ECR BRÁNA

ECR obálka 3.0

Autor:	AQUASOFT spol. s r.o., Jergon Roman;Radek Němec

Poslední aktualizace:	16.8.2016
Jméno souboru:	ECRbranaRozhraniECRObalka3v12
Počet stran:	15

HISTORIE DOKUMENTU

Verze	Datum	Podpis
1.0	4. 9. 2015	Úvodní verze
1.1	13. 11. 2015	Do kapitoly 3.2 přidána poznámka o možnosti použít el. razítko pro potřeby deklaranta
1.2	16. 8. 2016	Opraveny chyby v příkladech (kapitoly 3.2.1, 3.2.2, 3.2.4)

OBSAH

1. ÚVOD

ECR brána je systém, který od roku 2002 slouží ke komunikaci celní správy s externími subjekty, převážně s deklaranty. Systém byl průběžně rozšiřován a modernizován, aby pružně reagoval na potřeby Celní správy. K jeho poslední zásadní modernizaci došlo v průběhu let 2015 - 2016 s cílem vyhovět novým legislativním požadavkům a navýšit přenosovou kapacitu.

Tato modernizace ECR brány sebou přináší i novou verzi ECR obálky, která rozšiřuje nabízené funkce pro deklarantskou veřejnost.

Po přechodné období lze i nadále používat stávající verzi ECR obálky 2.0 bez jakýchkoliv změn, nelze v ní však samozřejmě využívat nově nabízené funkce.

2. KOMUNIKAČNÍ ZPRÁVA ECR OBÁLKA 3.0

ECR obálka je hlavní komunikační zpráva na ECR bráně a slouží k uložení informací o přenášené zprávě, které jsou důležité pro její správné nasměrování na následnou centrální aplikaci Celní správy, ověření zabezpečení a pro sledování jejího toku systémem.

Veškerá data určená pro průchod ECR bránou musí být do této obálky zapouzdřena.

2.1. ZMĚNY VŮČI VERZI 2.0

Nová verze ECR obálky je zaměřena na splnění aktuálních legislativních požadavků ohledně použití elektronických podpisů a přidání funkčnosti pro přenos přiložených binárních souborů, které nebudou podepsány.

Přehled hlavních změn ECR obálky 3.0 oproti verzi 2.0:

- Jmenný prostor kořenového elementu je změněn na http://www.cs.mfcr.cz/schemas/EcrEnvelope/V_3.0
- Ruší se atribut Hlavicka/@VerzeObalky (verze obálky je uvedena ve jmenném prostoru)
- Ruší se element SmimeZprava
- V elektronickém podpisu je vyžadováno použití hashovacích algoritmů SHA2 a rozšíření XAdES-BES, viz kapitola 3.2
- Nový nepovinný element „Attachments“
  - Obsah této části nesmí být podepsán elektronickým podpisem
  - V této části jsou opakovatelné elementy BinaryAttachment (binární příloha), XmlAttachment (XML příloha)
  - Binární příloha i XML příloha mají povinný unikátní identifikátor (uvedený v atributu Id příslušného XML elementu)
  - Přílohy mohou být šifrovány (dle nastavení požadavků domény)
- Konec podpory šifrovacího algoritmu 3DES
- Pojmenování všech prvků ECR obálky v anglickém jazyce

2.2. SYSTÉM KONTROL

Na příchozí ECR obálku jsou aplikovány kontroly specifické dle komunikační domény. Patří mezi ně zejména:

- Kontrola proti XML schématu ECR obálky pro konkrétní doménu.
- Logická kontrola vyplňování ECR obálky proti přenášeným datům.
- Logická kontrola vyplněných údajů proti registrům komunikačních subjektů.
- Rozšifrování a ověření zaručeného elektronického podpisu.

Dále popsaný způsob vyplňování je obecně platný, nicméně pro jednotlivé komunikační domény může být zúžen a zpřísněn.

2.3. STRUKTURA OBÁLKY

Základní přehled o elementech dává přiložený HTML dokument a XSD šablony. V dalších odstavcích jsou pak jednotlivé elementy popsány podrobněji.

2.3.1 Element „Header“

Tento element je povinný a obsahuje hlavní informace o agendě a datové výměně. Každá instance ECR obálky musí mít vygenerovaný unikátní GUID (UUID) identifikátor v atributu „EnvelopeGuid“ a uvedenu komunikační doménu v atributu „Domain“.

2.3.2 Element „MessageMetadata“

Tento povinný element zapouzdřuje informace o konkrétní přenášené zprávě nebo datové výměně. Udává se „Type“ datové výměny, což je u XML komunikací zpravidla jméno kořenového elementu přenášené zprávy. Dále se uvádí volitelně až dva identifikátory z přenášené zprávy, které mohou mít význam pro směrování zprávy. Způsob vyplňování elementů „PrimaryID“ a „SecondaryID“ je určen doménou a typem výměny, nicméně ve většině domén se požaduje uvést alespoň jeden z těchto identifikátorů. Do budoucna se počítá s tím, že se mohou vyskytnout scénáře, ve kterých bude nutné uvádět v obálce ještě další atributy - pro ně je určena volitelná struktura „Attributes/Attribute“.

2.3.3 Element „AdditionalInformation“

Tento nepovinný element je v současné době nepoužíván a rezervován pro případná rozšíření ECR obálky v budoucích doménách.

2.3.4 Element „Participants“

Povinný element, obsahuje informace o všech zúčastněných komunikačních partnerech. Každý zúčastněný komunikační partner musí při průchodu zprávy jeho systémem přidat jeden podelement „Participant“ a uvést svoji roli do atributu „Role“, svoji identifikaci do atributu „Identification“, datum a čas průchodu zprávy jeho systémem do atributu „DateAndTime“ a informace o programovém modulu, který obálku zpracovával, do atributů „ApplicationName“ a „ApplicationVersion“.

Tři nejčastější současné role jsou „declarant“ (vyplní deklarant, resp. jeho SW), „operator“ (vyplní např. VAN operátor) a „grc“ (vyplní ECR brána). Všechny role kromě „grc“ musí povinně vyplnit všech 5 atributů kromě volitelného „ScenarioGUID“, což je GUID, u něhož ECR brána zaručuje, že ho v nezměněné podobě vrátí v odpovědi ve stejné sekci. ECR brána vyplňuje pouze roli a datum a čas a volitelně dle domény identifikaci scénáře. V odpovědi ECR brány se všechny sekce vracejí, informace o programovém modulu se však u nich zpravidla již neuvádí.

2.3.5 Datový přenášený obsah

Příchozí ECR obálka musí obsahovat právě jeden z elementů „XmlMessage“ nebo „BinaryMessage“. V těchto elementech je samotný přenášený datový obsah. Nepovinně může být použit také element „Attachments“.

2.3.5.1 Element „XmlMessage“

Element se uvede, mají-li přenášená data formát XML. Ta jsou pak přímo vložena jako podelement elementu Data. Zde je důležité, aby kořenový element přenášené zprávy měl uveden svůj vlastní defaultní jmenný prostor, popřípadě měl defaultní jmenný prostor nastavený na prázdnou hodnotu (xmlns=““). Je to z důvodu zabránění interpretace zprávy v kontextu jmenného prostoru ECR obálky.

Mají-li být data šifrována, umístí se do struktury XML encryption do elementu „EncryptedData“. Do elementu „Signature“ je možné uvést zaručený elektronický podpis. Obsah elementů „EncryptedData“ a „Signature“ je určen doporučeními W3C a popsán v kapitole 3.

Do elementu Signature/Object/QualifyingProperties/SignedProperties/SignedDataObjectProperties/DataObjectFormat/MimeType elektronického podpisu se vždy uvede hodnota „text/xml“.

2.3.5.2 Element „BinaryMessage“

Element se uvede, mají-li přenášená data binární nebo jinou než XML povahu. Ta jsou pak přímo zakódovaná v base64 jako textová hodnota elementu Data. Mají-li být data šifrována, umístí se do elementu

„EncryptedData“ struktury XML encryption. Do elementu „Signature“ je možné uvést zaručený elektronický podpis. Obsah elementů „EncryptedData“ a „Signature“ je určen doporučeními W3C a popsán v kapitole 3.

Do elementu Signature/Object/QualifyingProperties/SignedProperties/SignedDataObjectProperties/DataObjectFormat/MimeType elektronického podpisu se uvede hodnota, která vyjadřuje formát přenášených binárních dat (po dekódování z base64), například „application/x-gzip“.

2.3.5.3 Element „Attachments“

Tento nepovinný element a jeho podelementy „BinaryAttachment“ a „XmlAttachment“ slouží k přenášení příloh, jejichž data nejsou zahrnuta do elektronického podpisu. Proto lze při použití tohoto elementu podepisovat element „XmlMessage“ nebo „BinaryMessage“ (uvedené v předchozích kapitolách) pouze pomocí způsobů uvedených v kapitolách 3.2.3 a 3.2.5.

Element musí obsahovat alespoň jeden z elementů „BinaryAttachment“ nebo „XmlAttachment“. Každý z nich je možné uvést opakovaně a každý z nich musí mít v atributu „ID“ uveden jednoznačnou identifikaci v rámci ECR obálky.

Každou přílohu lze zaslat buďto zašifrovaně (v příslušné příloze použit podelement „EncryptedData“), nebo v otevřeném tvaru (podelement „Data“).

2.3.6 Element „Error“

Tento element nesmí být uveden v příchozí zprávě. Je vždy generován ECR bránou a informuje o některém z problémů, ke kterým může při příjmu a kontrole ECR obálky dojít. Odpovídá-li ECR brána zprávou ECR obálka s uvedeným elementem „Chyba“, není uveden žádný z datových elementů z kapitoly 2.3.5.

Element má vždy uveden číselný kód chyby (atribut „Code“) a její zdroj (atribut „Type“). Volitelně je obsažen popis chyby (atribut „Description“) a u některých chyb i další doprovodná informace v podobě chybných dat v textovém elementu „Data“. Rovněž volitelně může být přítomen atribut „EnvelopeGUID“, jenž odkazuje na identifikaci ECR obálky, k níž je chyba hlášena.

Seznam chyb, které může ECR brána aktuálně vracet, je uveden v kapitole 4.

3. FORMÁTY ZABEZPEČENÍ DAT ECR OBÁLKY

3.1. ŠIFROVÁNÍ

Šifrování v ECR obálce se plně řídí doporučením W3C XML Encryption (http://www.w3.org/TR/xmlenc-core/). Na rozdíl od tohoto standardu však není algoritmus 3DES.

Je-li datový obsah šifrován, použije se jako vstup pro XML encryption element „Data“. Ten je následně nahrazen elementem EncryptedData už ve jmenném prostoru příslušného W3C doporučení. Příklad použitého šifrování je na následujícím zkráceném XML fragmentu:

```
<XmlMessage SignatureContext="datacontent">
    <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element"
xmlns="http://www.w3.org/2001/04/xmlenc#">
      <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc" />
      <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
        <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
          <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5" />
          <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
            <X509Data>
              <X509IssuerSerial>
                <X509IssuerName>CN=PostSignum Qualified CA 2, O="Česká pošta, s.p. [IČ 47114983]",
C=CZ</X509IssuerName>
                <X509SerialNumber>1791234</X509SerialNumber>
              </X509IssuerSerial>
            </X509Data>
          </KeyInfo>
          <CipherData>
            <CipherValue>......</CipherValue>
          </CipherData>
        </EncryptedKey>
      </KeyInfo>
      <CipherData>
        <CipherValue>......</CipherValue>
      </CipherData>
    </EncryptedData>
  </XmlMessage>
```

3.2. ZARUČENÝ ELEKTRONICKÝ PODPIS

Zaručený elektronický podpis se v ECR obálce řídí doporučením W3C XML Advanced Electronic Signatures (HTTP://WWW.W3.ORG/TR/2003/NOTE-XADES-20030220/), který je rozšířením doposud používaného XML Signature (HTTP://WWW.W3.ORG/TR/2002/REC-XMLDSIG-CORE-20020212/).
Podpis musí splňovat formát XAdES-BES dle specifikace ETSI TS 101 903 XAdES v1.4.1 (HTTP://URI.ETSI.ORG/01903/V1.4.1/TS_101903V010401P.PDF). Podpis musí být vytvořen jedním z podporovaných SHA-2 algoritmů (SHA-256, SHA-384, SHA-512) a navíc splňovat požadavky dle následujících dokumentů:

- 2011/130/EU: Rozhodnutí Komise ze dne 25. února 2011, kterým se stanoví minimální požadavky na přeshraniční zpracování dokumentů elektronicky podepsaných příslušnými orgány podle směrnice 2006/123/ES Evropského parlamentu a Rady o službách na vnitřním trhu
- 2014/148/EU: Prováděcí rozhodnutí Komise ze dne 17. března 2014, kterým se mění rozhodnutí 2011/130/EU, kterým se stanoví minimální požadavky na přeshraniční zpracování dokumentů elektronicky podepsaných příslušnými orgány podle směrnice 2006/123/ES Evropského parlamentu a Rady o službách na vnitřním trhu
- ETSI TS 103 171 V2.1.1 (2012-03) - odkazován z 2014/148/EU

Použití zaručeného elektronického podpisu nad otevřeným XML přináší jako výhodu fakt, že - pokud nejsou šifrována – jsou data čitelná. To usnadňuje případné dohledávání problémů. Při používání XML signature je nutné mít na paměti zejména následující fakta:

- Je vždy nutné provést kanonizaci XML a uvést ji ve struktuře podpisu
- Předmětem podpisu je vždy element SignatureInfo, který obsahuje miniaturu (SHA2 hash) podepisovaného dokumentu.

- Znaky typu mezera, tabulátor a formátování konce řádku jsou pro XML signature významnými znaky, kanonizace je neodstraňuje a jejich změna vede k porušení podpisu. Mezi procesem podepsání zprávy a ověření podpisu tedy nesmí dojít například k přeformátování XML!
- Pojmenování jmenných prostorů a jejich hodnoty jsou pro XML signature významnými znaky, kanonizace s nimi nijak nepracuje a nesmí být měněny. Je také nutné vést v patrnosti, že přidání pojmenovaného jmenného prostoru se promítne v DOM reprezentaci do všech podřízených elementů, které tento jmenný prostor zdědí, což může narušit podpis, přestože nadřazený element nebyl předmětem podpisu.
- V ECR bráně je vždy předlohou pro výpočet miniatury samotná přenášená zpráva uvnitř elementu „Data“, nikoliv element „Data“ samotný.

Zaručený elektronický podpis v ECR obálce může pro potřeby odesílatele obsahovat kvalifikované časové razítko, které musí být uvedeno v části UnsignedSignatureProperties (dle specifikace XAdES-T, viz ETSI TS 101 903 V 1.4.1).

Následuje popis možných typů provedení elektronického podpisu v ECR obálce.

3.2.1 Typ Enveloping

Tento typ znamená, že přenášená zpráva není umístěna jako potomek elementu Data (ten zůstává prázdný), ale je obsažena přímo jako definovaný objekt uvnitř struktury „Signature“, která je potomkem elementu „XmlMessage“ nebo „BinaryMessage“. Při tomto použití nemusí být předepsána žádná transformace, ale výsledek nelze šifrovat. Příklad uvádí následující fragment:

```
<XmlMessage>
    <Data/>
    <Signature Id="signature" xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
        <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
        <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
        <Reference Id="xadesReference" URI="#dataID">
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>SdDJFqPLeG12V22sQOy0LqEv7eR/DX+rihg09e5FJWA=</DigestValue>
        </Reference>
        <Reference URI="#xadesSignedProperties"
Type="http://www.w3.org/2000/09/xmldsig#SignatureProperties">
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>+yU8sk4agb71zQng2k1wOlZlQZU6ZYpYSwShnXOFhsQ=</DigestValue>
        </Reference>
      </SignedInfo>
      <SignatureValue>MOml…</SignatureValue>
      <KeyInfo>
        <X509Data>
          <X509Certificate>…</X509Certificate>
        </X509Data>
      </KeyInfo>
      <Object Id="dataID">
        <CZ515A xmlns="">
          <H>
            <QH16>0</QH16>
            <H02>1578142511544H569C7Z</H02>
            …
        </CZ515A>
      </Object>
      <Object>
        <QualifyingProperties Target="signature" xmlns="http://uri.etsi.org/01903/v1.3.2#">
          <SignedProperties Id="xadesSignedProperties">
            <SignedSignatureProperties>
              <SigningTime>2015-09-03T10:57:10.5520816Z</SigningTime>
              <SigningCertificate>
                <Cert>
                  <CertDigest>
                    <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"
xmlns="http://www.w3.org/2000/09/xmldsig#" />
                    <DigestValue xmlns="http://www.w3.org/2000/09/xmldsig#">…</DigestValue>
                  </CertDigest>
                  <IssuerSerial>
                    <X509IssuerName xmlns="http://www.w3.org/2000/09/xmldsig#">CN=PostSignum
Qualified CA 2, O="Česká pošta, s.p. [IČ 47114983]", C=CZ</X509IssuerName>
                    <X509SerialNumber
xmlns="http://www.w3.org/2000/09/xmldsig#">1791234</X509SerialNumber>
```

```
                </IssuerSerial>
              </Cert>
            </SigningCertificate>
          </SignedSignatureProperties>
          <SignedDataObjectProperties>
            <DataObjectFormat ObjectReference="#xadesReference">
              <MimeType>text/xml</MimeType>
            </DataObjectFormat>
          </SignedDataObjectProperties>
        </SignedProperties>
      </QualifyingProperties>
    </Object>
  </Signature>
</XmlMessage>
```

3.2.2 Typ Enveloped, mimo zprávu, SignatureContext = "envelope"

Tento typ znamená, že je podpis vytvořen v kontextu celé ECR obálky (v době podpisu již byla sestavena) a element Signature je umístěn jako potomek elementu „XmlMessage“ nebo „BinaryMessage“. Vždy musí být uvedeny transformace „enveloped-signature“ a „xpath“, která určuje cestu k podepisovaným datům.

Vzhledem k tomu, že použití xpath transformace je pro XML zprávy se složitější strukturou velmi výpočetně náročné, je tento typ podpisu striktně omezen jen na typ ECR obálky s elementem BinarniZprava.

```
<BinaryMessage>
    <Data>……</Data>
    <Signature Id="signature" xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
        <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
        <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
        <Reference URI="" Id="xadesReference">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
            <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
              <XPath xmlns:transns="http://www.cs.mfcr.cz/schemas/EcrEnvelope/V_3.0">ancestor-or-
self::transns:Data[/transns:EcrEnvelope/transns:BinaryMessage/transns:Data=.]</XPath>
            </Transform>
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>lt3zXAk3TR9TvKdZcmWw4s/V6GYG9d9jWzeAK2EOW74=</DigestValue>
        </Reference>
        <Reference URI="#xadesSignedProperties"
Type="http://www.w3.org/2000/09/xmldsig#SignatureProperties">
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>VDFziXXHFKWmKtxKvVFjANK+lJEBDW1Oj9wRsnjzPOk=</DigestValue>
        </Reference>
      </SignedInfo>
      <SignatureValue>……</SignatureValue>
      <KeyInfo>
        <X509Data>
          <X509Certificate>……</X509Certificate>
        </X509Data>
      </KeyInfo>
      <Object>
        <QualifyingProperties Target="signature" xmlns="http://uri.etsi.org/01903/v1.3.2#">
          <SignedProperties Id="xadesSignedProperties">
            <SignedSignatureProperties>
              <SigningTime>2015-09-03T11:13:44.1404616Z</SigningTime>
              <SigningCertificate>
                <Cert>
                  <CertDigest>
                    <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"
xmlns="http://www.w3.org/2000/09/xmldsig#" />
                    <DigestValue xmlns="http://www.w3.org/2000/09/xmldsig#">……</DigestValue>
                  </CertDigest>
                  <IssuerSerial>
                    <X509IssuerName xmlns="http://www.w3.org/2000/09/xmldsig#">CN=PostSignum
Qualified CA 2, O="Česká pošta, s.p. [IČ 47114983]", C=CZ</X509IssuerName>
                    <X509SerialNumber
xmlns="http://www.w3.org/2000/09/xmldsig#">1791234</X509SerialNumber>
                  </IssuerSerial>
                </Cert>
              </SigningCertificate>
            </SignedSignatureProperties>
            <SignedDataObjectProperties>
```

```
            <DataObjectFormat ObjectReference="#xadesReference">
              <MimeType>application/x-gzip</MimeType>
            </DataObjectFormat>
          </SignedDataObjectProperties>
        </SignedProperties>
      </QualifyingProperties>
    </Object>
  </Signature>
</BinaryMessage>
```

3.2.3 Typ Enveloped, mimo zprávu, SignatureContext = "datacontent"

Princip tohoto podpisu je stejný jako v případě 3.2.2 s tím rozdílem, že podpis vznikl nad přenášenou zprávou, která nebyla ještě vložena do ECR obálky. Stejný postup se pak musí aplikovat při ověřování podpisu. Protože toto není výchozí nastavení atributu „SignatureContext“, musí být na příslušném elementu tento atribut uveden.

```
<XmlMessage SignatureContext="datacontent">……</XmlMessage>
```

Stejně jako u 3.2.2 platí restrikce pouze na použití s elementem BinarniZprava.

3.2.4 Typ Enveloped, uvnitř zprávy, SignatureContext = "envelope"

Tento typ znamená, že je podpis vytvořen v kontextu celé ECR obálky (v době podpisu již byla sestavena) a element Signature je umístěn jako potomek kořenového elementu podepisované zprávy. Vždy musí být uvedena transformace „enveloped-signature“.

```
<XmlMessage SignatureContext="datacontent">
  <Data>
    <CZ515A xmlns="">
      <H>
        <QH16>0</QH16>
        <H02>1578142511544H569C7Z</H02>
        ……
      <Signature Id="signature" xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
          <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
          <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
          <Reference URI="" Id="xadesReference">
            <Transforms>
              <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
            <DigestValue>SGtA2UIk+bu1vW2Q7/1cr9v5cMYsE3THj/hS1huWYkE=</DigestValue>
          </Reference>
          <Reference URI="#xadesSignedProperties"
Type="http://www.w3.org/2000/09/xmldsig#SignatureProperties">
            <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
            <DigestValue>e8+dA14qIy8keWzZkXcDIg+p3zDd7M8TMPer3sk6WDY=</DigestValue>
          </Reference>
        </SignedInfo>
        <SignatureValue>……</SignatureValue>
        <KeyInfo>
          <X509Data>
            <X509Certificate>……</X509Certificate>
          </X509Data>
        </KeyInfo>
        <Object>
          <QualifyingProperties Target="signature" xmlns="http://uri.etsi.org/01903/v1.3.2#">
            <SignedProperties Id="xadesSignedProperties">
              <SignedSignatureProperties>
                <SigningTime>2015-09-03T11:28:28.1495064Z</SigningTime>
                <SigningCertificate>
                  <Cert>
                    <CertDigest>
                      <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"
xmlns="http://www.w3.org/2000/09/xmldsig#" />
                      <DigestValue xmlns="http://www.w3.org/2000/09/xmldsig#">……</DigestValue>
                    </CertDigest>
```

```
                    <IssuerSerial>
                      <X509IssuerName xmlns="http://www.w3.org/2000/09/xmldsig#">CN=PostSignum
Qualified CA 2, O="Česká pošta, s.p. [IČ 47114983]", C=CZ</X509IssuerName>
                      <X509SerialNumber
xmlns="http://www.w3.org/2000/09/xmldsig#">1791234</X509SerialNumber>
                    </IssuerSerial>
                  </Cert>
                </SigningCertificate>
              </SignedSignatureProperties>
              <SignedDataObjectProperties>
                <DataObjectFormat ObjectReference="#xadesReference">
                  <MimeType>text/xml</MimeType>
                </DataObjectFormat>
              </SignedDataObjectProperties>
            </SignedProperties>
          </QualifyingProperties>
        </Object>
      </Signature>
    </CZ515A>
  </Data>
</XmlMessage>
```

3.2.5 Typ Enveloped, uvnitř zprávy, SignatureContext = ”datacontent”

Princip tohoto podpisu je stejný jako v případě 3.2.4 s tím rozdílem, že podpis vznikl nad přenášenou zprávou, která nebyla ještě vložena do ECR obálky. Stejný postup se pak musí aplikovat při ověřování podpisu. Protože toto není výchozí nastavení atributu „SignatureContext“, musí být na příslušném elementu tento atribut uveden.

```
<XmlMessage SignatureContext="datacontent">……</XmlMessage>
```

3.3. Kombinace zaručeného elektronického podpisu a šifrování

Při kombinaci obou zabezpečení se nejprve provede vytvoření zaručeného elektronického podpisu a následně se provede šifrování elementu „Data“. V tomto případě nelze použít typ XML Signature popsaný v kapitole 3.2.1, protože by data uvnitř elementu „Signature“ ve výsledku nebyla zašifrována.

Při zpracování příchozí zprávy se postupuje obráceně, nejprve se dešifruje a u výsledku ověří podpis.

4. PŘÍLOHA A – ČÍSELNÍK CHYB ODESÍLANÝCH ECR BRÁNOU

Znak „$“ je zástupným znakem pro hodnotu doplněnou dle dat, u kterých byla zjištěna chyba.

Kód	Text chyby (anglicky)
	Popis chyby
0	Invalid incoming XML
	Neplatné příchozí XML (zpráva není XML nebo neodpovídá schématu ECR obálka).
1	Internal server error
	Vnitřní (blíže neurčená) chyba serveru.
2	Invalid incoming XML message type
	Neznámý typ zprávy v příchozím XML.
3	Message domain is not valid
	Doména uvedená v ECR obálce není známa.
4	ID for declarant is unknown
	ID deklaranta není známo. Nejčastější příčinou tohoto stavu je, že deklarant nemá správně nastavené komunikační parametry v systému ZJP (např. ZJP obsahuje neplatný certifikát).
5	VAN operator $ is not allowed for domain $
	VAN operátor nemá povoleno posílat zprávy do této domény.
6	At least primary or secondary identifier have to be filled in
	V ECR obálce je nutno uvést alespoň jeden identifikátor (primární nebo sekundární).
7	Message $ is not allowed to pass through in the domain $
	Zprávu daného typu nelze poslat do této domény.
12	Decrypt XML failed with: $
	Chyba při dešifrování zprávy ve formátu XML Encryption.
13	Verify sign XML failed due to an incorrect sign format with: $
	Chyba při ověření zprávy ve formátu XML Signature.
14	Verify sign XML failed due to message changes
	Chyba při ověření zprávy ve formátu XML Signature: zpráva byla po podpisu modifikována.
15	Verify sign XML failed due to the incorrect certificate chain or certificate revocation. Certificate number: $
	Chyba při ověření podpisu zprávy. Certifikát použitý pro podpis je neplatný (nedůvěryhodný certifikát; certifikát, jemuž ještě nezačala/skončila platnost; certifikát je uveden v seznamu odvolaných certifikátů;…).
16	Verify sign XML failed due to the unauthorized certificate. Certificate number: $
	Chyba při ověření podpisu zprávy. Certifikát použitý pro podpis není uveden u povolení v systému celní správy.

18	Incorrect data security found: $
	Neplatné zabezpečení dat. Tato chyba nastává například v momentě, kdy je zaslaná zpráva pouze digitálně podepsána, ale pro tento druh komunikace je vyžadováno i šifrování zprávy. Více viz podrobnosti chyby.
	Incorrect data security found: MPSV identifier expected in sign certificate
	Neplatné zabezpečení dat. Certifikát použitý pro podepsání zprávy neobsahuje MPSV identifikátor, ačkoliv je pro danou doménu vyžadován.
19	Strong RSA key (2048bit) and SHA2 digest required
	Je vyžadován podpis zprávy algoritmem rodiny SHA2 za použití certifikátu s minimální délkou klíče 2048 bitů.
	Invalid certificate asymmetric algorithm: $
	Certifikát, jímž byla zpráva podepsána, používá nepovolený asymetrický algoritmus. Povolené algoritmy jsou RSA a DSA.
	Invalid certificate signature algorithm: $
	Certifikát, jímž byla zpráva podepsána, byl certifikační autoritou podepsán nepovoleným algoritmem. Povolené algoritmy jsou SHA-256, SHA-384 a SHA-512.
40	Envelope is neither binary or xml
	ECR obálka neobsahuje ani element BinarniZprava ani element XmlZprava.
53	XML schema general validation error
	Zaslané XML neodpovídá schématu EcrObalka.
54	Invalid child element
	Element zaslané ECR obálky obsahuje nedovolený podelement.
55	Incomplete content
	Element zaslané ECR obálky neobsahuje povinný podelement nebo atribut.
56	Value too short
	Hodnota daného atributu v zaslané ECR obálce je příliš krátká.
57	Value too long
	Hodnota daného atributu v zaslané ECR obálce je příliš dlouhá.
58	Invalid datatype
	Hodnota daného atributu v zaslané ECR obálce je nesprávného typu.
59	Pattern constraint failed
	Hodnota daného atributu v zaslané ECR obálce neodpovídá vzoru definovanému pro tento atribut.
60	Attribute missing
	V zaslané ECR obálce chybí povinný atribut.
70	Primary identifier is not equal to inner message
	Primární identifikátor v ECR obálce (PrimaryID) neobsahuje stejnou hodnotu jako primární identifikátor v zaslané zprávě.
71	Secondary identifier is not equal to inner message
	Sekundární identifikátor uvedený v ECR obálce (SecondaryID) neobsahuje stejnou hodnotu jako sekundární identifikátor v zaslané zprávě.

72	Message type is not equal to inner message
	Typ zprávy uvedený v ECR obálce není roven názvu kořenového elementu přenášené zprávy.
73	Unable to perform inner XML controls in order to incomplete inner message
	Nelze provést kontrolu na shodu primárních nebo sekundárních identifikátorů v ECR obálce a v přenášené zprávě z důvodu nekompletní přenášené zprávy.
74	Invalid use of XPath transformation
	XPath transformace není v tomto případě povolena (její použití je povoleno jen s elementem BinarniZprava)