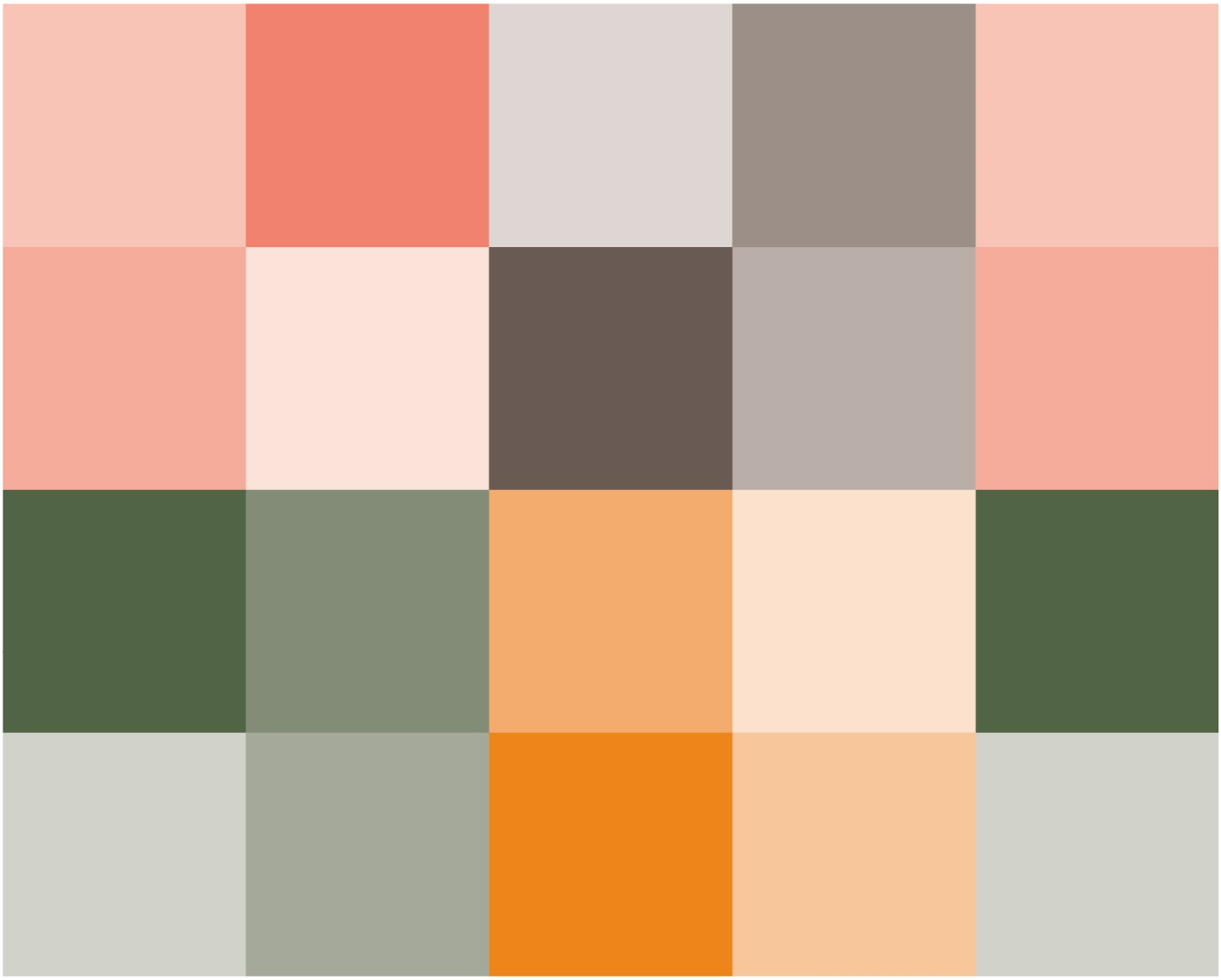


Kuverteringsprofil XHE

Beskrivning av tekniska kuverteringsprofil för XHE

Version: 1.0

Målgrupper: IT-arkitekter, utvecklare

MYNDIGHETEN FÖR DIGITAL FÖRVALTNING
Agency for Digital Government

Sammanfattning

Sammanfattning av kuverteringsprofilen för XHE

Denna kuverteringsprofil beskriver teknologiskt hur ett meddelande paketeras och kuverteras med hjälp av den tekniska specifikationen XHE, på ett sätt som gör det möjligt att ha en rationell hantering i verksamheters system, applikationer och accesspunkter, vilka inte är beroende av ett meddelandes interna format och struktur. Kuverteringsprofilen som är baserad på XML teknologier ger dessutom möjligheter att signera kuvertet och att kryptera nyttolasten (verksamhetsmeddelandet).

Kuverteringsprofilen är baserad på den begreppsmässiga och logiska meddelandemodellen som definierar vad ett meddelande och kuvert är och består av. Se ”eDelivery Transportinfrastruktur – Beskrivning” för mer information om meddelandemodellen.

Innehållsförteckning

Sammanfattning **1**

1 Inledning **3**

1.1 Dokumentstruktur *4*

1.2 Målgrupper *4*

1.3 Referenser *4*

1.4 Federationsspecifika anpassningar *5*

2 Meddelandemodell enligt XHE **5**

2.1 Informationsmodell *5*

2.2 Syntaxmappning *7*

2.3 Namnrymder som används i kuvertet *11*

2.4 Tomma XML-element och attribut *11*

2.5 Valideringsregler för syntax *12*

3 Kryptering och signering av meddelande **14**

3.1 Certifikatspubliceringstjänsten *14*

3.2 Ordningsföljd avseende signering och kryptering *14*

3.3 Specifika krav på utgivare av signerings- och krypteringscertifikat *14*

3.4 Kryptering av nyttolast - konfidentialitet *15*

3.4.1 Sårbarhet i XML Encryption vid användning av blockchiffer utan integritetsskydd 15

3.4.2 Struktur för XML Encryption – EncryptedData 16

3.4.3 Parametersättning XHE 17

3.4.4 Parametersättning av XML Encryption 17

3.5 Signering av kuvert – undertecknande *18*

3.5.1 Struktur för XML Digital signature 19

3.5.2 Parametersättning av XML Digital Signature 20

4 Vägledning **22**

4.1 Vid behov av sammansatt XML-schema *22*

1 Inledning

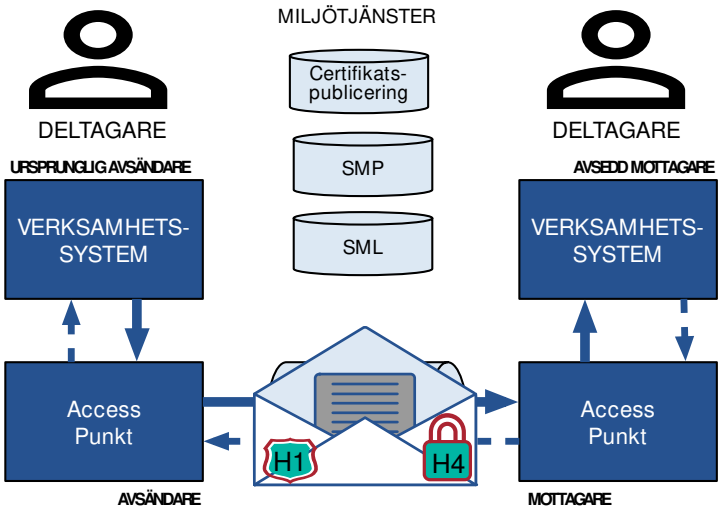
Beskrivning av innehållet i kuverteringsprofilen för XHE

Denna kuverteringsprofil beskriver teknologiskt hur ett meddelande paketeras och kuverteras på ett sätt som gör det möjligt att ha en rationell hantering i verksamheters system, applikationer och accesspunkter, vilka inte är beroende av ett meddelandes interna format och struktur. Kuverteringsprofilen ger dessutom möjligheter att signera kuvertet och att kryptera nyttolasten (verksamhetsmeddelandet).

Kuvertet baseras på en standard från OASIS BDXR-kommittén, vilken är samma organisation som står bakom standarderna SMP, BDXL/SML och AS4. XHE kuvertering är baserad på XML teknologin.

Kuverteringsprofilen är baserad på den begreppsmässiga och logiska meddelandemodellen som definierar vad ett meddelande och kuvert är och består av. Se ”Transportinfrastruktur – Beskrivning” för information om meddelande-, kuverterings-, och adresseringsmodeller.

Detta dokument beskriver enbart avvikelser från, restriktioner av och tillägg till de underliggande eDelivery specifikationer som ska användas för kommunikation inom transportinfrastrukturen. För detaljerad information hänvisas till underliggande specifikationer.



Figur 1 Illustration av kuverteringsprofilens fokus

Kuverteringsprofilen använder följande tekniska specifikationer:

- Exchange Envelope Header (XHE) Version 1.0, OASIS
- XML Encryption [XMLENC] (https://www.w3.org/TR/xmlenc-core1/).
- XML Signature [XMLDIGSIG] (https://www.w3.org/TR/xmldsig-core1/)

1.1 Dokumentstruktur

Detta dokument innehåller följande delar:

- Beskrivning av XHE-struktur för kuvertering av meddelande.
- Kryptering och signering av meddelande.
- Vägledning för inkludering av nyttolast-xsd i XHE

Regler är formaterade och identifierade enligt följande formatmall:

[a] Regeltext för första regeln a.

[b] Regeltext för andra regeln b.

En regel refereras unikt inom plattformen genom ”<dokument> ’-’ <sektion i dokument> ’.’ <regelidentitet>”. Exempel: ”plattform-2.1.a”.

En regel refereras lokalt inom dokument genom ”<sektion>’. ’<regelidentitet>”. Exempel: ”4.1.a”.

1.2 Målgrupper

Detta dokument syftar till att stödja följande intressenter i deras arbete, dess informationsbehov samt ge svar på vanligt förekommande frågeställningar.

Intressenter:

- IT-arkitekter och utvecklare
  - Som utvärderar, analyserar, designar, bygger, och testar programvaror.

1.3 Referenser

Referens till	Länk	Kommentar
Exchange Envelope Header (XHE) Version 1.0, OASIS	https://docs.oasis-open.org/bdxr/xhe/v1.0/xhe-v1.0-oasis.html	

XML Encryption [XMLENC]	https://www.w3.org/TR/xmlenc-core1/	
XML Signature [XMLDIGSIG]	https://www.w3.org/TR/xmldsig-core1/	
XML Signature Best Practices [SIGPRC]	https://www.w3.org/TR/xmldsig-bestpractices	
XML Path Language (XPath) 3.0 [XPATH]	https://www.w3.org/TR/xpath-30	
IANA Media Types [IANA]	https://www.iana.org/assignments/media-types/media-types.xhtml	

1.4 Federationsspecifika anpassningar

Detta dokument innehåller ett par regler, krav eller principer som en federation kan anpassa i federationsdeklarationen. Dessa anpassningspunkter är numrerade enligt formen [A1],[A2],[A3] osv.

2 Meddelandemodell enligt XHE

Ett meddelande med dess metadata och nyttolast kuverteras enligt det XML-baserade regelverket XHE utgiven av OASIS.

I det fall meddelandet signeras och krypteras i deltagarens verksamhetssystem/meddelandetjänst så måste meddelandekuvertering utföras där. I fall då signering och kryptering inte nyttjas kan meddelandekuvertering alternativt utföras i accesspunktsfunktionen.

2.1 Informationsmodell

Tabellen nedan beskriver de uppgifter som ett kuvert innehåller samt deras placering (XPATH) i syntaxen XHE. Mer detaljer kring strukturen på XHE beskrivs i kapitlet Syntaxmappning.

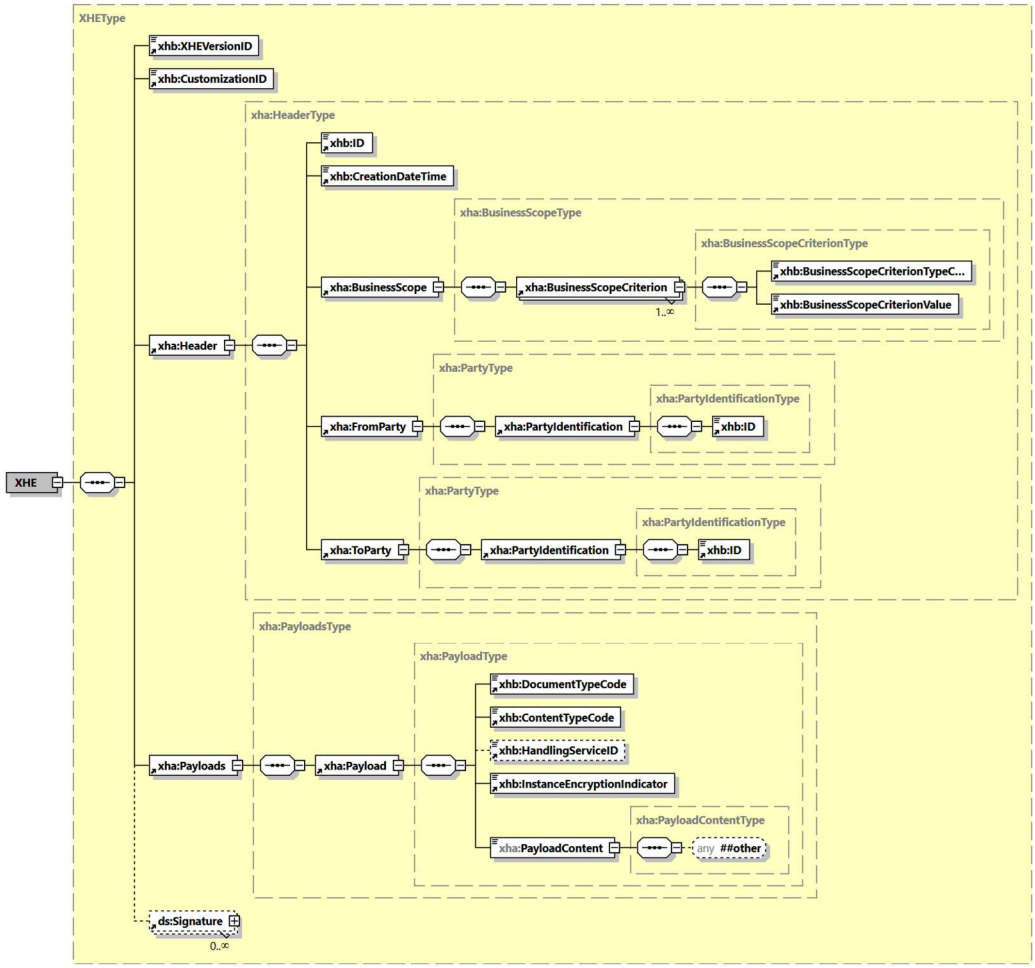
Informationsentitet	Beskrivning (placering i parentes)
Meddelandets identitet	Unik identifiering av meddelandet. *(/x:XHE/xha:Header/xhb:ID)*
Tid för utfärdande	Datum och klockslag då meddelandet skapades. *(/x:XHE/xha:Header/xhb:CreationDateTime)*
Ursprunglig avsändare	Den ursprungliga avsändarens identifierare. *(/x:XHE/xha:Header/xha:FromParty/xha:PartyIdentification/xhb:ID)*
Avsedd mottagare	Den avsedda mottagarens identifierare. *(/x:XHE/xha:Header/xha:ToParty/xha:PartyIdentification/xhb:ID)*
Uppgift om SMP DocumentIdentifier (Meddelandets typ)	En identifierare som används av avsändande accesspunkt för att hämta korrekt tjänstemetadata (teknisk mottagningsadress) från SMP. Identifiering av meddelandets typ och identifieringssystem (scheme). Se kodlista ”Typer av meddelanden” *(/x:XHE/xha:Header/xha:BusinessScope/xha:BusinessScopeCriterion[xhb:BusinessScopeCriterionTypeCode="DOCUMENTID"]/xhb:BusinessScopeCriterionValue)*
Uppgift om SMP ProcessIdentifier	En identifierare som används av avsändande accesspunkt för att hämta korrekt tjänstemetadata (teknisk mottagningsadress) från SMP och identifieringssystem (scheme). Se kodlista ”Typer av processer” *(/x:XHE/xha:Header/xha:BusinessScope/xha:BusinessScopeCriterion[xhb:BusinessScopeCriterionTypeCode="PROCESSID"]/xhb:BusinessScopeCriterionValue)*
Uppgift om Federation	En identifierare som informerar avsändande accesspunkt vilken federation som meddelandet ska förmedlas inom. Accesspunkten kan använda uppgiften för att validera att den företräder Deltagaren i aktuell federationen. Se kodlista ”Federationer”

	(/x:XHE/xha:Header/xha:BusinessScope/xha:BusinessScopeCriterion[xhb:BusinessScopeCriterionTypeCode="FEDERATIONID"]/xhb:BusinessScopeCriterionValue)
Nyttolast	Det verksamhetsmeddelande som kuverteras. Om kryptering end-to-end används är nyttolasten krypterad och strukturerad enligt XML Encryption. *(/x:XHE/xha:Payloads/xha:Payload/xha:PayloadContent)*
Nyttolastens tekniska format	Uppgift om den syntax/format som nyttolasten har (såsom EDIFACT, SDK XML-format, OASIS UBL XML-format osv) *(/x:XHE/xha:Payloads/xha:Payload/xhb:DocumentTypeCode)* *(/x:XHE/xha:Payloads/xha:Payload/xhb:ContentTypeCode)*
Signatur av meddelandet	Signatur av kuvertet. *(/x:XHE/Signature)*

2.2 Syntaxmappning

Kuvertet baseras på en XHE teknisk specifikation från OASIS BDXR-kommittén. Detaljer om elementens datatyper och namnrymder framgår i standarden och dess scheman [XHE].

[a]	Ett kuvert måste följa de kardinalitetsbegränsningar, instruktioner och regler som beskrivs i syntaxbindningen.



Kard	Elementnamn	Instruktion/regel
	XHE	Rotelement för kuvertet
1..1	• xhb:XHEVersionID	Versionen på XHE-schemat. Ska sättas till 1.0
1..1	• xhb:CustomizationID	Identifiering av denna profil av XHE-formatet. Ska sättas till ”urn:fdc:digg.se:edelivery:xhe:1”
1..1	• xha:Header	
1..1	• • xhb:ID	Meddelandets identitet ska vara globalt unikt och i UUID-format.
1..1	• • xhb:CreationDateTime	Datum och klockslag då meddelandet skapats. Datumformatet ska inkludera tidzon (där Z anges för UTC och övriga tidzoner indikeras med

		avvikelse från UTC enligt formatet ”+/- timmar”. Exempel: ”2021-03-24T17:22:10+01:00”
1..1	• • xha:BusinessScope	Business scope används för att identifiera den tjänstmetadata som är nödvändig för att skicka meddelandet.
5..5	• • • xha:BusinessScopeCriterion	Uppgifter om nycklar för identifiering av mottagningstjänst i SMP. Elementet består av två underliggande element som bildar värdepar (typkod+värde). Värdepar anges 5 gånger, en gång för respektive typ (DOCUMENTID, DOCUMENTID_SCHEME, PROCESSID, PROCESSID_SCHEME och FEDERATIONID.
1..1	• • • • xhb:BusinessScopeCriterionTypeCode	**DOCUMENTID** (motsvarnade DocumentIdentifier i SMP) **DOCUMENT_SCHEME** (motsvarar DocumentIdentifier/@scheme i SMP) **PROCESSID** (motsvarande ProcessIdentifier i SMP) **PROCESSID_SCHEME** (motsvarar ProcessIdentifier/@scheme i SMP) **FEDERATIONID** (indikerar vilken federation/SMP som ska användas för att identifiera mottagningstjänst)
1..1	• • • • xhb:BusinessScopeCriterionValue	Värdet som svarar mot typkoden.
1..1	• • xha:FromParty	Avsändande Deltagares identifiering
1..1	• • • xha:PartyIdentification	
1..1	• • • • xhb:ID	Identifierare för deltagaren @schemeID ska vara ”iso6523-actorid-upis”
1..1	• • xha:ToParty	Mottagande Deltagares identifiering
1..1	• • • xha:PartyIdentification	
1..1	• • • • xhb:ID	Identifierare för deltagaren @schemeID ska vara ”iso6523-actorid-upis”
1..1	• xha:Payloads	
1..1	• • xha:Payload	Struktur som beskriver nyttolast.

1..1	• • • xhb:DocumentTypeCode	Information om nyttolastens syntax/format. Om den kuverterade nyttolasten är i XML-format ska namnrymd och lokalt namn på rotelementet anges i formatet expanded qualified name, EQName [1][XPATH]. Exempel: Q{urn:oasis:names:specification:ubl:schema:xsd:ApplicationResponse-2}ApplicationResponse Om annat format än XML används ska kodvärdet överenskommas inom tillämpningsområdet.
1..1	• • • xhb:ContentTypeCode	Formatets innehållstyp (content type) enligt IANA Media Types koder [IANA]. För nyttolast som är XML-baserade ska koden application/xml anges.
0..1	• • • xhb:HandlingServiceID	En identifiering av mottagande Deltagares interna funktion/modul/tjänst som detta kuvert ska behandlas i. Värdet för detta element behöver överenskommas bilateralt mellan Deltagarna. Mottagande Deltagare måste ha en rutin för att omhänderta meddelanden även i det fall uppgiften saknas (men förväntats).
1..1	• • • xhb:InstanceEncryptionIndicator	Om nyttolasten är krypterad ska detta element sättas till ”true”, i annat fall ”false”. Kryptering är aktuellt då utökade säkerhetsmekanismer för end-to-end används.
1..1	• • • xha:PayloadContent	Nyttolasten i klartext eller i krypterad form genom användning av XML-Encryption. Nyttolast som inte är XML-baserad ska anges i Base64-kodad form.
0..n	• ds:Signature	Avsändande Deltagares signatur på meddelandekuvertet och dess innehåll (då utökade

[1] https://www.w3.org/TR/xpath-30/#doc-xpath30-EQName

		säkerhetsmekanismer för signering används).

2.3 Namnrymder som används i kuvertet

Följande XML namnrymder används i kuvertet. Prefix som används i syntaxbindningen (xha,xhb,ds) visar vilken namnrymd som respektive element hör hemma i.

Namnrymd	Kommentar
http://docs.oasis-open.org/bdxr/ns/XHE/1/ExchangeHeaderEnvelope	Rotelements namnrymd. Default i syntaxbeskrivningen
http://docs.oasis-open.org/bdxr/ns/XHE/1/AggregateComponents	Prefix ”xha” i syntaxbeskrivningen i kapitel 4.
http://docs.oasis-open.org/bdxr/ns/XHE/1/BasicComponents	Prefix ”xhb” i syntaxbeskrivningen
http://www.w3.org/2000/09/xmldsig#	Prefix ”ds” i syntaxbeskrivningen

2.4 Tomma XML-element och attribut

Ett element/attribut anses vara tomt om det inte innehåller något textinnehåll och inte några underliggande XML-element. Element och attribut som uteslutande innehåller ”whitespace” (mellanslag, tab-tecken eller radbrytning) är också att betrakta som tomt.

[a] Tomma element och attribut ska inte anges.

Exempel på tomma element:

```
<xhb:HandlingServiceID>    </xhb:HandlingServiceID>
<xhb:HandlingServiceID></xhb:HandlingServiceID>
<xhb:HandlingServiceID />
```

Exempel på tomt attribut:

```
< xhb:ID schemeID="">
```

2.5 Valideringsregler för syntax

Denna kuverteringsprofil använder XML-standarden XHE inklusive dess XSD-scheman.

[a] Ett kuvert som inte validerar korrekt gentemot XHE XSD-scheman är inte att betrakta som följsamt gentemot denna specifikation.

En uppsättning valideringsregler finns definierade för att kontrollera att ett kuvert är följsamt gentemot denna specifikation. Reglerna finns även representerade som schematron-regler för maskinell och automatisk validering.

[b] Ett kuvert som bryter mot någon av valideringsreglerna är inte att betrakta som följsamt gentemot denna specifikation.

Regel-ID	Regel	Allvarlighet
R1-XHE	Endast XML-element och attribut som angivits i denna specifikation får användas	Fatal
R2-XHE	Tomma element eller attribut får inte anges	Fatal
R3-XHE	CustomizationID måste ha korrekt värde enligt kodlista	Fatal
R4-XHE	Ett BusinessScopeCriterionTypeCode ska anges med koden DOCUMENTID	Fatal
R5-XHE	Ett BusinessScopeCriterionTypeCode ska anges med koden DOCUMENTID_SCHEME	Fatal
R6-XHE	Ett BusinessScopeCriterionTypeCode ska anges med koden PROCESSID	Fatal
R7-XHE	Ett BusinessScopeCriterionTypeCode ska anges med koden PROCESSID_SCHEME	Fatal
R8-XHE	Ett BusinessScopeCriterionTypeCode ska anges med koden FEDERATIONID	Fatal
R9-XHE	XHEVersionID ska ha värde 1.0	Fatal

R10-XHE	ID för FromParty ska ha attributet 'iso6523-actorid-upis'	Fatal
R11-XHE	ID för ToParty ska ha attributet 'iso6523-actorid-upis'	Fatal
R12-XHE	PayloadContent ska innehålla elementet 'EncryptedData' om InstanceEncryptionIndicator är 'true'	Fatal
R13-XHE	PayloadContent ska inte innehålla elementet 'EncryptedData' om InstanceEncryptionIndicator är 'false'	Fatal
R14-XHE	Element och attribut ska anges i enlighet med den tillåtna kardinalitet som framgår i syntaxmappningen[2].	Fatal

[2] Ett element som exempelvis är upprepningsbart i XHE-standarden kan enligt denna specifikation vara begränsat till att inte få anges mer än en gång.

3 Kryptering och signering av meddelande

Denna kuverteringsprofil har stöd för försändelser (nyttolast) som krypterats och signerats mellan Deltagarna. Användning av kryptering och signering mellan Deltagare kräver att båda parter är förberedda och att det finns överenskommelse (mellan parterna eller inom federationen som helhet) om att dessa utökade säkerhetsmekanismer ska användas.

3.1 Certifikatspubliceringstjänsten

Certifikat för signering och kryptering finns tillgängliga genom slagning i Certifikatspubliceringstjänsten eller genom bilaterala överenskommelser mellan avsändande och mottagande Deltagare.

3.2 Ordningsföljd avseende signering och kryptering

Då signering och kryptering används ska kryptering av nyttolasten göras först och därefter signeras meddelandekuvertet i sin helhet.

På motsvarande sätt ska mottagaren kontrollera meddelandekuvertets signatur först och därefter dekryptera den kuverterade nyttolasten.

Om signaturen inte validerar korrekt ska dekryptering inte utföras.

3.3 Specifika krav på utgivare av signerings- och krypteringscertifikat

Denna specifikation ställer inte några formkrav (nyckellängd, utgivare osv) på certifikaten som används.

En federationsoperatör kan i sin federationsdeklaration ställa följande specifika krav avseende certifikat:

Anpassning [A1]:
Formkrav på certifikat (nyckellängd, användning av attribut osv)

Anpassning [A2]:
Certifikatsutgivare som kan användas

Anpassning [A3]:
Om det ska vara separata certifikat för signering och kryptering eller om samma certifikat ska användas i båda syftena.

3.4 Kryptering av nyttolast - konfidentialitet

När kryptering av meddelande används ska det göras i enlighet med W3C-standarden ”XML Encryption” [XMLENC].

Standarden beskriver både processen för att kryptera XML-element och hur krypterade data representeras i XML-format. Schemat för XML Encryption är inte inkluderat i XHE utan kan laddas ner separat.

Kryptering och dekryptering görs med den avsedda mottagarens publika respektive privata nyckel. Mottagarens publika nyckel måste vara tillgänglig för den som utför krypteringen.

Krypteringen genomförs genom att en tillfällig meddelandenyckel skapas och används för att kryptera meddelandet. Denna meddelandenyckel krypteras sedan med mottagarens publika nyckel. Det innebär att det finns två block av krypterat data i XML-strukturen, den krypterade meddelandenyckeln och det krypterade meddelandet. När mottagaren ska läsa meddelandet används mottagarens privata nyckel för att dekryptera meddelandenyckeln som i sin tur används för att dekryptera det krypterade meddelandet.

I klartextform placeras nyttolasten i XHE-elementet PayloadContent. Detta element är av typen xsd:any vilket innebär att det inte är bundet till ett specifikt typ av XML-meddelandeformat. När nyttolasten i klartext krypteras enligt XML Encryption så blir resultatet en ny XML-struktur som innehåller information om nycklar, krypteringsalgoritmer mm. Denna XML-struktur, som har rotelement EncryptedData, ersätter klartextnyttolasten i PayloadContent.

3.4.1 Sårbarhet i XML Encryption vid användning av blockchiffer utan integritetsskydd

En sårbarhet är identifierad då icke integritetsskyddade blockchiffer-algoritmer används tillsammans med XML Encryption. Sårbarheten kan utnyttjas genom en så kallad ”oracle attack” där en anropande applikation upprepande gånger och på ett systematiskt sätt skickar ett krypterat meddelande för dekryptering där det krypterade meddelandet modifieras med mycket små förändringar varje gång. Genom att utvärdera

svarsmeddelanden som returneras från den dekrypterande applikationen kan anropande applikation till slut dra slutsatser om utseendet av den klartext som krypterades. Det finns ett antal motåtgärder för denna typ av sårbarhet varav några används i denna specifikation.

1. Innan dekryptering görs så kontrolleras meddelandets signatur. Om signaturvalidering misslyckas så förkastas meddelandet och inget försök att utföra dekryptering på meddelandet görs.
2. Då meddelandet inte dekrypteras vid felaktig signatur så returneras heller ingen felkod som informerar om resultat av dekryptering som kan användas vid en ”oracle attack”.

Problemet med blockchiffer-algoritmer och XML Encryption kan lösas genom att använda krypto med integritetsskydd, så som AES GCM, som introducerades i XML Encryption version 1.1. När AES GCM används utökas krypteringsskyddet med ett integritetsskydd som kontrollerar att krypterade data inte förändrats innan dessa dekrypteras. På så sätt uppnås samma skydd vid kryptering som uppnås genom att som ovan signera krypterade data och att kontrollera denna signatur innan dekryptering.

Då Microsofts ”.Net Framework” inte har stöd för att hantera XML Encryption 1.1-standarden gör DIGG idag bedömningen att AES GCM introducerar stor komplexitet för implementationer som baseras på Microsofts ramverk. Vidare bedömer DIGG att de motåtgärder som finns inbyggda i eDelivery plattformen gör att blockchiffer-algoritmer kan användas i detta sammanhang.

3.4.2 Struktur för XML Encryption – EncryptedData

För information om XML namnrymder, se [XMLENC].

Kard	**Elementnamn**	**Kommentar**
	xenc:EncryptedData	Rotelement för den krypterade nyttolasten
1..1	• xenc:EncryptionMethod/@Algorithm	Krypteringsmetod
1..1	• ds:KeyInfo	Information om sessionsnyckel
1..1	• • xenc:EncryptedKey	
1..1	• • • xenc:EncryptionMethod/@Algorithm	Krypteringsalgoritm av sessionsnyckel
1..1	• • • ds:KeyInfo	
1..1	• • • • ds:X509Data	
1..1	• • • • • ds:X509Certificate	Certifikat som använts vid kryptering av sessionsnyckel
1..1	• • • xenc:CipherData	

1..1	• • • • xenc:CipherValue	Krypterad sessionsnyckel
1..1	• xenc:CipherData	
1..1	• • xenc:CipherValue	Krypterad nyttolast

3.4.3 Parametersättning XHE

Elementet Payload innehåller vissa metadata om det meddelande som skickas.

[a] När kryptering används ska det indikeras genom att elementet *InstanceEncryptionIndicator* sätt till ”`true`”

Exempel:

```
<InstanceEncryptionIndicator>true</InstanceEncryptionIndicator>
```

3.4.4 Parametersättning av XML Encryption

Parameter/algoritm	**Värde**
Typ av objekt som krypteras	http://www.w3.org/2001/04/xmlenc#Element *eller* http://www.w3.org/2001/04/xmlenc#Content
Krypteringsalgoritm för sessionsnyckel	http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p
Krypteringsalgoritm för nyttolasten	http://www.w3.org/2001/04/xmlenc#aes256-cbc

3.4.4.1 Typ av krypterat innehåll

[a] När nyttolasten som krypteras är XML-baseras ska innehållstypen anges till ”Element”.

Exempel:

```
<xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element">
```

[b] När nyttolasten som krypteras inte är XML-baserad ska innehållstypen anges till ”Content”.

Exempel:

```
<xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#E
lement">
```

3.4.4.2 Kryptering av sessionsnyckel

Sessionsnyckeln krypteras med hjälp av mottagarens publika nyckel.

[a] Algoritmen ”`rsa-oaep-mgf1p`” ska användas för kryptering av den tillfälliga symmetriska sessionsnyckeln.

Exempel:

```
<xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/
xmlenc#rsa-oaep-mgf1p">
```

3.4.4.3 Kryptering av meddelandet

Meddelandet krypteras med hjälp av sessionsnyckeln.

[a] Algoritmen ”`AES256`” ska användas för kryptering av meddelandet.

Exempel:

```
<xenc:EncryptionMethod Algorithm="
http://www.w3.org/2001/04/xmlenc#aes256-cbc">
```

3.4.4.4 Inkludering av X509 Certifikat

Syftet med att inkludera certifikatet är att förenkla hantering vid certifikatsbyten samt hantering av arkiverade meddelanden.

[a] Den publika delen av det certifikat som använts vid kryptering ska inkluderas i PEM-format (BASE64-kodat).

Exempel:

```
<KeyInfo>
  <X509Data>
    <X509Certificate>MIIEY ... nk=</X509Certificate>
(Exemplet är nedkortat)
  </X509Data>
</KeyInfo>
</Signature>
```

3.5 Signering av kuvert – undertecknande

När signering av meddelande används ska det göras i enlighet med W3C-standarden ”XML Signature” [XMLDIGSIG].

Standarden beskriver både processen för att skapa elektronisk signatur och hur den representeras i XML-format.

W3C har även publicerat ”Best Practices” [SIGPRC] som ger goda råd för behandlingen av digitala signaturer.

I XML Signature beskrivs flera olika sätt knyta signaturen till den nyttolast som signeras. Denna kuverteringsprofil nyttjar metoden ”Enveloped Signature” vilket innebär att signaturen läggs in som en del i XML-strukturen.

Vid validering måste hela XML-strukturen överensstämma med den XML som signerades. För att försäkra sig om att XML-strukturen återges på samma sätt både vid signering och vid validering används en kanoniseringstransformering och vid validering exkluderas signaturen från XML-strukturen.

[a]	Mottagaren ska kontrollera signeringscertifikatets giltighet och att det inte är spärrat.
[b]	Om Certifikatspubliceringstjänsten används så ska signeringscertifikatet också jämföras med det certifikat som publicerats för ändamålet.

Jämförelsen görs för att försäkra sig om att signeringscertifikatet som använts är det som är registrerat för Deltagaren i Certifikatspubliceringstjänsten.

3.5.1 Struktur för XML Digital signature

Kard	**Elementnamn**	**Kommentar**
	ds:Signature	Rotelement för signaturen
1..1	• ds:SignedInfo	
1..1	• • ds:CanonicalizationMethod/@Algorithm	Kanoniseringsmetod
1..1	• • ds:SignatureMethod/@Algorithm	Signeringsalgoritm
1..1	• • ds:Reference/@URI	
1..1	• • • ds:Transforms	
1..1	• • • • ds:Transform/@Algorithm	Transformeringsmetod
1..1	• • • ds:DigestMethod/@Algorithm	Hashningsmetod
1..1	• • • ds:DigestValue	
1..1	• ds:SignatureValue	Signaturen
1..1	• ds:KeyInfo	
1..1	• • ds:X509Data	
1..1	• • • ds:X509Certificate	Certifikat som använts för signatur

3.5.2 Parametersättning av XML Digital Signature

Följande parametrar ska användas vid användning av [XMLDIGSIG]

Parameter/algoritm	**Värde**
Kanoniseringsmetod	Någon av de som metoder som är obligatoriska att stödja enligt [XMLDIGSIG]
Signeringsalgoritm	http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
Transformeringsmetod	http://www.w3.org/2000/09/xmldsig#enveloped-signature
Hashningsmetod	http://www.w3.org/2001/04/xmlenc#sha256

3.5.2.1 Kanoniseringsalgoritm

[a] Kanoniseringsmetod ska nyttja någon av de algoritmer som anges som obligatoriska att stödja i [XMLDIGSIG]

Exempel:

```
<CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
```

3.5.2.2 Signeringsalgoritm

[a] Signeringsalgoritm ska vara ”RSAwithSHA256”.

Exempel:

```
<SignatureMethod
Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256"/>
```

3.5.2.3 Transformeringsmetod

[a] Signaturen ska inkluderas i kuvertets XML-struktur genom att följa ”Enveloped”-metoden.

Exempel:

```
<Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/>
```

3.5.2.4 Hashningsmetod

[a] Hashvärdet ska beräknas utifrån XML-strukturen enligt algoritmen ”SHA256”.

Exempel:

```
<DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
```

3.5.2.5 Inkludering av X509 Certifikat

[a] Den publika delen av det certifikat som använts vid signering ska inkluderas i PEM-format (BASE64-kodat).

Exempel:

```
<KeyInfo>
  <X509Data>
    <X509Certificate>MIIEY ... nk=</X509Certificate>
(Exemplet är nedkortat)
  </X509Data>
</KeyInfo>
</Signature>
```

4 Vägledning

4.1 Vid behov av sammansatt XML-schema

Vissa applikationer kan vara hjälpta av att ha ett sammansatt XML-schema som inkluderar både kuvert och meddelandeformat. XHE-schemat är generiskt så till vida att vilken typ av XML-meddelande som helst kan förpackas i det (genom att elementet PayloadContent är av typen xsd:any). Om en implementatör vill anpassa XHE-schemat för att inkludera ett specifikt meddelandeformat kan man göra på följande sätt:

- Ladda ner XSD-schema från https://docs.oasis-open.org/bdxr/xhe/v1.0/xhe-v1.0-oasis.html
- Lägg till en XSD Import i schemamodulen *XHE-PayloadContentDataType-1.0.xsd* som pekar ut aktuellt XML Schema för verksamhetsmeddelandet.
- Ersätt den typdefinition som ursprungligen anger xsd:any mot en referens till rotelementet i det importerade schemat.

Exempel:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://docs.oasis-open.org/bdxr/ns/XHE/1/AggregateComponents"
xmlns:tns="urn:riv:infrastructure:messaging:MessageWithAttachments:2"
targetNamespace="http://docs.oasis-
open.org/bdxr/ns/XHE/1/AggregateComponents" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.1">
  <!--import here all payload schemas-->
  <!-- ===== Type Declaration ===== -->
  <xsd:import
namespace="urn:riv:infrastructure:messaging:MessageWithAttachments:2"
schemaLocation="../sdk-document-message-2.0_2020-04-08/sdk-document-
message/core_components/infrastructure_messaging_MessageWithAttachments_2.0
.xsd"/>
  <xsd:complexType name="PayloadContentType" mixed="true">
    <xsd:sequence>
      <xsd:element ref="tns:messagePayload"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```