve interoperability. This Annex is a profile for the use of the CMS to facilitate the cryptographic protection of the integrity and authentication of a metadata binding, and describes which of the different elements of service that need to be present on origination and reception, in order to claim conformance to this profile.

In order to highlight the differences and avoid duplication of text from CMS, a delta specification approach has been taken. This Annex will refer to the relevant sections of CMS 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 Annex is read together with CMS.

Additional CMS elements of service may be required to support security services, such as message authentication, confidentiality, integrity and non-repudiation. This Annex does not profile the additional security services. Unless exceptions are noted, all statements that apply to CMS and additional security services profiles (that are required to be supported) also apply to this profile.

The notational conventions used for this Annex 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 STANAG 4778 (Reference [2]) referenced in this profile.
- `Courier font` indicates syntax derived from the CMS and S/MIME Specifications (References [8], [16], [18], [19] and [20]) referenced in this profile.

E.2. General Requirements

Unless otherwise stated, all statements that apply to Cryptographic Message Syntax (CMS, Reference [3]) also apply to this profile.

All of the mandatory elements of service (NOT explicitly marked as OPTIONAL), SHALL be supported by implementations which claim conformance to this profile. All of the elements of service that are mandatory to be generated on origination are also mandatory to be processed on reception.

An entity that creates CMS metadata bindings conformant with this profile (known as Originator) is REQUIRED to perform the rules for Message Digest Calculation Process and Signature Generation Process as specified in CMS Section 5.4 and 5.5. An entity that interprets and processes CMS metadata bindings conformant with this profile (known as Recipient) is REQUIRED to perform the rules for Message Digest Calculation Process and Signature Verification Process as specified in CMS Section 5.4 and 5.6.

E.3. CMS Profile

This section refers to section 5 in CMS.

General Syntax

`ContentInfo` SHALL be supported to encapsulate the `SignedData` in accordance with CMS. Conventions for inner wrappers SHALL comply with either Secure/Multipurpose Internet Mail Extensions (S/MIME, Reference [8]) depending on the type of content conveyed.

The `contentType` field SHALL be supported.

The `content` field SHALL be supported.

Data Content Type

This section refers to section 4 in CMS.

Conventions for inner wrappers SHALL comply with S/MIME depending on the type of content conveyed.

Signed-data Content Type

This section refers to section 5 in CMS.

Conventions for inner wrappers SHALL comply with S/MIME depending on the type of content conveyed.

SignedData Type

This section refers to section 5.1 in CMS.

In strategic systems with high throughput, the certificates field SHALL be included. X.509 version 3 certificates SHALL be supported.

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

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

There may be circumstances when the certificates SHOULD NOT be included, e.g. in tactical systems with low bandwidth.

The crls field is NOT REQUIRED.

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

The digestAlgorithms field SHALL contain DigestAlgorithmIdentifiers that conform to the specifications detailed in Table *2-6*.

Algorithm Identifier	**Mandatory/Optional/ Prohibited**
id-sha224 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) 4 }	Prohibited
id-sha256 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) 1 }	Mandatory
id-sha384 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) 2 }	Optional
id-sha512 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) 3 }	Optional
id-shake256 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) 2 12 }	Optional

Table 2-6: CMS Message Digest Algorithms

The use of the mandatory and optional DigestAlgorithmIdentifiers specified in this profile MAY be further specified dependent upon national or organizational policy and agreed between implementation communities.

EncapsulatedContentInfo Type

This section refers to section 5.2 in CMS.

The `eContentType` field SHALL be supported.

The `eContentType` SHALL be set to the object identifier of the object to be signed: `id-data`.

The use of the `eContent` field SHALL be supported depending upon the signed-only format as specified in S/MIME.

SignerInfo Type

This section refers to section 5.3 in CMS.

Originators and the Recipients SHOULD be able to handle multiple instances of `SignerInfo`.

The `SignerIdentifier` `issuerAndSerialNumber` field SHALL be supported.

The `SignerIdentifier` `subjectKeyIdentifier` field SHALL be supported.

The `digestAlgorithm` SHALL be supported as specified in Table *2-6*.

The `digestAlgorithm` SHALL be among those listed in the `digestAlgorithms` field of the associated `SignedData`.

The `signatureAlgorithm` field SHALL conform to the specifications detailed in Table *2-7*.

Algorithm Identifier	Mandatory/Optional/ Prohibited
id-dsa-with-sha224 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-dsa-with-sha2(3) 1 }	Optional
id-dsa-with-sha256 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-dsa-with-sha2(3) 2 }	Mandatory
sha224WithRSAEncryption OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 14 }	Optional
sha256WithRSAEncryption OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 11 }	Mandatory
sha384WithRSAEncryption OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 12 }	Optional
sha512WithRSAEncryption OBJECT IDENTIFIER ::= { iso(1)	Optional

member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 13 }	
ecdsa-with-SHA224 OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) 1 }	Optional
ecdsa-with-SHA256 OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) 2 }	Optional
ecdsa-with-SHA384 OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) 3 }	Optional
ecdsa-with-SHA512 OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) 4 }	Optional
id-RSASSA-PSS-SHAKE256 OBJECT IDENTIFIER ::= { iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) algorithms(6) 31 }	Optional
id-ecdsa-with-shake256 OBJECT IDENTIFIER ::= { iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) algorithms(6) 33 }	Optional

Table 2-7: CMS Signature Algorithms

The use of the mandatory and optional `signatureAlgorithm` specified in this profile MAY be further specified dependent upon national or organizational policy and agreed between implementation communities.

The `signedAttrs` filed SHALL be supported (see Section **Signed Attributes**).

Signed Attributes

The `SignerInfo` type allows unsigned and signed attributes to be included along with a signature.

A Recipient is REQUIRED to support all signed attributes on reception for the purpose of validating the signature value.

Requirements for processing of the attributes specified in this profile SHALL be adhered to.

No processing of the internal structure or semantics of any other attribute (not specified in this profile), or any of its sub-elements is required unless a specific claim of conformance is made to support the attribute type.

Additional attributes and values for these attributes may be defined in the future. A Recipient SHOULD handle attributes or values that it does not recognise in a graceful manner.

The `contentType` attribute SHALL be supported, as specified in CMS. Its value specifies the content type of the `contentInfo` being signed.

The `messageDigest` attribute SHALL be supported, as specified in CMS. The hash value received in this attribute SHALL NOT be used for signature validation (i.e. it SHALL be recalculated).

The `signingTime` attribute SHALL be supported.

The `eSSSecurityLabel` attribute, specified in Enhanced Security Services for S/MIME (ESS, Reference [20]), SHALL NOT be supported.

The `equivalentLabels` attribute , specified in ESS SHALL NOT be supported.

The bindingData attribute, specified in this profile, SHALL be supported (see Section Binding Information below).

E.4. Binding Information

The bindingData attribute type specifies the information required from the Binding Data Object that is required to be cryptographically protected.

The bindingData attribute type SHALL be present in `signed-data`.

The bindingData attribute SHALL be a signed attribute; it SHALL NOT be an unsigned attribute.

The following object identifier identifies the bindingData attribute:

```
id-bindingData OBJECT IDENTIFIER ::= { iso(1)
        identified-organization(3) nato(26) stanags(0)
        Generic Binding Mechanism(4778) infosec(1) 1 }
```

bindingData attribute values have ASN.1 type BindingData:

```
BindingData ::= SET {
      bindingType BindingType,
      bindingId BindingIdentifier,
      bindingDataInfos BindingDataInfos }

BindingType  ::= UTF8String
        -- The value from the Binding-Data header field binding-type parameter.
```

BindingIdentifier ::= UTF8String
    -- The identifier value from the *BindingInformation SignatureReference @URI* attribute.

BindingDataInfos ::= SEQUENCE SIZE (1..MAX) OF BindingDataInfo

BindingDataInfo ::= SEQUENCE {
    -- Per-*MetadataBinding* information is represented in this type.
  mbId MetadataBindingIdentifier,
  algId DigestAlgorithmIdentifier,
    -- Import DigestAlgorithmIdentifier from Reference [18]
    -- algId identifies the digest algorithm, and any associated parameters, under which the *MetadataBinding* is digested. The algId SHALL match the digest algorithm for the `SignerInfo` in which this bindingData attribute value is placed.
  metadataBindingDigest MetadataBindingDigest }

MetadataBindingIdentifier ::= UTF8String
    -- The value from the *@Id* attribute of the *MetadataBinding*.

MetadataBindingDigest ::= OCTET STRING
    -- The result of digesting the MetadataBinding associated with the MetadataBindingIdentifier.

UTF8String ::= [UNIVERSAL 12] IMPLICIT OCTET STRING
    -- The content of this type conforms to Reference [21].

A bindingData attribute SHALL have a single attribute value, even though the syntax is defined as a SET OF `AttributeValue`. There SHALL NOT be zero or multiple instances of `AttributeValue` present.

The `SignedAttributes` syntax is defined as a SET OF `Attributes`. The `SignedAttributes` in a `signerInfo` SHALL include only one instance of the bindingData attribute.

Generate bindingData

For each *MetadataBinding* element included in the *bindingInformation* element there SHALL be a *URI* attribute with a shortname XPointer as the attribute value that identifies each *MetadataBinding* element.

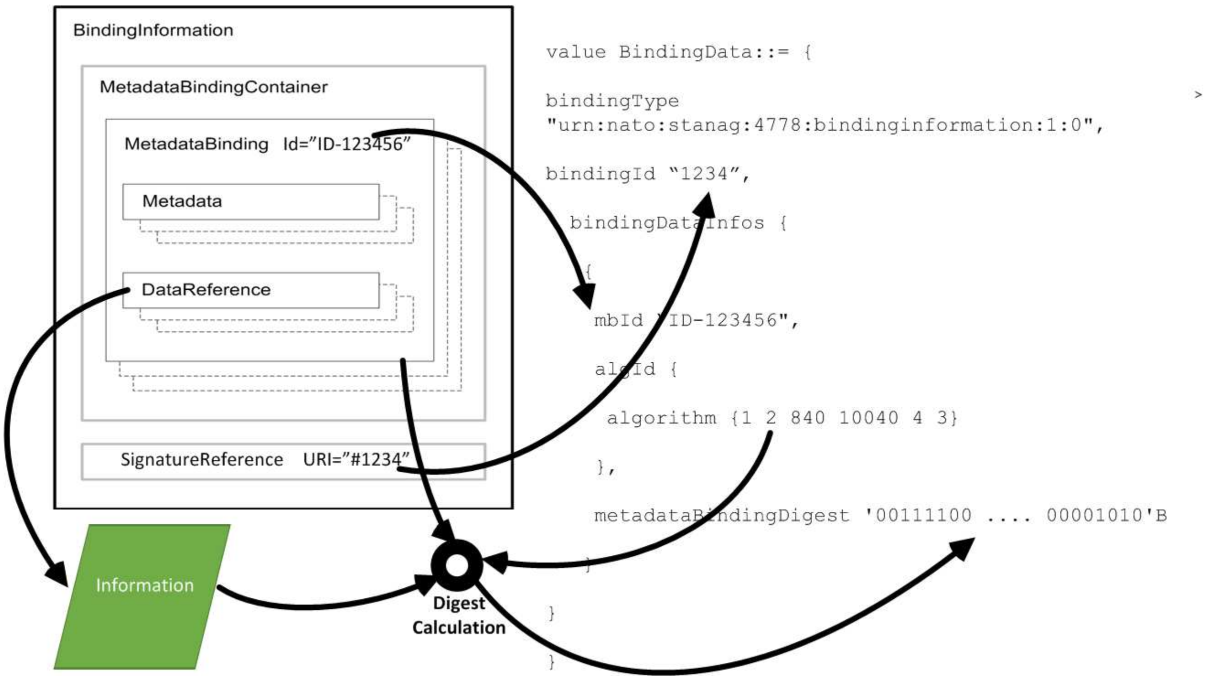
The bindingType SHALL be *urn:nato:stanag:4778:bindinginformation:1:0*.

The bindingId SHALL be the identifier of the *@URI* attribute value of the *SignatureReference* element contained in the BDO.

For each *MetadataBinding* element from the *bindingInformation*:

1) Create a BindingDataInfo;
2) Record the *@Id* attribute value in the MetadataBindingIdentifier;
3) Record the `SignerInfo digestAlgorithm` in the algId.
4) Normalise the *MetadataBinding* element conformant with the XML Normalization process specified in Annex A of this document;
5) The output from the normalised *MetadataBinding* element SHALL be passed into the digest calculation along with the algId digest algorithm, and any associated parameters; and,
6) Record the result of the digest calculation in the metadataBindingDigest.

Figure 2-6 bellows illustrates the relationship between the *bindingInformation* and the BindingData signed attribute.



```
value BindingData::= {

bindingType
"urn:nato:stanag:4778:bindinginformation:1:0",

bindingId "1234",

  bindingDataInfos {

  {

   mbId "ID-123456",

   algId {

    algorithm {1 2 840 10040 4 3}

   },

   metadataBindingDigest '00111100 .... 00001010'B

  }

}

}
```

Figure 2-6: Relationship between the Binding Information and the BindingData signed attribute

Processing bindingData

The value from the *Binding-Data* header field *binding-type* parameter SHALL be compared with the value from the id-bindingData bindingType field.

The identifier value from the *@URI* attribute of the *SignatureReference* element SHALL be compared with the id-bindingData bindingId field.

If the *binding-type* parameter and bindingType field both match with *urn:nato:stanag:4778:bindinginformation:1:0* and the bindingId and @URI values match the following process for each BindingDataInfo SHALL be performed:

1) Locate the *MetadataBinding* element from the *bindingInformation* stored in the *Binding-Data* header field *binding-data-object* parameter that contains the *@id* attribute value in the MetadataBindingIdentifier;
2) Normalise the *MetadataBinding* element conformant with Annex A XML Normalization process specified in this profile.
3) The output from the normalised *MetadataBinding* element SHALL be passed into the digest calculation along with the algId digest algorithm, and any associated parameters; and,
4) Compare the result of the digest calculation with the metadataBindingDigest.

For each *MetadataBinding* element from the *bindingInformation* there SHALL be a matching BindingDataInfo.

E.5. Signature Generation

The *MetadataBinding DataReference @URI* attribute value SHALL be used to dereference the MIME entity that is to be prepared for signing dependent upon the signed-only format (specified in Section 3.5 of SMIME) required.

In addition to the CMS profile specified in this document, the following procedures SHALL be adhered to:

- The procedures for generating the bindingData signed attribute are specified in Section Generating bindingData above.
- The procedures for message digest calculation are specified in Section 5.4 of CMS.
- The procedures for signature generation are as specified in Section 5.5 of CMS.

E.6. Signature Verification

In addition to the CMS profile specified in this document, the following procedures SHALL be adhered to:

- The procedures for signature verification are as specified in Section 5.6 of CMS.
- For the cryptographic protection for the integrity of the binding to be valid the procedures for processing the bindingData signed attribute specified in Section Processing bindingData above SHALL be adhered to and the comparisons of digests SHALL match.

CHAPTER 3 Simple Mail Transfer Protocol Binding Profile

3.1. Introduction

This profile specifies the mechanism for binding metadata to MIME entities, such as internet mail messages. 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 (Reference [6]) and MIME body (Reference [6]), and does not include the internet email headers (Reference [3]).

This profile supports binding metadata to a MIME entity that is a message including only the MIME message headers (Reference [6]) and MIME body (Reference [6]).

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.

This profile does not support the capability for referencing a sub-part or sub-parts of a message. A separate profile will specify how to bind metadata to a sub-part or sub-parts of a message.

3.2. Identification

The profile for SMTP is uniquely identified by the Canonical Identifier shown in Table 3-1.

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:smtp
Version Identifier	urn:nato:stanag:4778:profile:smtp:1:2

Table 3-1 Profile Identifiers

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 3-1.

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

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

3.3. Standards (Reference)

Reference [1] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
Reference [2] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium
Reference [3] IETF RFC 5322, “Internet Message Format”, at http://tools.ietf.org/html/rfc5322, October 2008.
Reference [4] IETF RFC 7444, “Security Labels in Internet Email”, K. Zeilenga and A. Melnikov, at http://tools.ietf.org/html/rfc7444, February 2015.
Reference [5] IETF RFC 2392, “Content-ID and Message-ID Uniform Resource Locators”, at http://tools.ietf.org/html/rfc2392, August 1998.
Reference [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
Reference [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.
Reference [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
Reference [9] IETF RFC 5234, “Augmented BNF for Syntax Specifications: ABNF”, at http://tools.ietf.org/html/rfc5234, January 2008
Reference [10] IETF RFC 822, “STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES”, at http://tools.ietf.org/html/rfc822, August 1982
Reference [11] IETF RFC 3986, “Uniform Resource Identifier (URI): Generic Syntax”, at http://tools.ietf.org/html/rfc3986, January 2005
Reference [12] IETF RFC 5646, “Tags for Identifying Languages”, at http://tools.ietf.org/html/rfc5646, September 2009

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

3.5. Internet Email Structure

The BDO is a detached BDO that MUST contain at least one *MetadataBinding* that contains a null *DataReference URI* attribute value. By conforming to this profile 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 as described at Reference [6].

The *DataReference xmime:contentType* attribute value is REQUIRED.

The *DataReference xmime:contentType* attribute value SHALL be `message/rfc822`.

This profile requires the use of a new header field that is based upon the protocol of the `SIO-Label` header field, as specified in (Reference [4]). The reason for a new header field is predicated upon the implied semantics of the `SIO-Label` header field for conveying the confidentiality of an electronic mail message as a whole (Reference [3]). This profile implies that any type of metadata (including confidentiality metadata) can be bound to any MIME entity.

The new header field name SHALL be “Binding-Data”, and its content consists of a set of key/value pairs.

Each key/value pair SHALL be referred to as a parameter.

Implementations conformant with this profile SHALL comply with the following formal header field syntax:

```
binding-data = “Binding-Data:” [FWS] binding-data-param-seq [FWS] CRLF

binding-data-param-seq = binding-data-param
       [ [FWS] ";" [FWS] binding-data-param-seq ]

binding-data-param = parameter
```

Parameter production SHALL conform to Reference [7].

As specified in Reference [7] parameter production permits white space immediately before and after the “=”.

FWS production SHALL conform to Reference [3].

CRLF production SHALL conform to Reference [9].

The Binding-Data header field SHALL be used to embed the BDO within the internet mail message.

The BDO SHALL be included in the Binding-Data header field “binding-data-object” parameter.

The Binding-Data “binding-data-object” parameter value SHALL be a quoted string that contains the base64 encoding of the BDO.

The Binding-Data “binding-data-object” parameter value SHALL always be present. It is noted that the Binding-Data “binding-data-object” parameter SHALL conform to Reference [7] 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 Binding-Data “binding-data-object” parameter SHALL be split into multiple Binding-Data “binding-data-object” parameters, as illustrated below[1].

```
binding-data-object*0="PFNlY0xhYmVsIHhtbG5zPSJodHRwOi8vZXhhbX";
binding-data-object*1="BsZS5jb20vc2VjLWxhYmVsLzAiPjxQb2xpY3lJ";
binding-data-object*3="48Q2xhc3NpZmljYXRpb24+MzwvQ2xhc3NpZmlj";
binding-data-object*2="ZGVudGlmaWVyIFVSST0idXJuOm9pZDoxLjEiLz";
binding-data-object*4="YXRpb24+PC9TZWNMYWJlbD4=";
```

It is noted that Binding-Data “binding-data-object” parameter value production implicitly allows for white space as Reference [7] relies on the Augmented Backus–Naur form (ABNF) as specified in Reference [10]. However, implementations SHALL be able to process Binding-Data “binding-data-object” parameter values that contain white space as illustrated below:

```
binding-data-object*0="PFNlY0xhYmVsIHhtbG5zPSJodHRwOi8vZXhhbXBsZS5jb20vc2VjLW
       xhYmVsLzAiPjxQb2xpY3lJ";
binding-data-object*1="ZGVudGlmaWVyIFVSST0idXJuOm9pZDoxLjEiLz48Q2xhc3NpZmlj
       YXRpb24+MzwvQ2xhc3NpZmlj";
binding-data-object*2="YXRpb24+PC9TZWNMYWJlbD4=";
```

It is also noted that Reference [7] allows for quoted-string values (for parameter production). As such, implementations SHALL be able to process Binding-Data “binding-data-object” parameter values that contain quoted-string values.

The Binding-Data “binding-type” parameter SHALL be a quoted string.

The Binding-Data “binding-type” parameter value SHALL be a Uniform Resource Identifier (URI, Reference [11]) that denotes the type, syntax and semantics for the binding mechanism represented by the Binding-Data “binding-data-object” parameter.

The Binding-Data “binding-type” SHALL always be present.

Implementations conformant with this profile SHALL contain a Binding-Data “binding-type” parameter with the value urn:nato:stanag:4778:bindinginformation:1:0.

[1] Note, as specified in Reference [7], the ordering of parameters can not be relied upon, therefore, the original parameter value is recovered by concatenating the multiple parameters, in order as specified in Reference [7].

Not all consumers may be metadata-aware and as such are not capable of processing the Binding-Data-Object “binding-data” parameter for the purposes of rendering a human-readable representation of the metadata bound to the MIME entity(or entities). To support consumers that are not metadata-aware the Binding-Data “marking” parameter MAY be used.

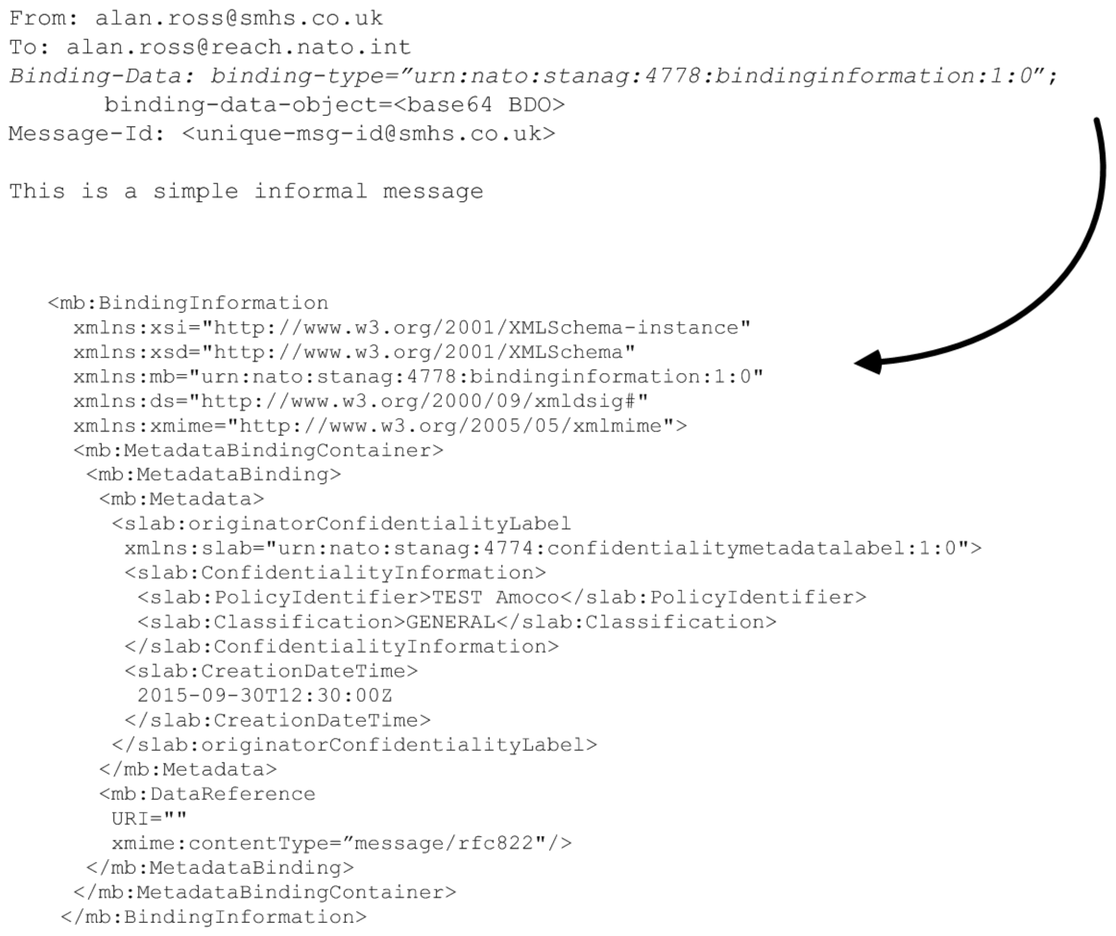
The Binding-Data “marking” parameter is a string that represents the human-readable representation of the metadata that is bound to the MIME entity (or entities) in the language indicated by the language field within the parameter value (if present, default language assumed if not present).

The Binding-Data “marking” parameter language field, if present, MUST have a value set to a language identifier specified in Tags for Identifying Languages (Reference [12]).

In the case a Binding-Data “marking” parameter is present with no language field within the parameter value, a default value of “en” SHALL be assumed to identify the language of the Binding-Data “marking” parameter.

Additional specifications for the production and semantics intended for the use of the Binding-Data “marking” MAY be provided in accompanying organizational, national or Community-Of-Interest implementation guidance documents.

An example of an Embedded BDO contained in the Binding-Data 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 3-1.



```
From: alan.ross@smhs.co.uk
To: alan.ross@reach.nato.int
Binding-Data: binding-type=”urn:nato:stanag:4778:bindinginformation:1:0”;
      binding-data-object=<base64 BDO>
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>TEST Amoco</slab:PolicyIdentifier>
       <slab:Classification>GENERAL</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
     </slab:originatorConfidentialityLabel>
    </mb:Metadata>
    <mb:DataReference
     URI=""
     xmime:contentType=”message/rfc822"/>
   </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
 </mb:BindingInformation>
```

Figure 3-1 Example of Binding Confidentiality Metadata Label to Email

3.6. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Chapter 2 Annex E CMS Binding Profile) 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 CMS Binding Profile (Chapter 2 Annex E) also apply to this profile for generating and validating cryptographic bindings.

It is RECOMMENDED that the Originator generate S/MIME multipart/signed format.

The Recipient SHALL support both the S/MIME multipart/signed and application/pkcs7-mime formats.

In addition to the Signature Generation Section of the CMS Binding Profile the following requirements SHALL be adhered to:

- The bindinginformation SHALL contain a new signature reference element, SignatureReference as specified in Section 4 of this profile that is a child element of the bindinginformation.
- The URI attribute of the SignatureReference element SHALL contain a fragment identifier (indicated by the presence of a “#” character and a substring to the right of the “#” in the URI) used to uniquely identify the S/MIME entity (as specified in the CMS Binding Profile).
- The MIME Content-Type header field value, that indicates the S/MIME entity, MAY be used as the SignatureReference xmime:contentType attribute value.

3.7. SignatureReference Schema

```xml
<?xml version='1.0' encoding='UTF-8'?>
<!--
********************************************************************
NATO UNCLASSIFIED
XML Schema To support dereferencing Cryptographic Artefacts.
I
/ \
-< + >-
\ /
I NCI AGENCY
## # #### # P.O. box 174
## # # # # 2501 CD The Hague
# # # # #
# # # # # # Core Enterprise Services
# ## #### #
A G E N C Y
********************************************************************
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:nato:stanag:4778:profile:cryptoartefact:1:0"
           xmlns:xml="http://www.w3.org/XML/1998/namespace" xmlns="urn:nato:stanag:4778:profile:cryptoartefact:1:0"
           xmlns:xmime="http://www.w3.org/2005/05/xmlmime"
           elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0" xml:lang="en">

 <xs:annotation>
  <xs:appinfo>
   <UniqueIdentifier>urn:nato:stanag:4778:profile:cryptoartefact:1:0</UniqueIdentifier>
   <Name>Dereference Cryptographic Artefact Schema</Name>
   <Definition>Schema To support dereferencing Cryptographic Artefacts</Definition>
   <VersionIndicator>1.0</VersionIndicator>
   <UsageGuidance>Used within NATO to Schema To support dereferencing Cryptographic Artefacts</UsageGuidance>
   <RestrictionType/>
   <RestrictionValue/>
  </xs:appinfo>
  <xs:documentation>
   The schema can be used to support dereferencing Cryptographic Artefacts.
  </xs:documentation>
 </xs:annotation>

  <xs:element name="SignatureReference">
  <xs:complexType>
   <xs:attribute name="URI" type="xs:anyURI" use="required"/>
   <xs:attribute ref="xmime:contentType"/>
   <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Figure 3-2 SignatureReference Schema

CHAPTER 4 Extensible Message And Presence Protocol Binding Profile

4.1. 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 is based upon the XEP-0258 (Reference [4]) specification to support carrying any type of metadata (including confidentiality metadata) contained in Embedded or Detached BDO for Message stanzas. As such, this profile supports the XMPP use cases for one-to-one instant messaging, multi-user chat and publish-subscribe notifications.

While XMPP-Core (Reference [7]) offers flexible extensibility for Message and Presence stanzas it is not the case for IQ stanzas. This profile specifies support for carrying a Detached BDO for IQ stanzas that contain item elements, such as XEP-0060 Publish-Subscribe (Reference [5]).

This profile does not support labelling Presence Stanzas.

4.2. Identification

The profile for XMPP is uniquely identified by the Canonical Identifier shown in Table 4-1.

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:xmpp
Version Identifier	urn:nato:stanag:4778:profile:xmpp:1:3

Table 4-1 Profile Identifiers

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 4-1.

This document deprecates the previous version identified by Version Identifier urn:nato:stanag:4778:profile:xmpp:1:2.

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

4.3. Standards (Reference)

Reference [1] IETF RFC 2634, “Enhanced Security Services for S/MIME”, at http://tools.ietf.org/html/rfc2634, June 1999.
Reference [2] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
Reference [3] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium
Reference [4] XEP-0258, “Security Labels in XMPP”, version 1.1, at http://www.xmpp.org/extensions/xep-0258.html, April 2013
Reference [5] XEP-0060, “Publish-Subscribe”, version 1.3, at http://www.xmpp.org/extensions/xep-0060.html, July 2010
Reference [6] IETF RFC 6122, “Extensible Messaging and Presence Protocol (XMPP): Address Format”, at http://tools.ietf.org/html/rfc6122, March 2011
Reference [7] IETF RFC 6120, “Extensible Messaging and Presence Protocol (XMPP): Core”, at http://tools.ietf.org/html/rfc6120, March 2011
Reference [8] IETF RFC 6121, “Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence”, at http://tools.ietf.org/html/rfc6121, March 2011
Reference [9] XEP-0030, “Service Discovery”, version 2.5rc3, at http://www.xmpp.org/extensions/xep-0030.html, October 2017

4.4. 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] and this profile.
- `Courier font` indicates syntax derived from XMPP (References [8]and [6]) and XEP-0258 (Reference [4]).

4.5. Message Stanza Structure

The `Message` stanza structure is specified in (Reference [8]). 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.

This profile specifies a new XML element, <BindingData/>, that SHALL be used to carry the BDO (refer to Section 8).

The <BindingData/> element SHALL contain one <BindingDataObject/> element, and an OPTIONAL <Marking/> element.

The Embedded or Detached BDO SHALL be contained as a child element of the <BindingDataObject/> element.

The <BindingDataObject/> element SHALL contain one or more BDOs.
Not all consumers may be metadata-aware and as such are not capable of processing the <BindingDataObject/> element for the purposes of rendering a human-readable representation of the metadata bound to the XMPP Message stanza (or subparts thereof). To support consumers that are not metadata-aware the <Marking/> element MAY be used.

The <Marking/> element, if present, SHALL contain a ‘xml:lang’ attribute to identify the language used to represent the human-readable rendering of the metadata.

Additional specifications for the production and semantics intended for the use of the <Marking/> element MAY be provided in accompanying organizational, national or Community-Of-Interest implementation guidance documents.

Figure 4-1 illustrates the high-level structure of a `Message` stanza that contains an Embedded BDO contained within a <BindingData/> element (as specified in this profile).

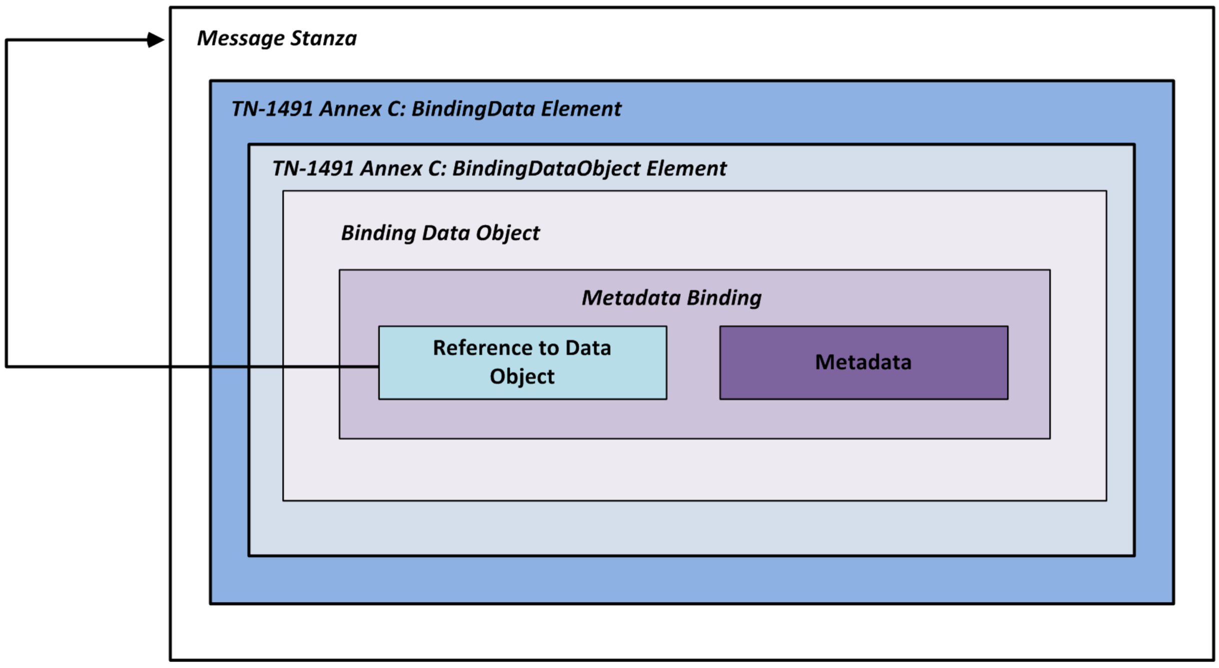
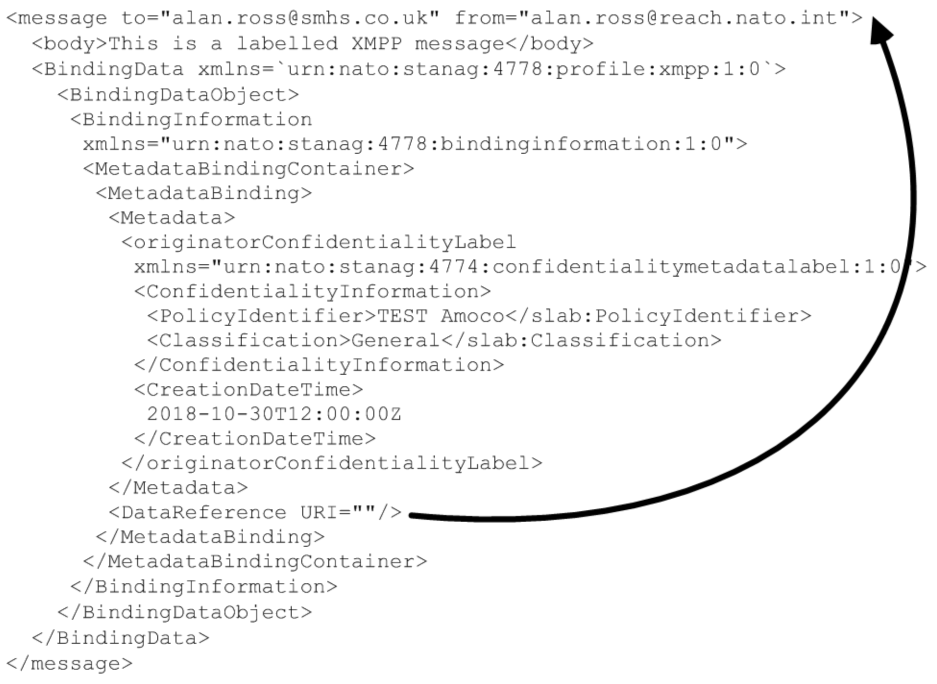


Figure 4-1 Structure of Message Stanza Containing Embedded BDO

An Embedded BDO MUST contain at least one *MetadataBinding* that contains a null *DataReference URI* attribute value only.

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 4-2. 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>
  <BindingData xmlns=`urn:nato:stanag:4778:profile:xmpp:1:0`>
    <BindingDataObject>
     <BindingInformation
      xmlns="urn:nato:stanag:4778:bindinginformation:1:0">
      <MetadataBindingContainer>
       <MetadataBinding>
        <Metadata>
         <originatorConfidentialityLabel
          xmlns="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
          <ConfidentialityInformation>
           <PolicyIdentifier>TEST Amoco</slab:PolicyIdentifier>
           <Classification>General</slab:Classification>
          </ConfidentialityInformation>
          <CreationDateTime>
           2018-10-30T12:00:00Z
          </CreationDateTime>
         </originatorConfidentialityLabel>
        </Metadata>
        <DataReference URI=""/>
       </MetadataBinding>
      </MetadataBindingContainer>
     </BindingInformation>
    </BindingDataObject>
  </BindingData>
</message>
```

Figure 4-2 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 `item` element (descendant of the `Message` stanza) to metadata is provided in Figure 4-3 below. This example illustrates the use of XPaths for referencing the `item` element. This example uses Confidentiality Metadata Labels (Reference [2]) as example metadata.

```
<message from="pubsub.smhs.co.uk" to="alan.ross@reach.nato.int">
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='princely_musings'>
      <item id='ae890ac52d0df67ed7cfdf51b644e901'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <title>Soliloquy</title>
          <summary>
To be, or not to be: that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take arms against a sea of troubles,
And by opposing end them?
          </summary>
          <link rel='alternate' type='text/html'
                href='http://denmark.lit/2003/12/13/atom03'/>
          <id>tag:denmark.lit,2003:entry-32397</id>
          <published>2003-12-13T18:30:02Z</published>
          <updated>2003-12-13T18:30:02Z</updated>
        </entry>
      </item>
    </items>
  </event>
  <BindingData xmlns=`urn:nato:stanag:4778:profile:xmpp:1:0`>
    <BindingDataObject>
     <mb:BindingInformation
      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>TEST Amoco</slab:PolicyIdentifier>
           <slab:Classification>GENERAL</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()='item' and namespace-uri()='http://jabber.org/protocol/pubsub#event'
and @id='ae890ac52d0df67ed7cfdf51b644e901']/[local-name()='event' and namespace-uri()='http://www.w3.org/2005/Atom']
           </ds:XPath>
          </ds:Transform>
         </ds:Transforms>
        </mb:DataReference>
       </mb:MetadataBinding>
      </mb:MetadataBindingContainer>
     </mb:BindingInformation>
    </BindingDataObject>
  </BindingData>
</message>
```

Figure 4-3 Example Detached Binding Data Object Contained in Message Stanza (XMPP)

4.6. IQ Stanza Structure

The `IQ` stanza structure is specified in (Reference [8]) and can only have a single unique child element. Other specifications define the elements to be used as the child of the `IQ` stanza. This profile specifies how to label `IQ` stanzas that contain `item` sub-elements exchanged between XMPP entities, such as XEP-0060 Publish-Subscribe (Reference [5]). Dependent upon system information exchange requirements it may be necessary that the child element (payload) of the item is bound to the metadata or subsets thereof are bound to the metadata. As such, Binding Information SHALL be represented as a Detached BDO.

This profile overrides the XMPP Publish-Subscribe specifications to support binding of metadata to the child element, and subsets thereof, for `item` elements within the `IQ` stanzas.

Figure C 4 illustrates the high-level structure of a `IQ` stanza that contains an `item` element which, in its turn, contains the detached BDO included within the <BindingData/> element.

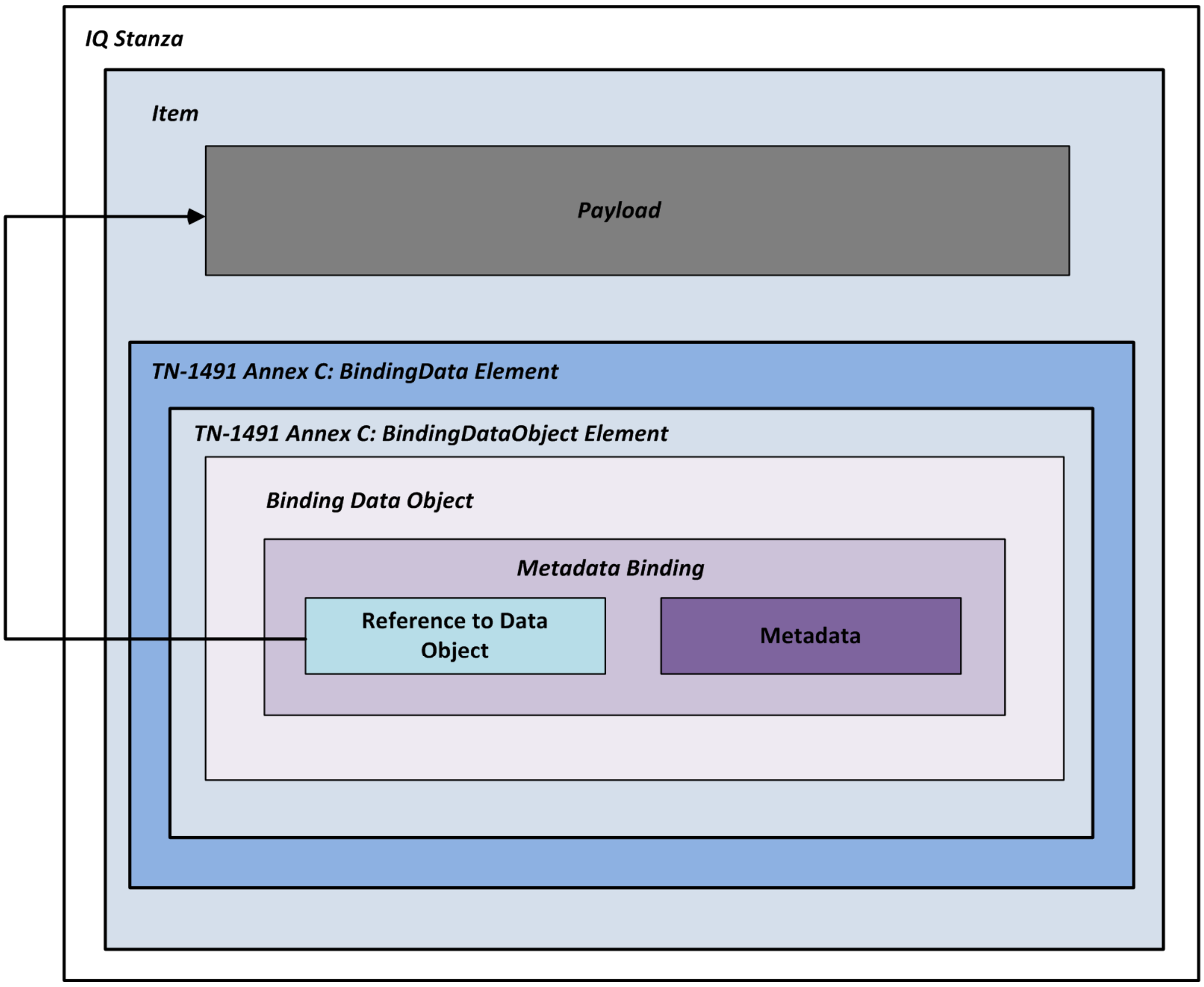


Figure 4-4 Structure of IQ Stanza containing a Detached BDO

The <BindingData/> element SHALL be the second child element of the `item` element.

When the `item` element contains the <BindingData/> child element, it SHALL have an id attribute with a value that uniquely identifies that item element.

The Detached BDO SHALL dereference the root node of the item element by containing one *DataReference URI* attribute value with a value that contains the value of the `item` id attribute value, along with 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 a child XPath element that dereferences the child element (payload) of the item 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 detached BDO contained in a `IQ` stanza that illustrates the binding of the payload (first child of the `IQ` stanza) to metadata is provided in Figure 4-5. This example illustrates the use of XPaths for referencing the payload (first child of the `IQ` stanza) element. This example uses Confidentiality Metadata Labels (Reference [2]) as example metadata.

```
<iq type='set' from="pubsub.smhs.co.uk" to="alan.ross@reach.nato.int" id='publish1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='princely_musings'>
      <item id='bnd81g37d61f49fgn581'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <title>Soliloquy</title>
          <summary>
To be, or not to be: that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take arms against a sea of troubles,
And by opposing end them?
          </summary>
          <link rel='alternate' type='text/html'
                href='http://denmark.lit/2003/12/13/atom03'/>
          <id>tag:denmark.lit,2003:entry-32397</id>
          <published>2003-12-13T18:30:02Z</published>
          <updated>2003-12-13T18:30:02Z</updated>
        </entry>
        <BindingData xmlns=`urn:nato:stanag:4778:profile:xmpp:1:0`>
         <BindingDataObject>
          <mb:BindingInformation
           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>TEST Amoco</slab:PolicyIdentifier>
                 <slab:Classification>GENERAL</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()='item' and namespace-uri()='http://jabber.org/protocol/pubsub'
and @id='ae890ac52d0df67ed7cfdf51b644e901']/[local-name()='event' and namespace-uri()='http://www.w3.org/2005/Atom']
                </ds:XPath>
               </ds:Transform>
              </ds:Transforms>
             </mb:DataReference>
            </mb:MetadataBinding>
           </mb:MetadataBindingContainer>
          </mb:BindingInformation>
         </BindingDataObject>
        </BindingData>
      </item>
    </publish>
  </pubsub>
</iq>
```

Figure 4-5 Example Detached BDO contained in IQ Stanza

4.7. BindingData Schema

```
<?xml version='1.0' encoding='UTF-8'?>
<!--
*********************************************************************
NATO UNCLASSIFIED
XML Schema for carrying STANAG 4778 Binding Information
in XMPP stanzas.
I
/ \
-< + >-
\ /
I NCI AGENCY
## # #### # P.O. box 174
## # # # # 2501 CD The Hague
# # # # #
# # # # # # Core Enterprise Services
# ## #### #
A G E N C Y
*********************************************************************
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:nato:stanag:4778:profile:xmpp:1:0"
           xmlns:xml="http://www.w3.org/XML/1998/namespace" xmlns="urn:nato:stanag:4778:profile:xmpp:1:0"
           elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0" xml:lang="en">

 <xs:annotation>
  <xs:appinfo>
   <UniqueIdentifier>urn:nato:stanag:4778:profile:xmpp:1:0</UniqueIdentifier>
   <Name>XMPP Binding Information Wrapper Schema</Name>
   <Definition>Schema for carrying STANAG 4778 Binding Information in XMPP Stanzas</Definition>
   <VersionIndicator>1.4</VersionIndicator>
   <UsageGuidance>Used within NATO to carry STANAG 4778 Binding Information in XMPP Stanzas</UsageGuidance>
   <RestrictionType/>
   <RestrictionValue/>
  </xs:appinfo>
  <xs:documentation>
   The schema can be used to carry STANAG 4778 Binding Information in XMPP Stanzas as specified in TN-1491 Edition 3.
  </xs:documentation>
 </xs:annotation>

 <xs:import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="xml.xsd"/>

 <xs:complexType name="markingType">
  <xs:annotation>
   <xs:appinfo>
    <UniqueIdentifier>urn:nato:stanag:4778:profile:xmpp:1:0:appinfo:markingType</UniqueIdentifier>
    <Name>Marking Type</Name>
    <Definition>Human-readable string.</Definition>
    <VersionIndicator>1.0</VersionIndicator>
    <UsageGuidance></UsageGuidance>
    <RestrictionType></RestrictionType>
    <RestrictionValue></RestrictionValue>
   </xs:appinfo>
   <xs:documentation>Human-readable representation of metadata bound to XMPP stanzas (or subsets thereof).</xs:documentation>
   <xs:documentation>String which may be used depending on organizational, national or Community-of-Interest policy</xs:documentation>
  </xs:annotation>
  <xs:attribute ref="xml:lang" use="required"/>
  <xs:anyAttribute processContents="lax"/>
 </xs:complexType>

 <xs:complexType name="bindingDataObjectType">
  <xs:annotation>
   <xs:appinfo>
    <UniqueIdentifier>urn:nato:stanag:4778:profile:xmpp:1:0:appinfo:bindingDataObjectType</UniqueIdentifier>
    <Name>BDO Type</Name>
    <Definition>An object to carry binding information for associating metadata to XMPP data objects.</Definition>
    <VersionIndicator>1.0</VersionIndicator>
    <UsageGuidance></UsageGuidance>
    <RestrictionType></RestrictionType>
    <RestrictionValue></RestrictionValue>
   </xs:appinfo>
   <xs:documentation>
   </xs:documentation>
  </xs:annotation>
  <xs:sequence>
   <xs:any namespace="##other" processContents="lax" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
 </xs:complexType>

 <xs:element name="BindingData">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="Marking" type="markingType" minOccurs="0"/>
    <xs:element name="BindingDataObject" type="bindingDataObjectType"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Figure 4-6 BindingData Schema

4.8. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Chapter 2 Annexes A, B and C XML Signature Binding Profile) 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 (Chapter 2 Annexes A, B and C XML Signature Binding Profile) also apply to this profile for generating and validating cryptographic bindings.

With 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 for XML Signature Core Signature Generation and Verification:

1) If the XMPP Binding Profile is supported between XMPP entities (an originating entity and a recipient entity) without re-scoping of the originating `Message` stanza:
    a. The Binding Information MAY be represented either as an Embedded BDO (the metadata SHALL be bound to the entire `Message` stanza) or a Detached BDO (the metadata SHALL be bound to a subset of the `Message` stanza); and,
    b. The content namespace SHALL be `jabber:client`.
2) Otherwise:
    a. The Binding Information SHALL be represented as a Detached BDO (the metadata SHALL be bound to a subset of the stanza); and,
    b. The content namespace SHALL be `jabber:client` .

CHAPTER 5 Office Open XML Formats Binding Profile

5.1. 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.

5.2. Identification

The profile for OOXML is uniquely identified by the Canonical Identifier shown in Table 5-1.

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:ooxml
Version Identifier	urn:nato:stanag:4778:profile:ooxml:1:2

Table 5-1 Profile Identifiers

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 5-1.

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

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

5.3. Standards (Reference)

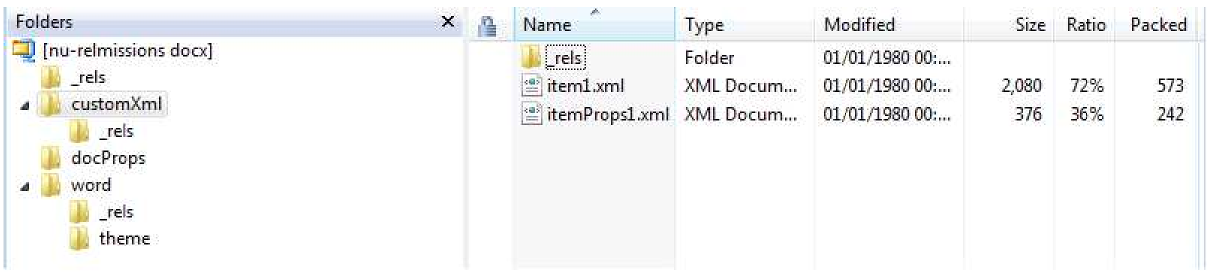
Reference [1] ISO/IEC 29500-2 “Office Open XML File Formats - Part 2: Open Packaging Conventions”, at http://standards.iso.org/ittf/PubliclyAvailableStandards/c061796_ISO_IEC_29500-2_2012.zip, August 2012
Reference [2] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
Reference [3]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

5.4. 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].

5.5. Structure

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



```
Folders
[nu-relmissions docx]
    _rels
    customXml
        _rels
    docProps
    word
        _rels
        theme
```

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 5-1 General Structure of an OPC Package

The structure, as shown in Figure 5-1, 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.)

5.6. 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 5-2).

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.
	/xl/media/*	The media (e.g.) pictures embedded on the worksheets.
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.

Table 5-2 Packages Files to be Referenced in a Binding to a Complete Document[2]

The common document properties package files (where they are present within the package) SHALL also be referenced in the binding (see Table 5-2).

Additional package files, beyond those listed in Table 5-2 and Table 5-3, MAY be referenced in the binding (e.g. packages files created by COI-specific Office Add-Ins).

[2] 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.

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.

Table 5-3 Common Packages Files to be Referenced in a Binding to a Complete Document

Figure 5-2 shows the contents of a CustomXML file, stored in /customXml/item1.xml, for a simple Microsoft Word document containing an embedded image. Its 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>TEST Amoco</slab:PolicyIdentifier>
         <slab:Classification>GENERAL</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 5-2 CustomXML file

5.7. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Chapter 2 Annexes A, B and C XML Signature Binding Profile) 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 (Chapter 2 Annexes A, B and C XML Signature Binding Profile) also apply to this profile for generating and validating cryptographic bindings.

It is RECOMMENDED that the requirements specified in Cryptographic Artefacts binding profile (Chapter 2 Annex A) URI Schemes are adhered to.

CHAPTER 6 Simple Object Access Protocol Binding Profile

6.1. Introduction

It is recognized 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.

6.2. Identification

The profile for SOAP is uniquely identified by the Canonical Identifier shown in Table 6-1.

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

Table 6-1 Profile Identifiers

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 6-1.

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.

6.3. Standards (Reference)

Reference [1] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
Reference [2] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium
Reference [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.
Reference [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.
Reference [5] W3C XMLDSIG-CORE, 2008, "XML- Signature Syntax and Processing (Second Edition)", at http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/, W3C Recommendation, W3C, 10 June 2008

6.4. Namespace Constraints

Table 6-2 summarises the XML namespaces and corresponding prefixes used throughout for the binding of metadata to SOAP data objects and portions thereof.

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

Table 6-2 XML Namespaces and Prefixes

6.5. 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.

6.6. 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 6-1. 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>TEST Amoco</slab:PolicyIdentifier>
         <slab:Classification>GENERAL</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 6-1 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 6-2. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

Figure 6-2 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>TEST Amoco</slab:PolicyIdentifier>
             <slab:Classification>GENERAL</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 6-2 Example Detached BDO for SOAP

6.7. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Chapter 2 Annexes A, B and C XML Signature Binding Profile) 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 (Chapter 2 Annexes A, B and C XML Signature Binding Profile) also apply to this profile for generating and validating cryptographic bindings.

It is RECOMMENDED that the requirements specified in Cryptographic Artefacts binding profile (Chapter 2 Annex A) URI Schemes are adhered to.

CHAPTER 7 Representational State Transfer (REST) Binding Profile

7.1. 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.

7.2. Identification

The profile for REST is uniquely identified by the Canonical Identifier shown in Table 7-1.

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:rest
Version Identifier	urn:nato:stanag:4778:profile:rest:1:2

Table 7-1 Profiles Identifier

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 7-1.

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

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

7.3. Standards (Reference)

Reference [1] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
Reference [2] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium
Reference [3] IETF RFC 7444, “Security Labels in Internet Email”, K. Zeilenga and A. Melnikov, at http://tools.ietf.org/html/rfc 7444, February 2015.
Reference [4] IETF RFC 7230, “Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing”, at http://tools.ietf.org/html/rfc7230, June 2014.
Reference [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.
Reference [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
Reference [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

7.4. 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 SIO -Label (Reference [3]) and HTTP (Reference [4]) referenced in this Annex.

7.5. 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 Binding-Data header field (refer to

Chapter 3:, based on the SIO-Label Reference [4]) as a HTTP Entity message header for HTTP Entity requests and responses for storing the BDO.
The BDO is a detached 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 Binding-Data header field SHALL be used to embed the BDO within the HTTP Entity message.

The BDO SHALL be included in the Binding-Data header field “binding-data-object” parameter.

The Binding-Data “binding-data-object” parameter value SHALL be the base64 encoding of the BDO.

The Binding-Data “binding-data-object” parameter value SHALL always be present. 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 Binding-Data “binding-data-object” parameter MUST not support Reference [7] for parameter value continuation.

The Binding-Data “binding-type” parameter SHALL be present with the value *urn:nato:stanag:4778:bindinginformation:1:0*.

Figure 7-1 illustrates an HTTP POST request with the Binding-Data HTTP header field with the header field value as specified in this Binding Profile. Figure 7-2 illustrates the base64 decoded value of the Binding-Data “binding-data-object” value parameter. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

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

<Document>
….
</Document>
```

Figure 7-1 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>TEST Amoco</slab:PolicyIdentifier>
           <slab:Classification>GENERAL</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 7-2 Base64 Decoded Embedded BDO illustrating the binding of the HTTP POST REQUEST

7.6. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Chapter 2 Annexes A, B and C XML Signature Binding Profile) 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 (Chapter 2 Annexes A, B and C XML Signature Binding Profile) also apply to this profile for generating and validating cryptographic bindings.

It is RECOMMENDED that the requirements specified in Cryptographic Artefacts binding profile (Chapter 2 Annex A) URI Schemes are adhered to.

In addition, 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.

CHAPTER 8 Generic Open Packaging Convention Binding Profile

8.1. 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.

8.2. Identification

The profile for generic OPC is uniquely identified by the Canonical Identifier shown in Table 8-1.

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:gopc
Version Identifier	urn:nato:stanag:4778:profile:gopc:1:2

Table 8-1 Profiles Identifier

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 8-1.

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

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

8.3. Standards (Reference)

Reference [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
Reference [2] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
Reference [3] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium

8.4. 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].

8.5. 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 8-1.

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*.

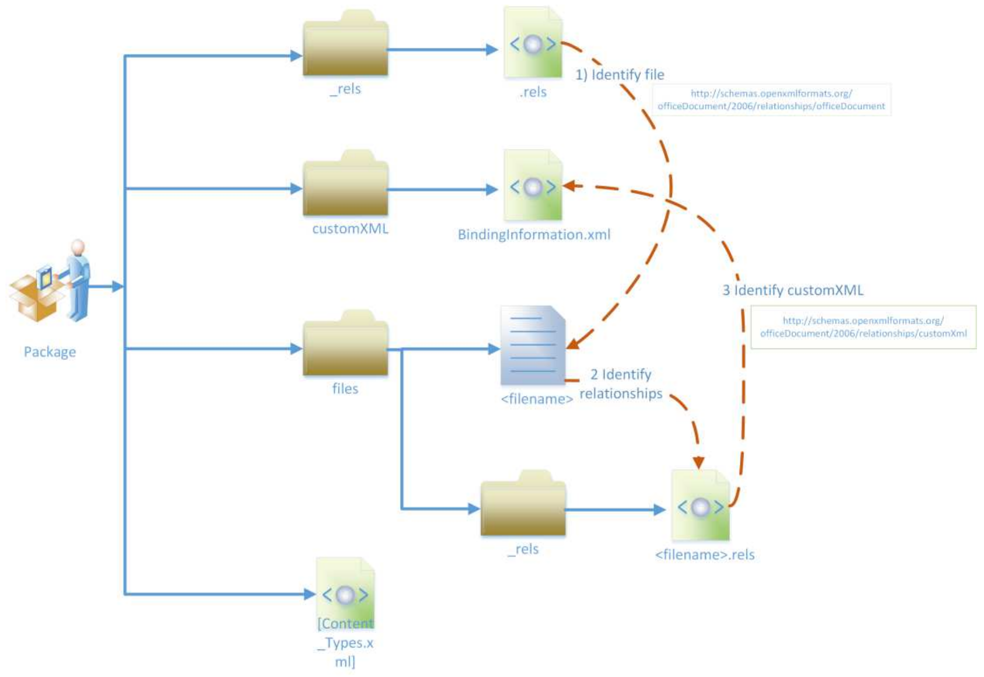


Figure 8-1 OPC Structure for packaging BindingInformation with an arbitrary file

This approach allows multiple files, of different types, to be held within the same package and be bound to distinct metadata. Figure 8-2 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>TEST Amoco</slab:PolicyIdentifier>
           <slab:Classification>GENERAL</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 8-2 Example Packaged CustomXML file

8.6. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Chapter 2 Annexes A, B and C XML Signature Binding Profile) 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 (Chapter 2 Annexes A, B and C XML Signature Binding Profile) also apply to this profile for generating and validating cryptographic bindings.

It is RECOMMENDED that the requirements specified in Cryptographic Artefacts binding profile (Chapter 2 Annex A) URI Schemes are adhered to.

CHAPTER 9 Sidecar Files Binding Profile

9.1. 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.

9.2. Identification

The profile for sidecar files is uniquely identified by the Canonical Identifier shown in Table 9-1.

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:sidecar
Version Identifier	urn:nato:stanag:4778:profile:sidecar:1:2

Table 9-1 Profiles Identifier

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 9-1.

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

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

9.3. Standards (Reference)

Reference [1] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium

Reference [2] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium

9.4. 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].

9.5. 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 9-1.

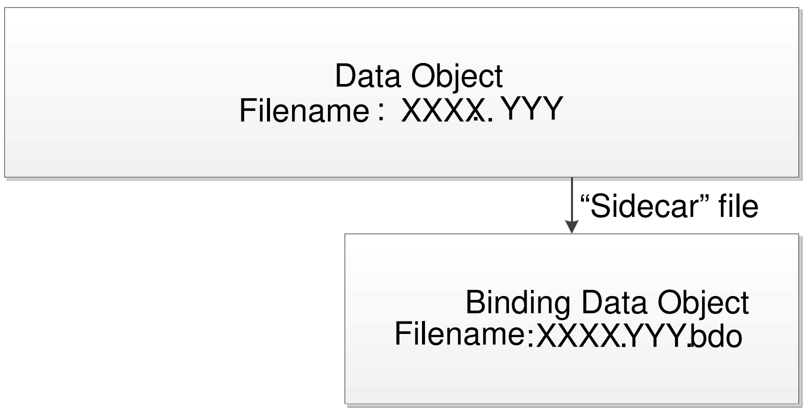


Figure 9-1 BDO as a Sidecar File

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 9-2 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>TEST Amoco</slab:PolicyIdentifier>
          <slab:Classification>GENERAL</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 9-2 Example Sidecar file

9.6. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Chapter 2 Annexes A, B and C XML Signature Binding Profile) 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 (Chapter 2 Annexes A, B and C XML Signature Binding Profile) also apply to this profile for generating and validating cryptographic bindings.

It is RECOMMENDED that the requirements specified in Cryptographic Artefacts binding profile (Chapter 2 Annex A) URI Schemes are adhered to.

CHAPTER 10 Extensible Metadata Platform Binding Profile

10.1. 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.

10.2. Identification

The profile for XMP is uniquely identified by the Canonical Identifier shown in Table 10-1.

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

Table 10-1 Profiles Identifier

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 10-1.

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.

10.3. Standards (Reference)

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

10.4. 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]).

10.5. 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 10-1:

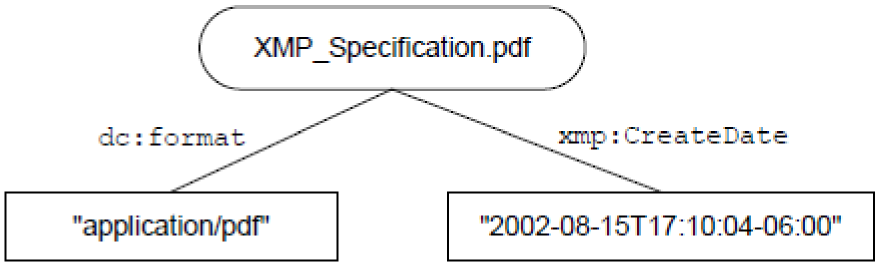


Figure 10-1 Two Simple XMP Properties, dc:format and xmp:CreateDate

- structure – a container for zero or more named fields – see Figure 10-2:; and

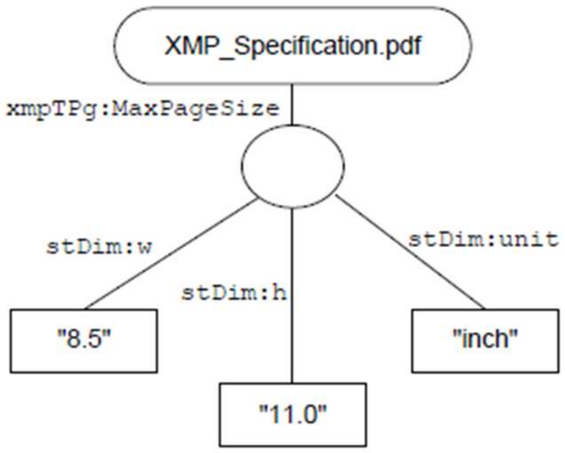


Figure 10-2 An XMP Structured Property, xmpTPg:MaxPageSize containing 3 fields

- array – a container for zero or more items e.g. to support multi-valued properties – see Figure 10-3:

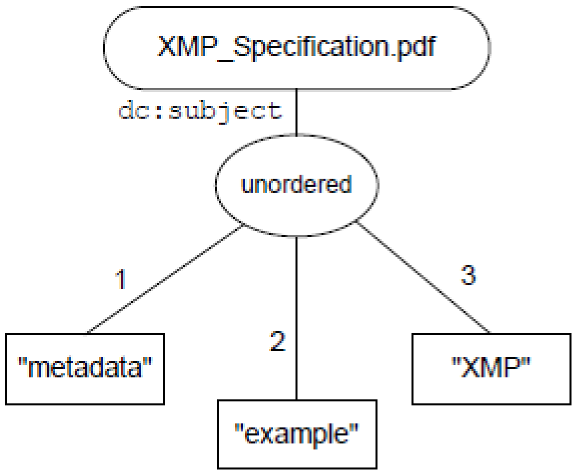


Figure 10-3 An XMP Array Property, dc:subject containing 3 items

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].
- The *BindingInformation* SHALL be a 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 10-4 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>
      &lt;mb:BindingInformation

xmlns:slab=&quot;urn:nato:stanag:4774:confidentialitymetadatalabel:1:0&quot;
      xmlns:xsi=&quot;http://www.w3.org/2001/XMLSchema-instance&quot;
      xmlns:xsd=&quot;http://www.w3.org/2001/XMLSchema&quot;
      xmlns:mb=&quot;urn:nato:stanag:4778:bindinginformation:1:0&quot;
      xmlns:ds=&quot;http://www.w3.org/2000/09/xmldsig#&quot;
      xmlns:xmime=&quot;http://www.w3.org/2005/05/xmlmime&quot;&gt;
      &lt;mb:MetadataBindingContainer&gt;
        &lt;mb:MetadataBinding&gt;
          &lt;mb:Metadata&gt;
            &lt;slab:originatorConfidentialityLabel

xmlns:slab=&quot;urn:nato:stanag:4774:confidentialitymetadatalabel:1:0&quot;&
gt;
              &lt;slab:ConfidentialityInformation&gt;
                &lt;slab:PolicyIdentifier&gt;TEST
Amoco&lt;/slab:PolicyIdentifier&gt;
                &lt;slab:Classification&gt;
GENERAL&lt;/slab:Classification&gt;
              &lt;/slab:ConfidentialityInformation&gt;
              &lt;slab:CreationDateTime&gt;
                2015-09-30T12:30:00Z
              &lt;/slab:CreationDateTime&gt;
            &lt;/slab:originatorConfidentialityLabel&gt;
          &lt;/mb:Metadata&gt;
          &lt;mb:DataReference URI=&quot;&quot;  /&gt;
        &lt;/mb:MetadataBinding&gt;
      &lt;/mb:MetadataBindingContainer&gt;
    &lt;/mb:BindingInformation&gt;
    </mbxmp:bindingInformation>
  </rdf:Description>
</rdf:RDF>
```

Figure 10-4 Example XMP Binding Packet (Canonical form)

Figure 10-5 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=”&lt;mb:BindingInformation
xmlns:slab=&quot;urn:nato:stanag:4774:confidentialitymetadatalabel:1:0&quot;
      xmlns:xsi=&quot;http://www.w3.org/2001/XMLSchema-instance&quot;
      xmlns:xsd=&quot;http://www.w3.org/2001/XMLSchema&quot;
      xmlns:mb=&quot;urn:nato:stanag:4778:bindinginformation:1:0&quot;
      xmlns:ds=&quot;http://www.w3.org/2000/09/xmldsig#&quot;
      xmlns:xmime=&quot;http://www.w3.org/2005/05/xmlmime&quot;&gt;
      &lt;mb:MetadataBindingContainer&gt;
        &lt;mb:MetadataBinding&gt;
          &lt;mb:Metadata&gt;
            &lt;slab:originatorConfidentialityLabel
xmlns:slab=&quot;urn:nato:stanag:4774:confidentialitymetadatalabel:1:0&quot;&
gt;
              &lt;slab:ConfidentialityInformation&gt;
                &lt;slab:PolicyIdentifier&gt;TEST
Amoco&lt;/slab:PolicyIdentifier&gt;
                &lt;slab:Classification&gt;
GENERAL&lt;/slab:Classification&gt;
              &lt;/slab:ConfidentialityInformation&gt;
              &lt;slab:CreationDateTime&gt;
                2015-09-30T12:30:00Z
              &lt;/slab:CreationDateTime&gt;
            &lt;/slab:originatorConfidentialityLabel&gt;
          &lt;/mb:Metadata&gt;
          &lt;mb:DataReference URI=&quot;&quot;  /&gt;
        &lt;/mb:MetadataBinding&gt;
      &lt;/mb:MetadataBindingContainer&gt;
    &lt;/mb:BindingInformation&gt;”
  </rdf:Description>
 </rdf:RDF>
```

Figure 10-5 Example XMP Binding Packet (Equivalent form)

10.6. XMP Sidecar File

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 10-6 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>
      &lt;mb:BindingInformation

xmlns:slab=&quot;urn:nato:stanag:4774:confidentialitymetadatalabel:1:0&quot;
      xmlns:xsi=&quot;http://www.w3.org/2001/XMLSchema-instance&quot;
      xmlns:xsd=&quot;http://www.w3.org/2001/XMLSchema&quot;
      xmlns:mb=&quot;urn:nato:stanag:4778:bindinginformation:1:0&quot;
      xmlns:ds=&quot;http://www.w3.org/2000/09/xmldsig#&quot;
      xmlns:xmime=&quot;http://www.w3.org/2005/05/xmlmime&quot;&gt;
      &lt;mb:MetadataBindingContainer&gt;
        &lt;mb:MetadataBinding&gt;
          &lt;mb:Metadata&gt;
            &lt;slab:originatorConfidentialityLabel
xmlns:slab=&quot;urn:nato:stanag:4774:confidentialitymetadatalabel:1:0&quot;&
gt;
              &lt;slab:ConfidentialityInformation&gt;
                &lt;slab:PolicyIdentifier&gt;TEST
Amoco&lt;/slab:PolicyIdentifier&gt;
                &lt;slab:Classification&gt;
GENERAL&lt;/slab:Classification&gt;
              &lt;/slab:ConfidentialityInformation&gt;
              &lt;slab:CreationDateTime&gt;
                2015-09-30T12:30:00Z
              &lt;/slab:CreationDateTime&gt;
            &lt;/slab:originatorConfidentialityLabel&gt;
          &lt;/mb:Metadata&gt;
          &lt;mb:DataReference URI=&quot;./example.mpg&quot;
xmime:contentType=&quot;audio/mpeg&quot; /&gt;
        &lt;/mb:MetadataBinding&gt;
      &lt;/mb:MetadataBindingContainer&gt;
    &lt;/mb:BindingInformation&gt;
    </mbxmp:bindingInformation>
  </rdf:Description>
 </rdf:RDF>
```

Figure 10-6 Example XMP Sidecar file (example.mpg.xmp)

10.7. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Chapter 2 Annexes A, B and C XML Signature Binding Profile) 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 (Chapter 2 Annexes A, B and C XML Signature Binding Profile) also apply to this profile for generating and validating cryptographic bindings.

It is RECOMMENDED that the requirements specified in Cryptographic Artefacts binding profile (Chapter 2 Annex A) URI Schemes are adhered to.

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 ANNEX A) 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>
```

CHAPTER 11 Web Service Messaging Profile Binding Profile

11.1. 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.

11.2. Identification

The profile for WSMP is uniquely identified by the Canonical Identifier shown in Table 11-1.

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

Table 11-1 Profiles Identifier

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 11-1.

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

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

11.3. Standards (Reference)

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

11.3. Namespace Constraints

Table 11-2 below summarises the XML namespaces and corresponding prefixes used throughout for the binding of metadata to WSMP data objects and portions thereof.

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

Table 11-2 XML Namespaces and Prefixes

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

11.5. 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:

a) a WSMP message wrapper with one or more WSMP data wrappers; or,

b) 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:

a) an Embedded Binding Data Object (BDO) that binds metadata to the WSMP message wrapper `WSMPMsg`; and/or,
b) 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 `WSMPMsg/MetadataBinding` element that SHALL include the *BindingInformation* element (as a child element of the `WSMPMsg/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 applicable , 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:wsmp:1:1'
</XPath>
```

Figure 11-1 XPath element

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.

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 11-2. 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>TEST Amoco</slab:PolicyIdentifier>
         <slab:Classification>GENERAL</slab:Classification>
        </slab:ConfidentialityInformation>
        <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()='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: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>TEST Amoco</slab:PolicyIdentifier>
         <slab:Classification>GENERAL</slab:Classification>
        </slab:ConfidentialityInformation>
        <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 11-2 Example WSMP Metadata Binding

11.6. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Chapter 2 Annexes A, B and C XML Signature Binding Profile) 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 (Chapter 2 Annexes A, B and C XML Signature Binding Profile) also apply to this profile for generating and validating cryptographic bindings.

It is RECOMMENDED that the requirements specified in Cryptographic Artefacts binding profile (Chapter 2 Annex A) URI Schemes are adhered to.

CHAPTER 12 Common XML Artefacts Binding Profile

12.1. 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 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])
- Genericode 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]).

12.2. Identification

The profiles for XML Artefacts are uniquely identified by the Canonical Identifiers shown in Table 12-1.

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
Genericode 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

Table 12-1 XML Artefact Profile Identifiers

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 12-1.

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

12.3. Standards (Reference)

Reference [1] NATO Standardization Agency (NSA) STANAG 4774, “Confidentiality Metadata Label Syntax”, MCMSB, NATO Headquarters, Brussels, Belgium, 14 April 2016.
Reference [2] NATO Standardization Agency (NSA) STANAG 4778, “Metadata Binding Mechanism”, MCMSB, NATO Headquarters, Brussels, Belgium
Reference [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.
Reference [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/PubiclyAvailableStandards/c055982_ISO_IEC_19757-3_2016.zip, 15 January 2016.
Reference [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.
Reference [6] Organization for the Advancement of Structured Information Standards (OASIS) “Code List Representation (Genericode)”, Version 1.0 , at https://docs.oasis-open.org/codelist/cs-genericode-1.0/doc/oasis-code-list-representation-genericode.pdf, 28 December 2007.
Reference [7] Organization for the Advancement of Structured Information Standards (OASIS) “Context/value Association using genericode 1.0”, at http://docs.oasis-open.org/codelist/ns/ContextValueAssociation/1.0/doc/context-value-association.pdf, 15 April 2010.
Reference [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.
Reference [9] Security Policy Information File (SPIF) at http://www.xmlspif.org/
Reference [10] http://docs.oasis-open.org/codelist/cs01-ContextValueAssociation-1.0/xsd/ContextValueAssociation.xsd

12.4. Namespace Constraints

Table 12-2 below summarises the XML namespaces and corresponding prefixes used throughout for the binding of metadata to Common XML Artefact data objects and portions thereof.

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

Table 12-2 XML Namespaces and Prefixes

12.5. 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.

12.6. 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 12-1. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

```
<xsd:schema xmlns:xs=”http://www/w3.rog/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>TEST Amoco</slab:PolicyIdentifier>
                  <slab:Classification>GENERAL</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 12-1 Example XML Schema Metadata Binding

12.7. 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 12-2. 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>TEST Amoco</slab:PolicyIdentifier>
              <slab:Classification>GENERAL</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 12-2 Example XML Schematron Metadata Binding

12.8. 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 12-3. 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.xsl”/>
  <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>TEST Amoco</slab:PolicyIdentifier>
              <slab:Classification>GENERAL</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 12-3 Example XML Stylesheet Metadata Binding

12.9. Generic Codelist Structure

The Genericode Code List (`https://docs.oasis-open.org/codelist/cs-genericode-1.0/xsd/genericode.xsd`) contains an `Annotation` element which allows for both human readable and machine-processible, 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 Genericode 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`). A BDO SHALL NOT be embedded in any other location within the Genericode 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 Genericode CodeList that illustrates the binding of the data, contained in the parent `gc:CodeList` element, to metadata is provided in Figure 12-4. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

```
<gc:CodeList xmlns:gc=” http://docs.oasis-open.org/codelist/ns/genericode/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>TEST Amoco</slab:PolicyIdentifier>
                  <slab:Classification>GENERAL</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 12-4 Example XML Genericode Metadata Binding

12.10. Context/Value Association Structure

The Context/Value Association (Reference [10]) 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 12-5. 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>TEST Amoco</slab:PolicyIdentifier>
                  <slab:Classification>GENERAL</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 12-5 Example XML Context/Value Association Metadata Binding

12.11. Security Policy Information File Structure

The Security Policy Information File (`http://ww.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 12-6. 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="TEST Amoco" id="1.2.840.113549.1.9.16.7.1" />
  <spif:securityClassifications>
    <spif:securityClassification name="GENERAL" lacv="6" hierarchy="6">
      <spif:markingData xml:lang="fr" phrase="GÉNÉRAL">
        <spif:code>documentStart</spif:code>
      </spif:markingData>
    </spif:securityClassification>
    <spif:securityClassification name="CONFIDENTIAL" lacv="7" hierarchy="7">
      <spif:markingData xml:lang="fr" phrase="CONFIDENTIEL">
        <spif:code>documentStart</spif:code>
      </spif:markingData>
    </spif:securityClassification>
    <spif:securityClassification name="HIGHLY CONFIDENTIAL" lacv="8"
hierarchy="10">
      <spif:markingData xml:lang="fr" phrase="TRÈS CONFIDENTIEL">
        <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>TEST Amoco</slab:PolicyIdentifier>
                <slab:Classification>GENERAL</slab:Classification>
                <slab-ext:Marking xml:lang="en">TEST Amoco GENERAL</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 12-6 Example XML SPIF Metadata Binding

12.12. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Chapter 2 Annexes A, B and C XML Signature Binding Profile) 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 (Chapter 2 Annexes A, B and C XML Signature Binding Profile) also apply to this profile for generating and validating cryptographic bindings.

It is RECOMMENDED that the requirements specified in Cryptographic Artefacts binding profile (Chapter 2 Annex A) URI Schemes are adhered to.

INTENTIONALLY BLANK

ADatP-4778.2(A)(1)