Egységes MELASZ formátum elektronikus aláírásokra

verzió: 1.0

MELASZ

Magyar Elektronikus Aláírás Szövetség

Tartalom

Változáskezelés

Verzió szám	dátum	a változás leírása
0.9	2003 november	kiinduló munkaanyag, az IHM által készített „XML formátumok elektronikus dokumentumok aláírásához és titkosításához a magyar közigazgatás elektronikus kommunikációjában” című anyag néhány fejezete
0.91	2005 február 26	Az első 4 munkacsoport ülés által elfogadott pontosítások alapján átírt változat
0.92	2005 március 9	A 6. munkacsoport ülésig beérkezett észrevételekkel kiegészített változat
0.93	2005 március 11	A 6. munkacsoport ülésen véglegesített, a MELASZ honlapjára véleményeztetés céljából felrakott verzió
0.94	2005 április 3	További pontosítások a 21 napos véleményezési időszakban beérkezett szerkesztői vélemények alapján.
0.95	2005 április 6	A 7. munkacsoport ülés által elfogadott módosítások alapján készült elsődleges változat.
0.96	2005 április 6	A 7. munkacsoport ülés által elfogadott módosítások alapján készült végleges változat, melyet MELASZ javaslatként megküldtünk az IHM-nek.
0.97	2005 június 12.	A 8. munkacsoport ülés által elfogadott módosítások alapján készült változat (befejezett XAdES-C).
0.98	2005 június 22.	A 9. munkacsoport ülés által elfogadott módosítások alapján készült változat (XAdES-A formátum beépítése).
0.99	2005 június 27.	A 10. munkacsoport ülés által elfogadott módosítások alapján készült változat (OCSP opció beépítése).
1.0 draft	2005 július 11.	A munkacsoport szavazásra jogosult tagjainak elfogadásával készült változat
1.0	2005 szeptember	A MELASZ elnökség által jóváhagyott, nyilvánosságra hozandó dokumentum

1. Bevezetés

1.1 Hatókör

Ez a dokumentum szabványos elektronikus aláírási formátumot (pontosabban formátumokat) határoz meg.

Az aláírási formátumok olyan elektronikus aláírások számára készültek, melyek hosszú távon is érvényesek maradnak, s bizonyítékot szolgáltatnak arra az esetre, ha az aláíró vagy az aláírást ellenőrző felek tagadni próbálnák érvényességüket.

A jelen dokumentumban meghatározott aláírási formátumok nyilvános kulcsú kriptográfián alapulnak, nyilvános kulcsú tanúsítvánnyal támogatott digitális aláírást alkalmazva.

Egyaránt alkalmazhatóak fokozott biztonságú és minősített elektronikus aláírásokra.

A jelen dokumentumban meghatározott aláírási formátumok az alábbi nemzetközi szabványokon alapulnak:

- RFC3369 Cryptographic Message Syntax (CMS) [4],
- ETSI TS 101 733 Electronic Signature Formats [1],
- RFC3275 XML-Signature Syntax and Processing (XMLDSIG) [3],
- ETSI TS 101 903 XML Advanced Electronic Signatures (XAdES) [2].

Jelen dokumentumot a MELASZ formátum munkacsoportjában résztvevő elektronikus aláíró alkalmazás fejlesztő cégek képviselői dolgozták ki. Egy olyan egységes formátum (a továbbiakban egységes MELASZ formátum) elfogadása és értelmezése volt a cél, mely a letagadhatatlanság céljából készített fokozott biztonságú és minősített elektronikus aláírásokra nézve biztosítja a különböző fejlesztésű hazai alkalmazások együttműködő képességét, az alábbi értelemben: a jelen megállapodásnak megfelelő aláírás-létrehozó alkalmazások képesek az egymás által létrehozott aláírásokat ellenőrizni, s azokat (az egységes formátumon belül) azonos eredményre jutva egységesen értelmezni.

Az egységes MELASZ formátum kidolgozása során a magyar közigazgatás igényeire kidolgozott „A közigazgatásban alkalmazható hosszú távú és archív elektronikus aláírás formátumok műszaki specifikációja” című ajánlás elvárásainak, valamint a szabványosításban élenjáró külföldi szervezetek által kidolgozott formátumoknak való megfelelés egyaránt kiemelt cél volt.

Az egységes MELASZ formátum a jelen megállapodásnak megfelelő aláíró alkalmazások számára a minimálisan aláírásba foglalandó adatok körét, s ezek kezelésének kötelező módját határozza meg. A minimálisan elvárt adatokat külön-külön is meghatározza az aláírás életciklusának alábbi szakaszaira:

- aláírás-létrehozás,
- kezdeti ellenőrzés kivárási idő letelte előtt,
- kezdeti ellenőrzés kivárási idő után,
- utólagos ellenőrzés,

s ezzel az aláírást létrehozó, a kezdeti ellenőrzést végző, illetve az utólagos ellenőrzést végző alkalmazásokra (modulokra) funkcionális elvárásokat támaszt.

Az egységes MELASZ formátum nem korlátozza egyéb opcionális elemek használatát, de ezekre az együttműködő képesség nem garantált.

1.2 Felépítés

A dokumentum további része az alábbi szerkezetet követi:

A 2. fejezet a későbbiek egyértelmű értelmezéséhez szükséges fogalmakat határozza meg.

A 3. fejezet az egységes MELASZ formátumok alapját képező két szabványos formátumcsalád (XML és a XAdES formátumok) általános felépítését tekinti át.

A 4. fejezet a „hosszú távú" MELASZ formátum egyértelmű értelmezéséhez és feldolgozásához szükséges általános szabályokat tartalmazza.

Az 5. fejezet az „archív" MELASZ formátum egyértelmű értelmezéséhez és feldolgozásához szükséges általános szabályokat tartalmazza.

A 6. fejezet az aláírás életciklusának különböző kitüntetett időpontjaiban (létrehozás, kezdeti ellenőrzés kivárási idő előtt, kezdeti ellenőrzés kivárási idő után, utólagos ellenőrzés, aláírás archiválása, archivált aláírás ellenőrzése) határozza meg a „hosszú távú" MELASZ formátumra elvárt információ tartalomra vonatkozó minimális elvárásokat.

A 7. fejezet az aláíró és az ellenőrző közötti lehetséges munkamegosztást vizsgálja, külön elemezve a tipikus (szimmetrikus) viszonyt, illetve azt a két speciális esetet, amikor vagy az aláírónak, vagy az aláírás ellenőrzőjének jelentősen könnyített feladata van a másik fél által elvégzett feladatok következményeként.

A 8. és 9. fejezetek a hivatkozásokat és a rövidítések jelentését adja meg.

2. Az aláírás ellenőrzés néhány alapfogalma

2.1 Az aláírások élettartama

Az elektronikus aláírások formátumára vonatkozó, illetve ellenőrzésével szemben támasztott követelmények függenek az elektronikus aláírás várható élettartamától. Az elektronikus aláírások ellenőrzésére vonatkozó mértékadó dokumentum [5] az alábbi eseteket különbözteti meg:

- *pillanatnyi aláírás: elektronikus aláírás, amelynek az élettartama rövidebb az aláírást követő első visszavonási állapot információ kiadásánál,*
- *rövid távú aláírás*: elektronikus aláírás, amelynek az ellenőrzése nem szükséges az aláíró tanúsítványának lejárta után,
- *hosszú távú aláírás*: elektronikus aláírás, amelynek az ellenőrzése szükséges a tanúsítványlánc bármely elemének a lejárta után is,
- *archív aláírás*: elektronikus aláírás, amelynek az ellenőrzése szükséges az aláírás során használt algoritmusok kriptográfiai elavulása után is.

Megjegyzés: A jelen dokumentumban meghatározott „hosszú távú" MELASZ formátum egy speciális hosszú távú aláírás (mely alkalmazható rövid távú aláírásként is), míg az „archív" MELASZ formátum egy speciális archív aláírás.

Az archív aláírási formátum támogatása (az egységes MELASZ formátumot támogató aláíró alkalmazások részéről) opcionális.

2.2 A kezdeti és az utólagos ellenőrzés

Az ellenőrzés kifejezést arra az eljárásra használják, amelynek során egy elektronikus aláírásról megállapítják, hogy érvényes-e vagy sem. Az ellenőrzés két speciális formáját különböztetjük meg:

- *kezdeti ellenőrzés:* az aláírás létrehozása után hamarosan végre kell hajtani annak érdekében, hogy azokat a kiegészítő információkat be lehessen gyűjteni, melyek a hosszú távú ellenőrzésekhez érvényessé teszik az aláírást.
- *utólagos ellenőrzés:* akár évekkel egy aláírás létrehozása után is végre lehet hajtani, és végrehajtásához nincs szüksége több adatra, mint amit a kezdeti ellenőrzés során már begyűjtöttek.

Az egységes MELASZ formátum mindkét fenti ellenőrzésre elvárásokat fogalmaz meg.

2.3 Az érvényesség eldöntéséhez szükséges információk

A hosszú távú aláírás érvényességének eldöntéséhez az aláírás kriptográfiai érvényességét, valamint az aláíró tanúsítványának az aláírás időpontjában való érvényességét bizonyító adatok szükségesek.

A tanúsítvány érvénytelenségét három tényező okozhatja:

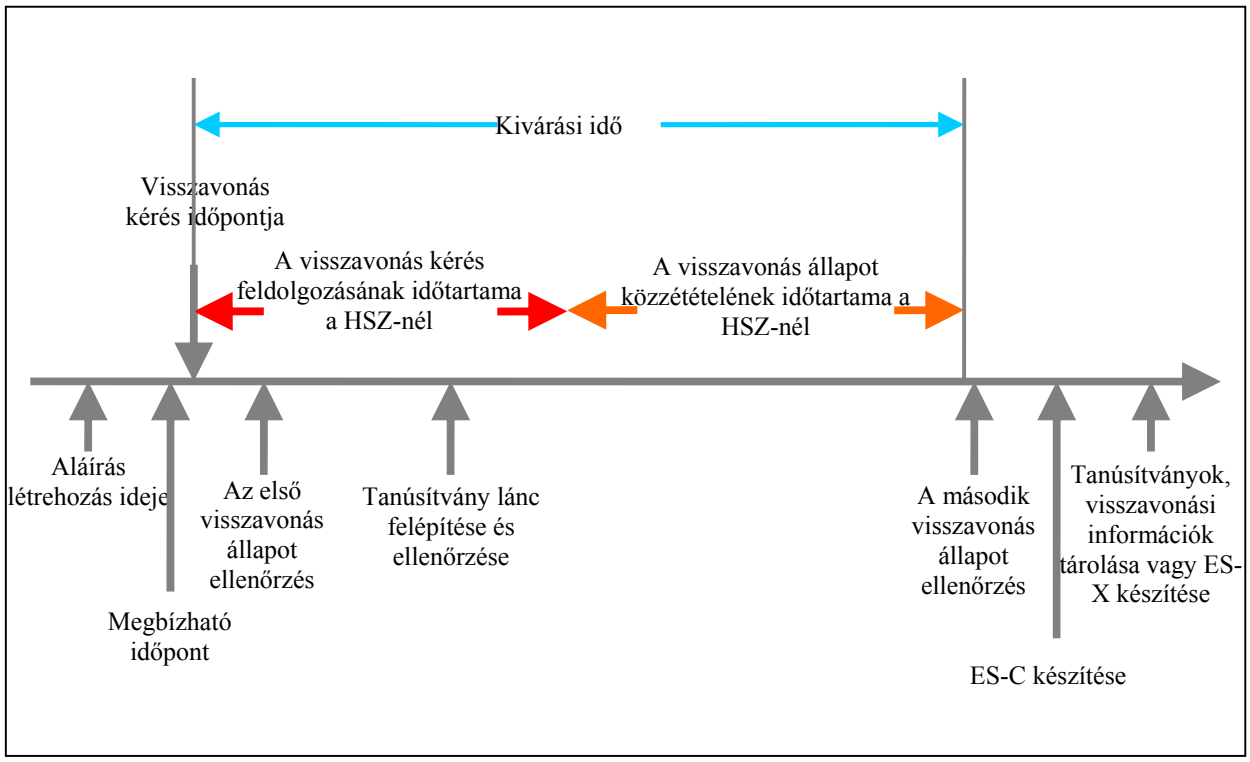
1. a tanúsítványlánc bármely eleméhez tartozó aláíró adat bizalmasságának sérülése,
2. az alkalmazott aláírási algoritmus vagy kulcshossz gyengesége,

3. szervezeti okok, mint például megváltozott hovatartozású vagy lejárt tanúsítvány.

A hosszú távú aláírás érvényességének eldöntéséhez az alábbi alap adatok szükségesek:

1. megbízható időinformáció, minél hamarabb az aláírás kiváltását követően beszerezve (annak állíthatósága érdekében, hogy az elektronikus aláírás ez időpont előtt készült),
2. visszavonási állapot információk beszerzése a tanúsítványlánc minden eleméről a kivárási idő eltelte után.

A kivárási idő lehetővé teszi a tanúsítvány visszavonási információ elterjesztését a visszavonási folyamatokban. Ez az időtartam lefedi azt az időt, ami egy felhatalmazott visszavonás kérésétől addig telik el, amikortól az érintett felek hozzáférhetnek a visszavonási információkhoz. Annak érdekében, hogy meg lehessen győződni arról, hogy az időbélyegzés időpontjában az aláíró tanúsítványa nem volt visszavonva vagy felfüggesztve, az aláírás ellenőrzőnek ki kell várnia a kivárási időt. A kivárási időt szemlélteti az alábbi ábra:



1. ábra Kivárási idő

A kivárási idő az a legrövidebb időtartam, amelyet a kezdeti ellenőrzéshez ki kell várni, annak érdekében, hogy az aláíró vagy egy más erre feljogosított szereplő által esetlegesen kért visszavonási kérelem megjelenhessen a szolgáltató által biztosított visszavonási állapot információk között. (Az 1. ábrán jelzett első visszavonás állapot ellenőrzés során tehát még nem kapható végleges eredmény.)

A kivárási idő mindkét visszavonási információ típus (CRL, OCSP) esetén értelmezett, jelentése is ugyanaz, viszont a két típus között jelentős különbség lehet a szükséges kivárási időben, melyet tipikusan a szolgáltató határoz meg szolgáltatási szabályzatában.

Megbízható időpontra információt kétféleképpen lehet beszerezni:

- egy olyan *időbélyeg* használatával, amely egy időbélyegzés-szolgáltatótól származik, vagy
- *időjelzés* használatával, amelyben egy biztonságos napló rögzíti az időjelzést és az elektronikus aláírás értékét.

Az egységes MELASZ formátum a megbízható időpontra időbélyegzés-szolgáltatótól származó időbélyeg alkalmazását várja el.

Az egységes MELASZ formátumot támogató aláírás-ellenőrző alkalmazások számára a kivárási idő CRL visszavonási információ alkalmazása esetén 24 óra, OCSP alkalmazás esetén 30 perc.

A kivárási időt az aláírásra vonatkozó időbélyegben szereplő időponttól kezdődően kell számítani.

A kivárási idő letelte után minél hamarabb végre kell hajtani a visszavonási állapot információk beszerzését.

Megjegyzés: Annak elvárása, hogy a visszavonási állapot információk beszerzését az időbélyegben szereplő időtől számított kivárási idő letelte után kell végrehajtani, nem mentesíti az egységes MELASZ formátumot támogató aláírás-ellenőrző alkalmazásokat annak ellenőrzésétől, hogy a visszavonási információk valóban az időbélyegben szereplő időpont után keletkeztek (a CRL vagy OCSP válaszban található thisUpdate, valamint az időbélyegben szereplő genTime elemek értékeinek egybevetésével).

Megjegyzés: Annak elvárása, hogy a kivárási idő letelte után minél hamarabb végre kell hajtani a visszavonási állapot információk beszerzését, nem mentesíti az egységes MELASZ formátumot támogató aláírás-ellenőrző alkalmazásokat annak ellenőrzésétől, hogy a visszavonási információk keletkezési időpontjában az ellenőrzött tanúsítvány még érvényes volt (a CRL vagy OCSP válaszban található thisUpdate, valamint az érintett tanúsítványban szereplő notAfter elemek értékeinek egybevetésével).

3. A MELASZ formátumok alapját képező formátumok

3.1 Az XML aláírás formátumok

Megjegyzés: Mindkét MELASZ formátum egy [3]-ban definiált XML (XMLDSIG) elektronikus aláírási formátum is egyben.

Az alábbiakban a MELASZ formátum specifikációk előkészítése érdekében áttekintjük a [3]-ban definiált XML elektronikus aláírási formátumokat.

Az XML elektronikus aláírás formátumok tetszőleges elektronikus tartalom (adat objektum) elektronikus aláírására használhatók. Az adatok lenyomata egyéb információkkal együtt egy külön elembe kerül, és ennek az elemnek a lenyomata lesz valójában elektronikus aláírással ellátva.

Az XML elektronikus aláírás formátum egy `Signature` elemben van leírva az alábbi struktúra szerint:

```
<ds:Signature ID?>
   <ds:SignedInfo>
     <ds:CanonicalizationMethod/>
     <ds:SignatureMethod/>
     (<ds:Reference>
       (<ds:Transforms>)?
       <ds:DigestMethod>
       <ds:DigestValue>
     </ds:Reference>)+
   </ds:SignedInfo>
   <ds:SignatureValue>
  (<ds:KeyInfo>)?
  (<ds:Object>)*
 </ds:Signature>
```

`SignedInfo:` ez az elem az aláírt adat objektumokról tartalmaz információt.

```
<element name="SignedInfo">
<complexType>
  <sequence>
    <element ref="ds:CanonicalizationMethod"/>
    <element ref="ds:SignatureMethod"/>
    <element ref="ds:Reference" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="Id" type="ID" use="optional"/>
</complexType>
</element>
```

`CanonicalizationMethod:` ez az elem adja meg azt az algoritmust, amit az aláírt adat objektum lenyomatolás előtti kanonizálására használtak.

```
<element name="CanonicalizationMethod">
<complexType>
  <sequence>
    <any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="Algorithm" type="uriReference" use="required"/>
</complexType>
</element>
```

`SignatureMethod:` ez az elem azokat az algoritmusokat tartalmazza, amely a kanonizált `SignedInfo` tartalmakat `SignatureValue` tartalommá alakítja. Ezek az algoritmusok lehetnek lenyomatoló eljárások, kulcsfüggő aláíró algoritmusok, valamint egyéb eljárások, mint például feltöltés. Az algoritmusok nevei aláírt attribútumok, ezáltal kivédhetőek az algoritmus cseréjén alapuló támadások. A különböző alkalmazások közötti kompatibilitás biztosítása érdekében [3] meghatározza a kötelezően alkalmazandó algoritmusokat. Ezen felül [3] meghatároz ajánlott, illetve szabadon választható algoritmusokat is.

```
<element name="SignatureMethod">
<complexType>
  <sequence>
    <any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
   </sequence>
 <attribute name="Algorithm" type="uriReference" use="required"/>
</complexType>
</element>
```

`Reference:` minden ilyen elem egy adat objektumra hivatkozik (mely alá lesz írva).

Minden `Reference` elem tartalmazza egy lenyomatoló eljárás meghatározását (`DigestMethod`), valamint a meghatározott adat objektum ilyen algoritmussal készült lenyomatát (`DigestValue`), Base64 kódolással. Tartalmazhat még adat átalakítási leírást (Transforms) is, amellyel a lenyomatoló eljárás bemenő adata – lenyomatkészítés előtt – kialakítható.

```
<element name="Reference">
<complexType>
  <sequence>
    <element ref="ds:Transforms" minOccurs="0"/>
    <element ref="ds:DigestMethod"/>
    <element ref="ds:DigestValue"/>
  </sequence>
  <attribute name="Id" type="ID" use="optional"/>
  <attribute name="URI" type="uriReference" use="optional"/>
  <attribute name="Type" type="uriReference" use="optional"/>
</complexType>
</element>
```

`SignatureValue:` ez az elem tartalmazza az aláírás eredményét, Base64 kódolással.

`KeyInfo`: ez az elem adja meg azt a kulcsot, amivel az aláírás érvényesítése (ellenőrzése) megtörténhet. [3] – jelen ajánlásnál némileg megengedőbben – lehetőséget ad ennek az elemnek az elhagyására. Ebben az esetben a kulcs információkat külső forrásból kell az ellenőrzőnek beszerezni. Mivel a `KeyInfo` a `SignedInfo` elemen kívül van, csak akkor kerül aláírásra, ha egy `Reference` elem hivatkozik rá.

`Object`: ez az (opcionális, de többször is előfordulható) elem tetszőleges adatokat tartalmazhat.

A létrehozott XML elektronikus aláírás lehet:

- különálló állomány az aláírt tartalomtól (*detached signature*),
- a Signature elembe ágyazott aláírt XML tartalom (*enveloping signature*), illetve
- aláírt XML tartalomba ágyazott (*enveloped signature*).

Megjegyzés: A MELASZ formátumokra mint speciális XML elektronikus aláírás formátumokra vonatkozó különleges szabályokat a 4.1 alfejezet részletezi.

3.2 A XAdES aláírás formátumok

Megjegyzés: Mindkét MELASZ formátum egy [2]-ben definiált XAdES elektronikus aláírási formátum is egyben.

Az alábbi XML struktúra az XMLDSIG és a különböző XAdES formátumok tartalmát tekinti át:

Megjegyzés: A MELASZ formátumokra mint speciális XAdES elektronikus aláírás formátumokra vonatkozó különleges szabályokat a 4.2 alfejezet részletezi.

4. A „hosszú távú” MELASZ formátum specifikációja

Megjegyzés: A „hosszú távú” MELASZ formátum specifikációja két lépésben történik:

- **a 4.1 alfejezet részletezi a „hosszú távú” MELASZ formátumra mint speciális XML elektronikus aláírás formátumra vonatkozó különleges szabályokat,**
- **a 4.2 alfejezet részletezi a „hosszú távú” MELASZ formátumra mint speciális XAdES elektronikus aláírás formátumra vonatkozó különleges szabályokat.**

4.1 A „hosszú távú” MELASZ formátumra vonatkozó XML szabályok

Követelmény: A „hosszú távú” MELASZ formátum (mint egy XML elektronikus aláírás formátum) egy Signature elemben van leírva az alábbi struktúra szerint:

```
<element name="Signature">
  <complexType>
    <sequence>
      <element ref="ds:SignedInfo"/>
      <element ref="ds:SignatureValue"/>
      <element ref="ds:KeyInfo"/>
      <element ref="ds:Object" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="Id" type="ID" use="required"/>
  </complexType>
</element>
```

Követelmény: A Signature elemre az alábbi (az XML aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **a KeyInfo elem kötelező[1],**
- **legalább egy Object elem kötelező[2],**
- **a Signature elem Id attribútum használata és egyedi azonosítóval való kitöltése kötelező[3].**

Követelmény: A „hosszú távú” MELASZ formátumot kezelő aláíró alkalmazások aláírás-létrehozó funkcióinak a Signature elem valamennyi fent megnevezett elemét létre kell hozniuk.

Követelmény: A „hosszú távú” MELASZ formátumot kezelő aláírás-ellenőrző programoknak egy-egy XML állományban kötelező módon meg kell találniuk az összes, Signature elemmel leírt aláírást, s ezek mindegyikét ellenőrizniük is kell.

A következő alfejezetek a fenti meghatározott Signature struktúra elemeinek egységes értelmezéséhez szükséges előírásokat részletezik.

[1] [3]-ban hiányozhat is.
[2] [3]-ban hiányozhat is.
[3] [3]-ban ez opcionális.

4.1.1 SignedInfo

Követelmény: A „hosszú távú” MELASZ formátum SignedInfo eleme az alábbi struktúrát követi:

```
<element name="SignedInfo">
<complexType>
  <sequence>
    <element ref="ds:CanonicalizationMethod"/>
    <element ref="ds:SignatureMethod"/>
    <element ref="ds:Reference" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="Id" type="ID" use="required"/>
</complexType>
</element>
```

Követelmény: A SignedInfo elemre az alábbi (az XML aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **a SignedInfo elem Id attribútum használata és egyedi azonosítóval való kitöltése kötelező[4],**
- **a SignedInfo elemben az aláírandó SignedInfo tag-et aláírás előtt kanonizálni kell.**

4.1.1.1 CanonicalizationMethod

A CanonicalizationMethod kötelező elem határozza meg a SignedInfo elemre végrehajtandó kanonizálási eljárást (mielőtt megtörténne a SignatureValue értékének kiszámítása).

Követelmény: A „hosszú távú” MELASZ formátum CanonicalizationMethod eleme az alábbi struktúrát követi[5]:

```
<element name="CanonicalizationMethod">
  <complexType>
    <sequence>
      <any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="Algorithm" type="uriReference" use="required"/>
  </complexType>
</element>
```

Követelmény: A CanonicalizationMethod elemre az alábbi (az XML aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **a CanonicalizationMethod elemben a kötelező Algorithm attribútum csak az alábbi értéket veheti fel: http://www.w3.org/TR/2001/REC-xml-c14n-20010315,**
  vagyis csak a (megjegyzések nélküli) XML kanonizáció támogatott[6],
- **az aláírást tartalmazó XML deklarációban a karakterkészlet megjelölésének UTF-8-nak kell lennie.**

[4] [3]-ban ez opcionális.
[5] Megegyezik a [3] elvárásával
[6] [3] az alábbi kanonizációs algoritmusokat engedi meg: minimális, (megjegyzések nélküli) XML, (megjegyzéssel kiegészített) XML kanonizáció.

4.1.1.2 SignatureMethod

A SignatureMethod kötelező elem azt az algoritmust határozza meg, amelyet az aláírás készítésénél és érvényesítésénél (ellenőrzésénél) használni kell.

Követelmény: A „hosszú távú" MELASZ formátum SignatureMethod eleme az alábbi struktúrát követi[7]:

```
<element name="SignatureMethod">
  <complexType>
    <sequence>
      <any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="Algorithm" type="uriReference" use="required"/>
  </complexType>
</element>
```

Követelmény: A SignatureMethod elemre az alábbi (az XML aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **a kötelező Algorithm attribútum csak az alábbi értéket veheti fel:**
  - **http://www.w3.org/2000/09/xmldsig#rsa-sha1,**

  vagyis kizárólag az SHA-1 lenyomatolással kombinált RSA aláírási algoritmus támogatott.

4.1.1.3 Reference

A SignedInfo elemben egy vagy több Reference elem fordulhat elő. Egy Reference elem tartalmazza a lenyomat függvényt és a lenyomat értékét, azonosítja az aláírt adat objektumot, az adat objektum típusát, valamint az adat objektum lenyomat készítés előtti átalakításának módját. Az adat objektum és az átalakítási metódus azonosítása teszi lehetővé az aláírt adat objektum lenyomatolás előtti formájának egyértelmű utólagos visszaállítását.

Követelmény: A Reference elem módosított struktúrája az alábbi:

```
<element name="Reference">
  <complexType>
    <sequence>
      <element ref="ds:Transforms" minOccurs="0"/>
      <element ref="ds:DigestMethod"/>
      <element ref="ds:DigestValue"/>
    </sequence>
    <attribute name="Id" type="ID" use="required"/>
    <attribute name="URI" type="uriReference" use="required"/>
    <attribute name="Type" type="uriReference" use="optional"/>
  </complexType>
</element>
```

[7] Megegyezik a [3] elvárásával

Követelmény: A Reference elemre az alábbi (az XML aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **Id: az Id attribútum használata és egyedi azonosítóval való kitöltése kötelező[8],**
- **URI: az URI attribútum jelenléte, és nem üres kitöltése kötelező[9],**
- **URI: az URI attribútumra az alábbi kiegészítő korlátozások vannak:**
  - **Az URI-ban megengedett mind a belső, mind a külső hivatkozás.**
  - **Az URI-ban XPointer hivatkozás nem megengedett.**
  - **Belső hivatkozás esetén (#valami) az XML bármely elemére lehet hivatkozni, aminek az adott nevű Id attribútuma van. Az ellenőrző program az adott XML-ben megkeresi ezt az elemet.**
  - **Külső hivatkozás esetén:**
    - **file://xxx – az alkalmazás megkeresi a fájlt az URI alapján (relatív vagy abszolút út). Ha nem találja, akkor egyéb módon (pl. felhasználói interakcióval) bekéri, vagy hibajelzést ad.**
    - **http, https: az alkalmazás letölti a fájlt az URI alapján. Amennyiben az URI-n 30x redirect van, akkor követni kell az átirányítást, és a 200-OK eredményt kell inputnak tekinteni.**
    - **Más mód (ldap://, ftp://, stb.) használata nem megengedett.**
  - **A „vegyes" mód (pl. http://example.com/bar.xml#chapter1) nem megengedett.**
- **Type: a Type attribútumra az alábbi kiegészítő korlátozások vannak:**
  - **A Type attribútum használata nem kötelező (azaz használható, de nem kell érteni), kivétel a SignedProperty-re való hivatkozásnál, mert ott a XAdES szerint kötelező[10].**
  - **Amennyiben a Type attribútum jelen van, tartalmának összhangban kell lennie az alábbi dokumentumokkal:**

Dokumentum	Fejezet	Elem	Type tartalma
[3] (XMLDSIG)	4.4.4	X509Data	http://www.w3.org/2000/09/xmldsig#X509Data
[3] (XMLDSIG)	4.5	Object	http://www.w3.org/2000/09/xmldsig#Object
[3] (XMLDSIG)	5.1	Manifest	http://www.w3.org/2000/09/xmldsig#Manifest
[3] (XMLDSIG)	5.2	SignatureProperties	http://www.w3.org/2000/09/xmldsig# SignatureProperties
[4] (XAdES)	6.3.1	SignedProperties	http://uri.etsi.org/01903/v1.2.2#SignedProperties

Követelmény: A referencia feldolgozásra (dereferálásra) vonatkozó szabályok a következők:

- **A referencia nem tartalmazhat XPath kifejezést.**
- **Az URI dereferencia vagy egy-egy korábbi transzformáció eredménye mindig bájt-folyam, vagy XPath node-set. A transzformációkra nézve:**
  - **ha az adat bájt-folyam és a következő transzformáció node-set-et feltételez, akkor meg kell próbálni a bájt-folyamot node-set-ként értelmezni/használni.**
  - **Ha az adat node-set és a következő transzformáció bájt-folyamot feltételez, akkor az alkalmazásnak a node-set-et C14N kanonizációval kell bájt-folyammá alakítania.**
- **A transzformációk végeztével alkalmazandó lenyomatolás az utolsó transzformációból kijövő bájt-folyamon értelmezendő.**
- **Külső referencia esetén a dereferálás mindig bájt-folyamot eredményez. Példák:**
  - **URI="http://example.com/bar.xml": Az a bájtfolyam, ami ezen a címen elérhető (várhatóan XML).**
  - URI="http://example.com/bar.xml#chapter1": Az adott címről letölthető XML „#chapter1" Id-jű eleme (mint XML elem, a nyitó és záró elemeivel együtt). Nem támogatott.
  - URI="": A fentiek szerint ez nem támogatott.
  - **URI="#chapter1": a szóban forgó aláírást tartalmazó XML-ből az az elem, amelynek Id-je „chapter1", minden leszármazottjával és névterével együtt, de megjegyzések nélkül.**

4.1.1.3.1 A Transforms elem

[8] [3]-ban ez opcionális
[9] [3]-ban ez opcionális
[10] lásd [4]: 6.3.1

Az opcionális Transforms elem meghatározott sorrendben Transform elemeket tartalmaz, leírva ezekkel azon műveleteket, melyeket az aláíró az adat objektumon lenyomatolás előtt elvégzett. Minden Transform kimeneti adata a következő Transform bemeneti adata is egyben.

Követelmény: A Transforms elem az alábbi struktúrát követi[11]:

```
<element name="Transforms">
  <complexType>
    <sequence>
      <element ref="ds:Transform" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Követelmény: A Transforms elemre az alábbi (az XML aláírási formátumokhoz képest kiegészítő) elvárások vonatkoznak:

- **amennyiben egyetlen Transform elemet sem tartalmaz, akkor – mivel tartalom nélkül állna – elhagyható,**
- **ha egyetlen Transform elem sincs, akkor a dereferálás eredménye egyben a transzformációk outputja is,**
- **első transzformációnál az input a dereferálásból származó bájt-folyam, egyéb esetekben minden transzformáció inputja az előző transzformáció outputja,**
- **az utolsó transzformáció eredménye kerül lenyomatolásra (mint bájt-folyam, azaz ha az node-set, akkor előbb kötelező kanonizáció kell).**

Minden (opcionális) Transform elem Algorithm attribútumot, valamint amennyiben az adott algoritmus ezt szükségessé teszi, paramétereket tartalmaz. Az Algorithm attribútum az átalakító algoritmust határozza meg. A Transform elem az algoritmus használatát befolyásoló kiegészítő adatokat is tartalmazhat.

Követelmény: A Transform elem az alábbi struktúrát követi:

```
<element name="Transform">
  <complexType>
    <sequence>
      <any namespace="##other" processContents="lax" minOccurs="0"
           maxOccurs="unbounded"/>
    </sequence>
    <attribute name="Algorithm" type="uriReference" use="required"/>
  </complexType>
</element>
```

Követelmény: A Transform elemre az alábbi (az XML aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **az XPath szűrés mint transzformáció nem megengedett,**
- **az Enveloped Signature Transform mint transzformáció nem megengedett,**
- **az XSLT transzformáció nem megengedett.**

- **a kötelező Algorithm attribútum csak az alábbi értékeket veheti fel:**
  - **http://www.w3.org/TR/2001/REC-xml-c14n-20010315,**

[11] Megegyezik a [3] elvárásával

- **http://www.w3.org/2000/09/xmldsig#BASE64,**

vagyis csak a kanonikus XML kanonizálási algoritmus (C14N) és a BASE64 kódolás támogatott.

Megjegyzés:

- **A C14N kanonizáció használata egyértelmű. A MELASZ formátum csak megjegyzés nélküli kanonizációt enged meg.**
- **A Base64 kódolás feltételezi, hogy az inputja BASE64 kódolva van (vagyis az aláírandó adatot BASE64 dekódolja), és BASE64->bináris konverziót végez. Amennyiben a BASE64 kódolt adat XML elemek között volt, úgy a nyitó és záró elemeket elhagyja, azok nem vesznek részt a lenyomatolásban. A base64 kódolt adatot tartalmazó elemek nem tartalmazhatnak leszármazott elemeket.**

4.1.1.3.2. Fájl tartalom beágyazása és aláírása

Fájl tartalmat base64 kódolással egy tetszőleges nevű xml tag-be kell elhelyezni a dokumentumon belül. Aláíráskor referenciát kell képezni erre az xml tag-re, a transzformációnak pedig kötelező jelleggel Base64 kódolásnak kell lennie. Ebben az esetben csak a fájl eredeti tartalmának a lenyomata képződik, a burkoló xml tag-é nem (Lásd 4.1.1.3.1). A fájl leírókat kötelezően a DataObjectFormat elembe kell elhelyezni (lásd 4.2 2.1). A DataObjectFormat elem nem támogatja a fájlnév tulajdonság használatát, ezért a fájl tartalmának xml burkoló tag-jébe egy nem kötelező, nem aláírt xml attribútumot lehet felvenni „FileName” néven, string típussal. Ez tartalmazza az aláírt fájl eredeti nevét.

```
<attribute name="FileName" type="xsd:string" use="optional"/>
```

Követelmény: A fájl tartalom beágyazására és aláírására vonatkozóan nincs korlátozás.

4.1.1.3.3 A DigestMethod elem

A kötelező DigestMethod elem azt a lenyomatoló algoritmust határozza meg, amelyet az aláírt adat objektum lenyomatolására használtak.

Követelmény: A DigestMethod elem az alábbi struktúrát követi[12]:

```
<element name="DigestMethod">
  <complexType>
    <sequence>
      <any namespace="##any" processContents="lax" minOccurs="0"
          maxOccurs="unbounded"/>
    </sequence>
    <attribute name="Algorithm" type="uriReference" use="required"/>
  </complexType>
</element>
```

Követelmény: A DigestMethod elemre az alábbi (az XML aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárás vonatkozik:

- **A kötelező Algorithm attribútuma csak az alábbi értéket veheti fel:**
  - **<xsd:enumeration value="http://www.w3.org/2000/09/xmldsig#sha1" />,**

vagyis kizárólag az SHA-1 algoritmus használható lenyomatolásra.

[12] Megegyezik a [3] elvárásával

4.1.1.3.4 A DigestValue elem

A kötelező DigestValue elem BASE64 kódoltan az aláírt adat lenyomatát tartalmazza.

Követelmény: A DigestValue elem az alábbi struktúrát követi[13]:

```
<element name="DigestValue" type="ds:DigestValue"/>
  <simpleType name="DigestValueType">
    <restriction BASE="BASE64Binary"/>
  </simpleType>
```

4.1.2 A SignatureValue elem

A SignatureValue kötelező elem BASE64 kódoltan tartalmazza az elektronikus aláírás értékét.

Követelmény: A SignatureValue elem az alábbi struktúrát követi:

```
<element name="SignatureValue" type="ds:SignatureValueType"/>
  <complexType name="SignatureValueType">
    <simpleContent>
      <extension BASE="BASE64Binary">
        <attribute name="Id" type="ID" use="required"/>
      </extension>
    </simpleContent>
  </complexType>
```

Követelmény: A SignatureValue elemre az alábbi (az XML aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárás vonatkozik:

- **az Id attribútum használata és egyedi azonosítóval való kitöltése kötelező.**

[13] Megegyezik a [3] elvárásával

4.1.3 A KeyInfo elem

A kötelező KeyInfo elem az aláírás érvényesítéséhez szükséges kulcs információkat vagy az azokra való hivatkozást tartalmazza. Opcionálisan tanúsítvány visszavonási listát is tartalmazhat.

Követelmény: A KeyInfo elem az alábbi struktúrát követi:

```
<element name="KeyInfo">
  <complexType>
    <sequence>
      <element ref="ds:X509Data"/>
    </sequence>
    <attribute name="Id" type="ID" use="required"/>
  </complexType>
</element>
```

Követelmény: A KeyInfo elemre az alábbi (az XML aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **a KeyInfo elem kötelező,**
- **a KeyInfo elemben csak X509Data elem szerepelhet,[14]**
- **a KeyInfo elemben az Id attribútum használata és egyedi azonosítóval való kitöltése kötelező[15].**
- **A KeyInfo elemnek kötelezően meg kell határoznia az aláírás ellenőrzésére használható nyilvános kulcsot, egy X509Data elemmel, s ebben a szükséges X509-es tanúsítvánnyal.**
- **A KeyInfo nincs aláírva, azaz nincs rá referencia a SignedInfo-ban[16].**

4.1.3.1 Az X509Data elem

Követelmény: Az X509Data elem az alábbi struktúrát követi[17]:

```
<element name="X509Data">
  <complexType>
    <choice>
      <sequence maxOccurs="unbounded">
         <choice>
           <element name="X509IssuerSerial"/>
           <element name="X509SKI" type="ds:CryptoBinary"/>
           <element name="X509SubjectName" type="string"/>
           <element name="X509Certificate" type="ds:CryptoBinary"/>
         </choice>
      </sequence>
      <element name="X509CRL" type="ds:CryptoBinary"/>
    </choice>
  </complexType>
</element>
```

[14] [3]-ban az X509Data egy sor más alternatíva között csak az egyik lehetőség
[15] [3]-ban ez opcionális
[16] A nem aláírt KeyInfo célja az ellenőrzés megkönnyítése a fogadó oldalán. Az ellenőrzés során meg kell oldani, hogy mindenképpen a megfelelő aláírói tanúsítvány legyen felhasználva, ennek megoldására azonban nincs előírás.
[17] Megegyezik a [3] elvárásával

Követelmény: Az X509Data elemre az alábbi (az XML aláírási formátumokhoz képest eltérő vagy kiegészítő) korlátozások vannak:

- **az X509Data elem használata kötelező,**
- **a tanúsítványlánc minden elemét – a gyökér kivételével – X509Certificate alakban kell szerepeltetni, a gyökértanúsítványt pedig X509IssuerSerial elemmel,**
- **az aláíró tanúsítványának szerepeltetése kötelező, a lánc többi eleme opcionális.**
- **a tanúsítványok közül első az aláíróé, ezután – ha van folytatás –a lánc többi eleme következik sorban, egészen a gyökérig.**

4.1.4 Az Object elem

Az Object elem meghatározását a következő (4.2) alfejezet részletezi. Ennek az az oka, hogy egy XAdES formátum egy olyan általános XML [3] formátum, amely az Object elemet másként (pontosabban) definiálja, a MELASZ formátumok pedig egyben XAdES formátumok is.

4.2 A „hosszú távú” MELASZ formátumra vonatkozó XAdES szabályok

Követelmény: A „hosszú távú” MELASZ formátum, mint egy XAdES elektronikus aláírás formátum, egy olyan (a 4.1 alfejezetben meghatározott) Signature elemmel van leírva, melynek Object eleme(i) az alábbi struktúrát követi(k):

Követelmény: Az Object elem az alábbi struktúrát követi:

```
<ds:Object>
  <QualifyingProperties>
    <SignedProperties>
    <UnsignedProperties>
  </QualifyingProperties>
</ds:Object>
```

Követelmény: Az Object elemre az alábbi (az XML aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárás vonatkozik:

- **az Object elemben az Id attribútum használata és egyedi azonosítóval való kitöltése kötelező[18].**

Az Object elem az elektronikus aláírás érvényesítő adatait tartalmazza egy QualifyingProperties elemben. A QualifyingProperties elem azokat az aláírási tulajdonságokat tartalmazza, amelyeket az XML aláíráshoz hozzá kell adni.

Követelmény: A QualifyingProperties elem az alábbi struktúrát követi:

```
<xsd:element name="QualifyingProperties" type="QualifyingPropertiesType"/>
 <xsd:complexType name="QualifyingPropertiesType">
   <xsd:sequence>
     <xsd:element name="SignedProperties" type="SignedPropertiesType"/>
         minOccurs="0"/>
     <xsd:element name="UnsignedProperties" type="UnsignedPropertiesType"
         minOccurs="0"/>
   </xsd:sequence>
   <xsd:attribute name="Target" type="xsd:anyURI" use="required"/>
   <xsd:attribute name="Id" type="xsd:ID" use="required"/>
</xsd:complexType>
```

Követelmény: A QualifyingProperties elemre az alábbi (a XAdES aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **a Target attribútumnak kötelezően a ds:Signature Id attribútumára kell mutatnia,**
- **a QualifyingProperties elemben az Id attribútum használata és egyedi azonosítóval való kitöltése kötelező[19].**

Az aláírási tulajdonságok két csoportba vannak osztva aszerint, hogy az aláíró azokat aláírta (SignedProperties) vagy sem (UnsignedProperties).

A SignedProperties elem azokat az aláírási tulajdonságokat és egyéb aláírt adatokat tartalmazza, amelyeket az aláíró aláír.

[18] [3]-ban ez opcionális
[19] [4]-ban ez opcionális

Követelmény: A SignedProperties elem az alábbi struktúrát követi:

```
<xsd:element name="SignedProperties" type="SignedPropertiesType" />
<xsd:complexType name="SignedPropertiesType">
  <xsd:sequence>
    <xsd:element name="SignedSignatureProperties"
    type="SignedSignaturePropertiesType"/>
    <xsd:element name="SignedDataObjectProperties"
    type="SignedDataObjectPropertiesType"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="required"/>
</xsd:complexType>
```

Követelmény: A SignedProperties elemre az alábbi (a XAdES aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **a SignedProperties elem alkalmazása kötelező, már az aláírás létrehozásakor,**
- **minden aláírás által védett információt (SignedProperties) egyetlen QualifyingProperties elemben kell összegyűjtve szerepeltetni,**
- **a SignedProperties elemre egy ds:Reference hivatkozásnak kell mutatnia, ahol a Reference elem Type attribútumának az értéke:**
  - **http://uri.etsi.org/01903/v1.2.2#SignedProperties,**
- **az Id attribútum használata és egyedi azonosítóval való kitöltése kötelező.**

Az UnsignedProperties elem az aláíró által nem aláírt aláírási tulajdonságokat tartalmazza.

Követelmény: Az UnsignedProperties elem az alábbi struktúrát követi:

```
<xsd:element name="UnsignedProperties" type="UnsignedPropertiesType"/>
<xsd:complexType name="UnsignedPropertiesType">
  <xsd:sequence>
    <xsd:element name="UnsignedSignatureProperties"
        type="UnsignedSignaturePropertiesType" minOccurs="0"/>
    <xsd:element name="UnsignedDataObjectProperties"
        type="UnsignedDataObjectPropertiesType" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

Követelmény: Az UnsignedProperties elemre az alábbi (a XAdES aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárás vonatkozik:

- **az UnsignedProperties elem alkalmazása kötelező a kezdeti ellenőrzéstől kezdve (de opcionálisan az aláírás létrehozásakor is használható).**

4.2.1 A SignedSignatureProperties elem

A SignedSignatureProperties elem azokat az aláíró által aláírt érvényesítő adatokat tartalmazza, amelyek a QualifyingProperties Target attribútumában hivatkozott aláírás érvényesítéséhez szükségesek.

Követelmény: A SignedSignatureProperties elem az alábbi struktúrát követi:

```
<xsd:element name="SignedSignatureProperties"
         type="SignedSignaturePropertiesType" />
<xsd:complexType name="SignedSignaturePropertiesType">
  <xsd:sequence>
    <xsd:element name="SigningTime" type="xsd:dateTime"/>
    <xsd:element name="SigningCertificate" type="CertIDListType"/>
    <xsd:element name="SignaturePolicyIdentifer"
         type="SignaturePolicyIdentifierType"/>
    <xsd:element name="SignatureProductionPlace"
         type="SignatureProductionPlaceType" minOccurs="0"/>
    <xsd:element name="SignerRole" type="SignerRoleType" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

Követelmény: A SignedSignatureProperties elemre az alábbi (a XAdES aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárás vonatkozik:

- **a SignedSignatureProperties elem SigningTime, SigningCertificate, valamint SignaturePolicyIdentifier elemeinek támogatása kötelező, míg a többi elem nem támogatott (azaz szerepeltethetőek, de értelmezésük nem kötelező).**

4.2.1.1 A SigningTime elem

Ez az elem az aláíró által állított aláírási időpontot tartalmazza (tájékoztatás céljára).

Követelmény: A SigningTime elem az alábbi struktúrát követi[20]:

```
<xsd:element name="SigningTime" type="xsd:dateTime"/>
```

[20] Megegyezik a [4] elvárásával

4.2.1.2 A SigningCertificate elem

A SigningCertificate elem használatának az a célja, hogy az aláírás után ne legyen lehetséges a tanúsítvány (észrevétlen) kicserélése. Hivatkozást tartalmaz a tanúsítványra vonatkozóan, illetve tárolja annak lenyomatát.

Követelmény: A SigningCertificate elem az alábbi struktúrát követi[21]:

```
<xsd:element name="SigningCertificate" type="CertIDListType"/>
<xsd:complexType name="CertIDListType">
  <xsd:sequence>
    <xsd:element name="Cert" type="CertIDType" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CertIDType">
  <xsd:sequence>
    <xsd:element name="CertDigest" type="DigestAlgAndValueType"/>
    <xsd:element name="IssuerSerial" type="ds:X509IssuerSerialType"/>
  </xsd:sequence>
<xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>

</xsd:complexType>
<xsd:complexType name="DigestAlgAndValueType">
  <xsd:sequence>
    <xsd:element ref="ds:DigestMethod"/>
    <xsd:element ref="ds:DigestValue"/>
  </xsd:sequence>
</xsd:complexType>
```

Követelmény: A SigningCertificate elemre az alábbi (a XAdES aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárás vonatkozik:

- **a SigningCertificate elem használata kötelező.**

4.2.1.3 A SignaturePolicyIdentifier elem

Az aláírási szabályzat azon szabályok összessége, amelyeket az elektronikus aláírás létrehozásakor, illetve ellenőrzésekor be kell tartani annak érdekében, hogy az elektronikus aláírás érvényes legyen. Az aláírási szabályzat egyértelmű azonosításának kétféle módja van:

1. Az elektronikus aláírás aláírt aláírási tulajdonságként tartalmazza az aláírási szabályzat egyértelmű és félreérthetetlen azonosítóját (az aláírási szabályzat egyedi hivatkozását), valamint az aláírási szabályzat lenyomatát (explicit aláírás szabályzat meghatározás).

2. Abban az esetben, amikor az aláírt adat objektum típusa, kiegészítve egyéb információkkal (mint például törvény vagy szerződés) egyértelműen meghatározza az alkalmazandó aláírási szabályzatot, az előző esetben leírt információk megadása nem kötelező (implicit aláírás szabályzat meghatározás).

Követelmény: A SignaturePolicyIdentifier elem az alábbi struktúrát követi[22]:

[21] Megegyezik a [4] elvárásával

```
<xsd:element name="SignaturePolicyIdentifier"
     type="SignaturePolicyIdentifierType"/>
<xsd:complexType name="SignaturePolicyIdentifierType">
  <xsd:choice>
    <xsd:element name="SignaturePolicyId" type="SignaturePolicyIdType"/>
    <xsd:element name="SignaturePolicyImplied"/>
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="SignaturePolicyIdType">
  <xsd:sequence>
    <xsd:element name="SigPolicyId" type="ObjectIdentifierType"/>
    <xsd:element ref="ds:Transforms" minOccurs="0"/>
    <xsd:element name="SigPolicyHash" type="DigestAlgAndValueType"/>
    <xsd:element name="SigPolicyQualifiers"
    type="SigPolicyQualifiersListType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SigPolicyQualifiersListType">
  <xsd:sequence>
    <xsd:element name="SigPolicyQualifiers" type="AnyType"
    maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

A SigPolicyQualifiers elem az aláírási szabályzatra történő hivatkozást és a felhasználó számára kötelezően megjelenítendő információt tartalmaz.

Követelmény: A SigPolicyQualifier elem az alábbi struktúrát követi[23]:

```
<xsd:element name="SPURI" type="xsd:anyURI"/>
<xsd:element name="SPUserNotice" type="SPUserNoticeType"/>
<xsd:complexType name="SPUserNoticeType">
  <xsd:sequence>
    <xsd:element name="NoticeRef" type="NoticeReferenceType" minOccurs="0"/>
    <xsd:element name="ExplicitText" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="NoticeReferenceType">
  <xsd:sequence>
    <xsd:element name="Organization" type="xsd:string"/>
    <xsd:element name="NoticeNumbers" type="IntegerListType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="IntegerListType">
  <xsd:sequence>
    <xsd:element name="int" type="xsd:integer" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

Követelmény: A SigPolicyQualifiers elemre az alábbi (a XAdES aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárás vonatkozik:

- **amennyiben a szabályzat explicit módon van meghatározva, a SigPolicyQualifiers elem használata kötelező.**

[22] Megegyezik a [4] elvárásával
[23] Megegyezik a [4] elvárásával

Megjegyzés: Az aláírási szabályzat kötelező megadása tehát az alábbi két módon történhet:

- **explicit módon OID vagy URL segítségével, vagy**
- **implicit módon, a SignaturePolicyImplied elemmel.**

4.2.2 A SignedDataObjectProperties elem

A SignedDataObjectProperties elem azokat az aláíró által aláírt érvényesítő, az aláírandó adatra vonatkozó adatokat tartalmazza, amelyek a QualifyingProperties Target attribútumában hivatkozott aláírás érvényesítéséhez szükségesek.

Követelmény: A SignedDataObjectProperties elem az alábbi struktúrát követi[24]:

```
<xsd:element name="SignedDataObjectProperties"
     type="SignedDataObjectPropertiesType"/>

<xsd:complexType name="SignedDataObjectPropertiesType">
  <xsd:sequence>
    <xsd:element name="DataObjectFormat" type="DataObjectFormatType"
    minOccurs="1" maxOccurs="unbounded"/>
    <xsd:element name="CommitmentTypeIndication"
         type="CommitmentTypeIndicationType" minOccurs="0"
         maxOccurs="unbounded"/>
    <xsd:element name="AllDataObjectsTimeStamp" type="TimeStampType"
         minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="IndividualDataObjectsTimeStamp" type="TimeStampType"
         minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

Követelmény: A SignedDataObjectProperties elemre az alábbi (a XAdES aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárás vonatkozik:

- **a SignedDataObjectProperties elem DataObjectFormat elemének támogatása kötelező, míg a többi elem nem támogatott (azaz szerepeltethetőek, de értelmezésük nem kötelező).**

[24] Megegyezik a [4] elvárásával

4.2.2.1 A DataObjectFormat elem

A DataObjectFormat elem tartalmazza az aláírt adat formátumával kapcsolatos dolgokat. Az alelemei közül a MimeType az, amelyet formátumunkban kötelező használni. Ennek értéke az aláírt adat mime formátumára utal.

Annyi DataObjectFormat elemnek kell szerepelnie, ahány Reference elem szerepel a SignedInfo elemben, tehát ahány adatot egyszerre aláírunk. Természetesen kivételt képeznek a XAdES aláírási formátumon belüli hivatkozások, mint például a SignedProperties Reference eleme. Az ObjectReference attribútumnak kell hivatkoznia a Reference elemekre (Id attribútumaira).

Követelmény: A DataObjectFormat elem az alábbi struktúrát követi[25]:

```
<xsd:element name="DataObjectFormat" type="DataObjectFormatType"/>
<xsd:complexType name="DataObjectFormatType">
  <xsd:sequence>
    <xsd:element name="Description" type="xsd:string" minOccurs="0"/>
    <xsd:element name="ObjectIdentifier" type="ObjectIdentifierType"
         minOccurs="0"/>
    <xsd:element name="MimeType" type="xsd:string"/>
    <xsd:element name="Encoding" type="xsd:anyURI" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="ObjectReference" type="xsd:anyURI" use="required"/>
</xsd:complexType>
```

Követelmény: A DataObjectFormat elemre az alábbi (a XAdES aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **a DataObjectFormat elem használata kötelező,**
- **a MimeType alelem használata kötelező.**

[25] Megegyezik a [4] elvárásával

4.2.3 Az UnsignedSignatureProperties elem

Az UnsignedSignatureProperties elem azokat az aláíró által nem aláírt érvényesítő adatokat tartalmazza, amelyek a QualifyingProperties Target attribútumában hivatkozott aláírás érvényesítéséhez (ellenőrzéséhez) szükségesek.

Követelmény: Az UnsignedSignatureProperties elem az alábbi struktúrát követi[26]:

```
<xsd:element name="UnsignedSignatureProperties"
     type="UnsignedSignaturePropertiesType"/>

<xsd:complexType name="UnsignedSignaturePropertiesType">
  <xsd:sequence>
    <xsd:element name="CounterSignature" type="CounterSignatureType"
         minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="SignatureTimeStamp" type="TimeStampType"
         minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="CompleteCertificateRefs"
         type="CompleteCertificateRefsType" minOccurs="1"/>
    <xsd:element name="CompleteRevocationRefs"
         type="CompleteRevocationRefsType" minOccurs="0"/>
    <xsd:element name="AttributeCertificateRefs"
         type="CompleteCertificateRefsType" minOccurs="0"/>
    <xsd:element name="AttributeRevocationRefs"
         type="CompleteRevocationRefsType" minOccurs="0"/>
    <xsd:choice>
      <xsd:element name="SigAndRefsTimeStamp" type="TimeStampType"
           minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="RefsOnlyTimeStamp" type="TimeStampType"
           minOccurs="0" maxOccurs="unbounded"/>
    </xsd:choice>
    <xsd:element name="CertificateValues" type="CertificateValuesType"
         minOccurs="0"/>
    <xsd:element name="RevocationValues" type="RevocationValuesType"
         minOccurs="0"/>
    <xsd:element name="ArchiveTimeStamp" type="TimeStampType"
         minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

Követelmény: Az UnsignedSignatureProperties elemre az alábbi (a XAdES aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **az UnsignedSignatureProperties elemben a SignatureTimeStamp, CompleteCertificateRefs és CompleteRevocationRefs elemek használata kötelező,**
- **a CertificateValues elem használata is kötelező abban az esetben, ha a tanúsítvány útvonalhoz tartozó tanúsítvány referenciák a CompleteCertificateRefs elemben belső hivatkozásra mutatnak,**

[26] Megegyezik a [4] elvárásával

- **a RevocationValues elem használata is kötelező abban az esetben, ha a visszavonási információkhoz tartozó CRL vagy OCSP válasz referenciák a CompleteRevocationRefs elemben belső hivatkozásra mutatnak,**
- **a többi elem nem támogatott (azaz szerepeltethetőek, de értelmezésük nem kötelező).**

4.2.3.1 A SignatureTimeStamp elem

A SignatureTimeStamp a ds:SignatureValue-ra vonatkoztatott időbélyeget foglalja magába.

Követelmény: A SignatureTimeStamp elem az alábbi struktúrát követi:

```
<xsd:element name="SignatureTimeStamp" type="TimeStampType"/>

<xsd:complexType name="TimeStampType">
    <xsd:sequence>
      <xsd:element name="Include" type="IncludeType" maxOccurs="unbounded"/>
      <xsd:element ref="ds:CanonicalizationMethod" minOccurs="0"/>
      <xsd:element name="EncapsulatedTimeStamp"
           type="EncapsulatedPKIDataType"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:ID" use="required"/>
  </xsd:complexType>

  <xsd:complexType name="IncludeType">
    <xsd:attribute name="URI" type="xsd:anyURI" use="required"/>
    <xsd:attribute name="referencedData" type="xsd:boolean" use="optional"/>
  </xsd:complexType>
```

Követelmény: A SignatureTimeStamp elemre az alábbi (a XAdES aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **a SignatureTimeStamp használata (legalább egyszer) kötelező,**
- **a SignatureTimeStamp elemet nem feltétlenül az aláírónak kell csatolnia (ha nincs a fogadott aláírásban időbélyeg, akkor az aláírás kezdeti ellenőrzését végző részéről kötelező a kérése és csatolása),**
- **a SignatureTimeStamp elemben kötelező szerepelnie pontosan egy Include elemnek, amelynek URI-ja a ds:SignatureValue elemre mutat,**
- **az EncapsulatedTimeStamp használata kötelező,**
- **az EncapsulatedTimeStamp esetében csak a base64 kódolt, alapértelmezett DER megengedett,**
- **az XMLTimeStamp elem nem támogatott[27],**
- **az Id attribútum használata és egyedi azonosítóval való kitöltése kötelező[28],**
- **a SignatureTimeStamp kérésekor kötelező az időbélyegzés-szolgáltatótól tanúsítványt is kérni.**

[27] [4] –ben az EncapsulatedTimeStamp és az XMLTimeStamp között választani lehet
[28] [4] -ben ez opcionális

4.2.3.2 A CompleteCertificateRefs elem

A CompleteCertificateRefs elem az aláíró tanúsítványát hitelesítő tanúsítvány útvonal tanúsítványaira való hivatkozásokat tartalmazza.

Követelmény: A CompleteCertificateRefs elem az alábbi struktúrát követi:

```
<xsd:element name="CompleteCertificateRefs"
     type="CompleteCertificateRefsType"/>

<xsd:complexType name="CompleteCertificateRefsType">
  <xsd:sequence>
    <xsd:element name="CertRefs" type="CertIDListType" />
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="required"/>
</xsd:complexType>
```

Követelmény: A CompleteCertificateRefs elemre az alábbi (a XAdES aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **a CompleteCertificateRefs használata kötelező, az aláírás létrehozásakor kell csatolni,**
- **Az Id attribútum használata és egyedi azonosítóval való kitöltése kötelező[29],**
- **A CompleteCertificateRefs nem tartalmazza az aláíró tanúsítványát, csak a tanúsítási útvonalból a közbenső és legfelső szintű CA tanúsítványokat[30] (a közbenső tanúsítványok esetében a CertIDType URI attribútumának szerepeltetése kötelező /hivatkozás/, míg a legfelső szintű tanúsítványok esetében a CertIDType URI attribútumának elhagyása megengedett /csak azonosítás/),**
- **A CompleteRevocationRefs elemtől eltérően itt a külső és a belső hivatkozások egyaránt megengedettek. Amennyiben a hivatkozás belső, akkor annak a CertificateValues elem megfelelő alelemére kell mutatnia.**

4.2.3.3 A CompleteRevocationRefs elem

A CompleteRevocationRefs elem hivatkozásokat tartalmaz az aláíró, valamint a hitelesítő tanúsítvány útvonal elemeire vonatkozó visszavonási állapot információkra.

Jelenleg két fő típusa van:

- az időszakosan frissülő CRL lista, illetve
- egy saját protokollon (OCSP) elérhető azonnali tanúsítvány állapotot szolgáltató szerver.

[29] [4] -ben ez opcionális
[30] lásd [4]: 4.4.3.2

Követelmény: A CompleteRevocationRefs elem az alábbi struktúrát követi:

```
<xsd:element name="CompleteRevocationRefs"
     type="CompleteRevocationRefsType"/>
<xsd:complexType name="CompleteRevocationRefsType">
  <xsd:sequence>
    <xsd:element name="CRLRefs" type="CRLRefsType" minOccurs="0"/>
    <xsd:element name="OCSPRefs" type="OCSPRefsType" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="required"/>
</xsd:complexType>
<xsd:complexType name="CRLRefsType">
  <xsd:sequence>
    <xsd:element name="CRLRef" type="CRLRefType" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CRLRefType">
  <xsd:sequence>
    <xsd:element name="DigestAlgAndValue" type="DigestAlgAndValueType"/>
    <xsd:element name="CRLIdentifier" type="CRLIdentifierType"
         minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CRLIdentifierType">
  <xsd:sequence>
    <xsd:element name="Issuer" type="xsd:string"/>
    <xsd:element name="IssueTime" type="xsd:dateTime" />
    <xsd:element name="Number" type="xsd:integer" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
</xsd:complexType>
<xsd:complexType name="OCSPRefsType">
  <xsd:sequence>
    <xsd:element name="OCSPRef" type="OCSPRefType" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="OCSPRefType">
  <xsd:sequence>
    <xsd:element name="OCSPIdentifier" type="OCSPIdentifierType"/>
    <xsd:element name="DigestAlgAndValue" type="DigestAlgAndValueType"
         minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="OCSPIdentifierType">
  <xsd:sequence>
    <xsd:element name="ResponderID" type="xsd:string"/>
    <xsd:element name="ProducedAt" type="xsd:dateTime"/>
  </xsd:sequence>
  <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
</xsd:complexType>
```

Követelmény: A CompleteRevocationRefs elemre az alábbi (a XAdES aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **a CompleteRevocationRefs elem használata kötelező, legkésőbb az aláírás kezdeti ellenőrzésekor kell csatolni,**
- **mind a CRL listát, mind az OCSP protokollt támogatni kell (a mindkét visszavonási állapot információ típus támogatása azt jelenti, hogy az aláírás létrehozásakor vagy az aláírás kezdeti ellenőrzésekor a visszavonási állapot információt elhelyező alkalmazásnak valamelyik információ típust csatolnia kell, míg a visszavonási állapot információt ellenőrző kezdeti vagy utólagos ellenőrzést végrehajtó alkalmazásnak mindkét információ típus ellenőrzésére alkalmasnak kell lennie),**
- **az Id attribútum használata és egyedi azonosítóval való kitöltése kötelező[31],**
- **a kivárási idő után végrehajtott kezdeti ellenőrzésnek csatolnia kell az aktuális CRL listát vagy az OCSP választ is, s erre a CompleteRevocationRefs elemben egy belső hivatkozásnak kell mutatnia a RevocationValues elem megfelelő alelemére.**

4.2.3.4 A CertificateValues elem

Követelmény: A CertificateValues elem az alábbi struktúrát követi:

```
<xsd:element name="CertificateValues" type="CertificateValuesType"/>
<xsd:complexType name="CertificateValuesType">
  <xsd:element name="EncapsulatedX509Certificate"
       type="EncapsulatedPKIDataType" minOccurs="0" maxOccurs="unbounded"/>
  <xsd:attribute name="Id" type="xsd:ID" use="required”/>
</xsd:complexType>

<xsd:complexType name="EncapsulatedPKIDataType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:base64Binary">
      <xsd:attribute name="Id" type="xsd:ID" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

Követelmény: A CertificateValues elemre az alábbi (a XAdES aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **a CertificateValues elem használata kötelező, amennyiben a tanúsítvány útvonalhoz tartozó tanúsítvány referenciák (CompleteCertificateRefs) belső hivatkozásra mutatnak (ilyenkor ezeknek a hivatkozásoknak a megfelelő EncapsulatedPKIData elemekre kell mutatniuk), s ilyen esetekben legkésőbb az aláírás kezdeti ellenőrzésekor kell csatolni,**
- **a CertificateValues elem tartalmazza az aláíró és az időbélyeg szolgáltató tanúsítvány hitelesítési útvonalának tanúsítványait (Amennyiben a CompleteCertificateRefs a gyökér tanúsítványt csak azonosítással adta meg /vagyis a CertIDType opcionális URI attribútuma nem szerepel/, akkor a CertificateValues elem gyökértanúsítványt nem tartalmaz. Amennyiben a CompleteCertificateRefs a gyökér tanúsítványt belső hivatkozással adta meg /vagyis a CertIDType opcionális URI attribútuma belső**

[31] [4] -ben ez opcionális

címmel szerepel/, akkor a CertificateValues elemnek tartalmaznia kell a gyökértanúsítványt.),

- **a tanúsítványokat X509-es formátumban BASE64-el kódolva kell elhelyezni, az EncapsulatedX509Certificate alelemekben,**
- **az Id attribútum használata és egyedi azonosítóval való kitöltése kötelező[32].**

4.2.3.5 A RevocationValues elem

A RevocationValues elem tartalmazza az aláíró és az időbélyeg szolgáltató tanúsítvány hitelesítési láncára vonatkozó visszavonási listákat vagy OCSP válaszokat, X509-es formátumban BASE64-el kódolva.

Követelmény: A RevocationValues elem az alábbi struktúrát követi:

```
<xsd:element name="RevocationValues" type="RevocationValuesType"/>
<xsd:complexType name="RevocationValuesType">
  <xsd:sequence>
    <xsd:element name="CRLValues" type="CRLValuesType" minOccurs="0"/>
    <xsd:element name="OCSPValues" type="OCSPValuesType" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="required"/>
</xsd:complexType>

<xsd:complexType name="CRLValuesType">
  <xsd:sequence>
    <xsd:element name="EncapsulatedCRLValue" type="EncapsulatedPKIDataType"
    maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="OCSPValuesType">
  <xsd:sequence>
    <xsd:element name="EncapsulatedOCSPValue" type="EncapsulatedPKIDataType"
    maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="EncapsulatedPKIDataType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:base64Binary">
      <xsd:attribute name="Id" type="xsd:ID" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

Követelmény: A RevocationValues elemre az alábbi (a XAdES aláírási formátumokhoz képest eltérő vagy kiegészítő) elvárások vonatkoznak:

- **a RevocationValues elem használata kötelező, amennyiben a visszavonási információkhoz tartozó (CRL vagy OCSP válasz) referenciák a CompleteRevocationRefs elemben belső hivatkozásra mutatnak,**
- **legkésőbb az aláírás kezdeti ellenőrzésekor (de az időbélyegben szereplő időponthoz képest a kivárási idő letelte után) kell csatolni,**

[32] [4] -ben ez opcionális

- **a CompleteRevocationRefs hivatkozásoknak CRL alkalmazása esetén a megfelelő EncapsulatedCRLValue elemekre, OCSP alkalmazása esetén pedig a megfelelő EncapsulatedOCSPValue elemekre kell mutatniuk,**
- **az Id attribútum használata és egyedi azonosítóval való kitöltése (mind a CRL, mind az OCSP alkalmazása esetén) kötelező[33].**

4.2.3.6 A Manifest elem

A MELASZ formátum nem engedi a Manifest elem használatát.

[33] [4] -ben ez opcionális

5. Az „archív” MELASZ formátum specifikációja

Az alábbiakban meghatározzuk azokat a kiegészítéseket, melyekkel a „hosszú távú” MELASZ formátum archiválási célra is megfelelő lesz.

Az „archív MELASZ formátum támogatása a „hosszú távú” MELASZ formátumot támogató aláíró alkalmazások számára nem kötelező. Amennyiben viszont egy a „hosszú távú” MELASZ formátumot támogató aláíró alkalmazás felvállalja az „archív” formátum támogatását is, akkor az alábbi követelmények mindegyikét is be kell tartania.

Az archív aláírási formátum még a következő potenciális veszélyek ellen is képes védelmet garantálni:

- az érintett hitelesítés-szolgáltatók (tanúsítvány kiadó, CRL vagy OCSP válaszokat aláíró) magánkulcsainak későbbi kompromittálódása,
- a tanúsítványok és dokumentumok aláíró algoritmusainak későbbi feltörése (beleértve ezek alatt a lenyomat függvényt és a digitális aláírásra alkalmazott algoritmust is).

A fentiek érdekében az archív aláírási formátumban az aláíró által nem aláírt érvényesítő adatokat (az UnsignedSignatureProperties elemben) ki kell egészíteni az alábbiakkal:

- SigAndRefsTimeStamp vagy RefsOnlyTimeStamp elem (opcionális új elvárás),
- Certification Values (szigorított elvárás),
- RevocationValues (szigorított elvárás),
- ArchiveTimeStamp (új elvárás).

5.1 A SigAndRefsTimeStamp elem

Azokban az esetekben, amikor visszavonási információként OCSP válaszokat használnak, időbélyegezni lehet magát az OCSP-t , az OCSP szolgáltató kulcsának kompromittálódása esetére.

Mivel az OCSP válasz felhasználónként és kérésről-kérésre is különböző, ezért minden fogadott aláírásra külön időbélyegzésre van szükség. Ezért az időbélyeget (a SigAndRefsTimeStamp elemet) nem csak az OCSP válaszra érdemes kérni, hanem (ugyanazért a költségért több elemre biztosítható integritás védelem érdekében) a következő elemekből képzett lenyomat értékre:

- digitális aláírás (ds: Signature elem),
- az aláírásra kért időbélyeg(ek) (SignatureTimeStamp elem(ek)),
- a tanúsítvány lánc referenciái (CompleteCertificateRefs elem),
- a visszavonási információkra való hivatkozások (CompleteRevocationRefs elem)

Követelmény: Az opcionális SigAndRefsTimeStamp elem az alábbi struktúrát követi[34]:

```
<xsd:element name="SigAndRefsTimeStamp” type="TimeStampType"/>
```

[34] Megegyezik a [4] elvárásával

Követelmény: Az „archív” MELASZ formátumot támogató alkalmazásoknak az opcionális SigAndRefsTimeStamp elem kérése előtt a következő Include elem sorozatot kell összeállítaniuk:

- **egy Include elem, amelynek URI-ja a ds:SignatureValue elemre mutat,**
- **egy-egy Include elem, minden SignatureTimeStamp elemre,**
- **egy Include elem, amelynek URI-ja a CompleteCertificateRefs elemre mutat,**
- **egy Include elem, amelynek URI-ja a CompleteRevocationRefs elemre mutat.**

Követelmény: Az „archív” MELASZ formátumot támogató alkalmazásoknak az opcionális SigAndRefsTimeStamp elem előkészítését, kérését, fogadását, ellenőrzését és aláírásba foglalását a kivárási idő után végrehajtott kezdeti ellenőrzés során kell végrehajtani.

5.2 A RefsOnlyTimeStamp elem

Azokban az esetekben, amikor visszavonási információként CRL listákat használnak, az időbélyeget (a RefsOnlyTimeStamp elemet) a következő elemekből képzett lenyomat értékre lehet kérni:

- a tanúsítvány lánc referenciái (CompleteCertificateRefs elem),
- a visszavonási információkra való hivatkozások (CompleteRevocationRefs elem)

Követelmény: A RefsOnlyTimeStamp elem az alábbi struktúrát követi[35]:

```
<xsd:element name="RefsOnlyTimeStamp” type="TimeStampType"/>
```

Követelmény: Az „archív” MELASZ formátumot támogató alkalmazásoknak az opcionális RefsOnlyTimeStamp elem kérése előtt a következő Include elem sorozatot kell összeállítaniuk:

- **egy Include elem, amelynek URI-ja a CompleteCertificateRefs elemre mutat,**
- **egy Include elem, amelynek URI-ja a CompleteRevocationRefs elemre mutat.**

Követelmény: Az „archív” MELASZ formátumot támogató alkalmazásoknak az opcionális RefsOnlyTimeStamp elem előkészítését, kérését, fogadását, ellenőrzését és aláírásba foglalását a kivárási idő után végrehajtott kezdeti ellenőrzés során kell végrehajtani.

5.3 A CertificateValues elem

Követelmény: Az „archív” MELASZ formátumban elhelyezendő CertificateValues elemre a 4.2.3.4 alfejezetben meghatározottakon kívül az alábbi is vonatkozik:

- **a CertificateValues elem használata kötelező, és a tanúsítvány útvonalhoz tartozó tanúsítvány referenciák (CompleteCertificateRefs) csak belső hivatkozásra mutathatnak.**

[35] Megegyezik a [4] elvárásával

5.4 A RevocationValues elem

Követelmény: Az „archív” MELASZ formátumban elhelyezendő RevocationValues elemre a 4.2.3.5 alfejezetben meghatározottakon kívül az alábbi is vonatkozik:

- **a RevocationValues elem használata kötelező, és a visszavonási információkhoz tartozó (CRL vagy OCSP válasz) referenciák a CompleteRevocationRefs elemben csak belső hivatkozásra mutathatnak.**

5.5 Az ArchiveTimeStamp elem

Az archív időbélyeg azt a célt szolgálja, hogy védelmet nyújtson az aláírás során felhasznált kriptográfiai algoritmusok feltörése, illetve az alkalmazott kulcsok kompromittálódása esetén is.

Archív időbélyeg alkalmazására van szükség az alábbi esetekben:

- az aláírás létrehozására használt lenyomat függvény vagy aláíró algoritmus (az alkalmazott kulcsméretet is figyelembe véve) már nem biztonságos,
- az időbélyegekben használt lenyomat függvény már nem biztonságos,
- a tanúsítványok, CRL-ek, OCSP és időbélyeg válaszok aláírásához használt hash függvény vagy aláíró algoritmus (az alkalmazott kulcsméretet is figyelembe véve) már nem biztonságos,
- a tanúsítványok, CRL-ek és OCSP és időbélyeg válaszok aláírásához használt szolgáltatói magánkulcsok kompromittálódtak.

Az archív időbélyeget még a fent felsorolt események bekövetkezése előtt kell létrehozni.

Követelmény: Az ArchiveTimeStamp elem az alábbi struktúrát követi[36]:

```
<xsd:element name="ArchiveTimeStamp” type="TimeStampType"/>
```

Követelmény: Az „archív” MELASZ formátumot támogató alkalmazásoknak az ArchiveTimeStamp elem kérése előtt a következő Include elem sorozatot kell összeállítaniuk:

- **egy-egy Include elem a ds:SignatureValue elemben található minden ds:Reference elemre (minden Include elem URI-ja ezen ds:Reference elemek egyikére mutat, a megfelelő referencedData attribútum értékének „true” állapota mellett),**
- **egy Include elem, amelynek URI-ja a ds:SignedInfo elemre mutat,**
- **egy Include elem, amelynek URI-ja a ds:SignatureValue elemre mutat,**
- **egy Include elem, amelynek URI-ja a ds:KeyInfo elemre mutat,**
- **egy-egy Include elem, minden SignatureTimeStamp elemre,**
- **egy-egy Include elem, minden CounterSignature elemre, amennyiben vannak ilyenek,**
- **egy Include elem, amelynek URI-ja a CompleteCertificateRefs elemre mutat,**

[36] Megegyezik a [4] elvárásával

- **egy Include elem, amelynek URI-ja a CompleteRevocationRefs elemre mutat,**
- **egy Include elem, amelynek URI-ja a CertificateValues elemre mutat, s amennyiben a CertficateValues elem még nem létezik, akkor ezt létre kell hozni,**
- **egy Include elem, amelynek URI-ja a RevocationValues elemre mutat, s amennyiben a RevocationValues elem még nem létezik, akkor ezt létre kell hozni,**
- **egy-egy Include elem minden SigAndRefsTimeStamp elemre, amennyiben vannak ilyenek (minden Include elem URI-ja a SigAndRefsTimeStamp elemek egyikére mutat),**
- **egy-egy Include elem minden RefsOnlyTimeStamp elemre, amennyiben vannak ilyenek (minden Include elem URI-ja a RefsOnlyTimeStamp elemek egyikére mutat),**
- **egy-egy Include elem minden már meglévő ArchiveTimeStamp elemre, amennyiben vannak ilyenek (minden Include elem URI-ja az ArchiveTimeStamp elemek egyikére mutat),**
- **egy-egy Include elem az aláírásban található minden olyan ds:Object elemre, melyre a ds:SignatureValue elemben található egyetlen ds:Reference elem sem hivatkozik (minden Include elem URI-ja ezen ds:Object elemek egyikére mutat).**

Követelmény: Minden archív időbélyeget még a magánkulcsok kompromittálódása, illetve a lenyomat függvények és az aláíró algoritmusok feltörhetővé válása előtt kell elhelyezni, az archív időbélyegzés időpontjában biztonságosnak tekintett lenyomat függvény és aláíró algoritmus alkalmazásával.

Követelmény: Az archív időbélyeg létrejöttekor már szerepelnie kell az aláíró és az összes addigi időbélyeg kiadó tanúsítványnak a láncellenőrzéséhez szükséges tanúsítványoknak és visszavonási információknak (CRL vagy OCSP válasz) a CompleteCertificateRefs, a CompleteRevocationRefs, a CertificateValues és a RevocationValues elemekben.

A gyökértanúsítványok azonosítása a CompleteCertificateRefs elemben kötelező, de a hivatkozással történő szerepeltetés is megengedett. Így a CertificateValues elemben a gyökértanúsítványok szerepeltetése opcionális.

6. Az aláírási formátum felépítésének szakaszai

6.1 Az aláírás létrehozása során elérendő formátum

Követelmény: A „hosszú távú" MELASZ formátumot támogató aláíró alkalmazások aláírás-létrehozó funkcióinak legalább XAdES-EPES formátumot kell biztosítaniuk, teljesítve az alábbiakat:

1. **A Signature elem valamennyi alábbi elemeit létre kell hozniuk, s végleges adattartalommal kell feltölteniük:**
   - **SignedInfo (4.1.1),**
   - **SignatureValue (4.1.2) és**
   - **KeyInfo (4.1.3)**
2. **A Signature elem alábbi elemeit létre kell hozniuk, s részleges adattartalommal kell feltölteniük:**
   - **Object (4.1.4).**
3. **A részlegesen feltöltött Object elemnek tartalmaznia kell az alábbi elemeket:**
   - **SigningTime (4.2.1.1),**
   - **SigningCertificate (4.2.1.2),**
   - **SignaturePolicyIdentifier (4.2.1.3),**
   - **DataObjectFormat (4.2.2.1),**
   - **CompleteCertificateRefs (4.2.3.2),**

Megjegyzés: Az Object elemben opcionálisan az alábbi elemek is elhelyezhetők:

- **SignatureTimeStamp (4.2.3.1),**
- **CompleteRevocationRefs (4.2.3.3),**
- **CertificateValues (4.2.3.4).**

6.2 Az aláírás kezdeti ellenőrzése során elérendő formátum

Követelmény: A „hosszú távú" MELASZ formátumot támogató aláíró alkalmazások kezdeti aláírás-ellenőrzést végző funkcióinak végre kell hajtaniuk az alábbiakat:

1. **Az aláírás létrehozás során támogatandó formátum (6.1 alatti 1., 2. és 3. pontok) ellenőrzése:**
   - **az elvárt minimális formátum hiányossága esetén „érvénytelen" eredmény mellett a kezdeti ellenőrzés befejezése, a bemeneti formátum változatlan hagyása mellett,**
   - **az elvárt minimális formátum megléte esetén a 2. pont végrehajtása.**

2. Az aláírás kiegészítése az alábbi nem aláírt aláírási tulajdonságokkal (amennyiben azt az aláírás létrehozásakor, vagy egy korábbi kezdeti ellenőrzés során nem helyezték még el):
- **SignatureTimeStamp (4.2.3.1),**
- **CompleteRevocationRefs (4.2.3.3),**
- **CertificateValues (4.2.3.4).**

A teljes kiegészítés sikertelensége esetén „befejezetlen" eredmény mellett a kezdeti ellenőrzés befejezése.
A teljes kiegészítés sikeressége esetén a 3. pont végrehajtása.

3. A „hosszú távú" MELASZ formátum elérésének kísérlete:
- **amennyiben az időbélyegben szereplő időponthoz képest a kivárási idő még nem telt el, „befejezetlen" eredmény mellett a kezdeti ellenőrzés befejezése.**
- **amennyiben az időbélyegben szereplő időponthoz képest a kivárási idő letelt, az aláírás kiegészítése az alábbi nem aláírt aláírási tulajdonsággal:**
  - **RevocationValues (4.2.3.5),**

  ennek a kiegészítésnek a sikertelensége esetén „befejezetlen" eredmény mellett a kezdeti ellenőrzés befejezése,
  a kiegészítés sikere esetén a kezdeti ellenőrzés 4. pontjának végrehajtása.

4. Az aláírás érvényességének ellenőrzése:
- **melynek eredménye „érvényes", „érvénytelen" és „befejezetlen" egyaránt lehet.**

Az aláírás (első) kezdeti ellenőrzését közvetlenül az aláírt dokumentum fogadása után célszerű végrehajtani, mivel időbélyeg hiányában ennek az ellenőrzésnek kell időbélyeget kérnie, s ebben az esetben ettől az időponttól számítódik a kivárási idő.

A „Befejezetlen" eredményt adó kezdeti ellenőrzést meg kell ismételni. Ezt az ismétlést a kivárási idő letelte után minél hamarabb célszerű végrehajtani, mert ekkor már beszerezhető minden szükséges érvényesítő adat, de ezek közül egyesek idővel nem elérhetőkké válnak.

6.3 Az aláírás utólagos ellenőrzése során elvárt formátum

A „hosszú távú" MELASZ formátumot támogató aláíró alkalmazások utólagos aláírás-ellenőrzést végző funkcióinak kiegészítő adatok beszerzése nélkül, a már korábban begyűjtött érvényesítő adatok alapján kell „érvényes" vagy „érvénytelen" ellenőrzési eredményre jutniuk.

Követelmény: A „hosszú távú" MELASZ formátumot támogató aláíró alkalmazások utólagos aláírás-ellenőrzést végző funkcióinak végre kell hajtaniuk az alábbiakat:

1. Az aláírás létrehozása során készített, majd a kezdeti ellenőrzés(ek) során kiegészített aláírásban a minimálisan elvárt érvényesítő adatok meglétének ellenőrzése.

2. A minimálisan elvárt érvényesítő adatok alapján az aláírás ellenőrzése.

6.4 Az aláírás archiválásakor elérendő formátum

Az „archív" MELASZ formátumot támogató aláíró alkalmazások archiválást végző funkcióinak kiegészítő adatokat kell beszerezniük, s az aláíráson elhelyezniük.

Követelmény: Az „archív" MELASZ formátumot támogató aláíró alkalmazások archiválást végző funkcióinak XAdES-A formátumot kell biztosítaniuk, teljesítve az alábbiakat:

1. **Első archiválás esetén az alábbi elemeket kell létrehozniuk, s az aláíráshoz csatolniuk:**
   - **opcionálisan SigAndRefsTimeStamp (5.1) vagy RefsOnlyTimeStamp (5.2)[37] ,**
   - **CertificateValues (5.3)[38],**
   - **RevocationValues (5.4)[39],**
   - **ArchiveTimeStamp (5.5).**

6.5 Az archivált aláírás ellenőrzések elvárt formátum

Az „archív" MELASZ formátumot támogató aláíró alkalmazások archivált aláírások ellenőrzését végző funkcióinak kiegészítő adatok beszerzése nélkül, a már korábban begyűjtött érvényesítő adatok alapján kell „érvényes" vagy „érvénytelen" ellenőrzési eredményre jutniuk.

Követelmény: Az „archív" MELASZ formátumot támogató aláíró alkalmazások archivált aláírások ellenőrzését végző funkcióinak végre kell hajtaniuk az alábbiakat:

1. **Az „archív" MELASZ formátumokban minimálisan elvárt érvényesítő adatok meglétének ellenőrzése.**
2. **A minimálisan elvárt érvényesítő adatok alapján az aláírás ellenőrzése.**

[37] Attól függően, hogy visszavonási információként OCSP vagy CRL használatos.
[38] A már aláíráshoz csatolt korábbi CertificateValues elem szükség szerinti módosításával.
[39] A már aláíráshoz csatolt korábbi RevocationValues elem szükség szerinti módosításával.

7. Lehetséges munkamegosztás az aláíró és az ellenőrző között

Egy általános aláíró alkalmazásnak az alábbi (egymást követően aktivizálandó) funkció vannak:

- aláírás-létrehozás,
- kezdeti ellenőrzés kivárási idő letelte előtt,
- kezdeti ellenőrzés kivárási idő után,
- utólagos ellenőrzés.

A fenti négy funkció eltérő erőforrás és ismeret igényeket támaszt, s ezek részben attól is függnek, hogy az időben megelőző funkciók milyen opcionális feladatok elvégzését vállalták fel:

- az aláírás létrehozásakor szükség van a magánkulcs tárolását és aktivizálását végző elemre (az úgynevezett tokenre, például egy intelligens kártyára), az aláírás kiváltásához szükséges hitelesítő adatra (például egy PIN kód megadására), esetleg hálózati kapcsolatra is (pl. időbélyegzés),
- a kivárási idő letelte előtt végrehajtott kezdeti ellenőrzés esetén szükség lehet hálózati kapcsolatra (pl. időbélyegzés), valamint annak ismeretére (és megértésére), hogy ezt a kezdeti ellenőrzést meg kell még ismételni a kivárási idő letelte után,
- a kivárási idő letelte után végrehajtott kezdeti ellenőrzés esetén általában (fájl protokoll használat kivételével) szükség van hálózati kapcsolatra, az aktuális CRL vagy az OCSP válasz beszerzéséhez,
- végül az utólagos ellenőrzés általában nem vár el már hálózati kapcsolatot (kivéve, ha az aláírandó adat vagy tanúsítvány vagy CRL http-s külső hivatkozáson keresztül érhető el), s eredménye könnyen értelmezhető (csak „érvényes” és „érvénytelen” eredmény lehetséges).

7.1 Szimmetrikus aláíró – ellenőrző viszony

Általános esetben az aláíró és az aláírást ellenőrző oldalon kompatibilis aláíró alkalmazások futnak, melyek mind a négy funkció aktivizálására képesek.

Az aláíró alkalmazás felhasználója egyaránt képes aláíró és aláírás ellenőrző szerepkörben dolgozni.

Ellenőrzéskor nem kell tokenjét aktivizálnia, de a teljes folyamat minden funkciójával tisztában kell lennie, a különböző esetek és variációk között el kell tudnia igazodni.

Az alábbi táblázat ezt a szimmetrikus viszonyt jellemzi.

	Aláíró	Ellenőrző
szükséges funkciók	• aláírás-létrehozás, • kezdeti ellenőrzés kivárási idő letelte előtt, • utólagos ellenőrzés	• kezdeti ellenőrzés kivárási idő letelte előtt, • kezdeti ellenőrzés kivárási idő után, • utólagos ellenőrzés
szükséges erőforrások	• hálózati kapcsolat • token (aláíró magánkulccsal)	• hálózati kapcsolat
szükséges ismeretek	• minden funkció ismerete • hitelesítő adat	• minden funkció ismerete
Egyéb előfeltételek	• szerződés hitelesítés-szolgáltatóval, • kapcsolat időbélyegzés-szolgáltatóval	• kapcsolat időbélyegzés-szolgáltatóval

1. táblázat: munkamegosztás szimmetrikus esetben

Aláírói oldalon az utólagos ellenőrzés azért szükséges, hogy az aláíró is képes legyen saját aláírásának későbbi ellenőrzésére.

7.2 Aláíró és ellenőrző szerver/kliens kapcsolata

Bizonyos esetekben érdek fűződhet ahhoz, hogy az aláíró, vagy az ellenőrző oldalán minél egyszerűbb alkalmazás legyen futtatható.

Abban a speciális esetben, amikor egy központi erőforrás (pl. MELASZ szolgáltatás által nyújtott digitális aláírással hitelesített információszolgáltatás, vagy egy elektronikusan kibocsátott közüzemi számla) az aláíró, és sok kliens az ellenőrző, ahhoz fűződhet érdek, hogy az ellenőrzés minél egyszerűbb módon mehessen végbe.

Az alábbi táblázat az ellenőrző számára elérhető legelőnyösebb aszimmetrikus viszonyt jellemzi.

	Aláíró	Ellenőrző
szükséges funkciók	• aláírás-létrehozás, • kezdeti ellenőrzés kivárási idő letelte előtt, • kezdeti ellenőrzés kivárási idő után, • utólagos ellenőrzés	• utólagos ellenőrzés
szükséges erőforrások	• hálózati kapcsolat • token (aláíró magánkulccsal)	• hálózati kapcsolat
szükséges ismeretek	• minden funkció ismerete • hitelesítő adat	• az utólagos ellenőrzés funkció ismerete
Egyéb előfeltételek	• szerződés hitelesítés-szolgáltatóval, • szerződés időbélyegzés-szolgáltatóval	---

2. táblázat: munkamegosztás szerver/kliens alapú aláíró/ellenőrző kapcsolat esetén

Az ellenőrző számára igen előnyös fenti eredmény csak akkor érhető el, ha az aláíró a következőket hajtja végre:

1. létrehozza az aláírást (XAdES-EPES formátumban),
2. közvetlenül ez után időbélyeget kér rá és helyez el az aláíráson (az 1. vagy a 3. lépésben),
3. közvetlenül ez után kezdeti ellenőrzést hajt végre (kivárási idő letelte előtt),
4. a kivárási idő letelte után ismételten kezdeti ellenőrzést hajt végre,
5. az utólagos ellenőrzésre így alkalmassá tett aláírt dokumentumot küldi el az ellenőrzőnek.

Ennek a munkamegosztásnak van egy hátránya is az ellenőrző szempontjából: csak a kivárási idő letelte után kapja csak meg az aláírt dokumentumot. Ha ez a hátrány nem vállalható, ez a munkamegosztás nem jöhet szóba.

7.3 Aláíró és ellenőrző kliens/szerver kapcsolata

Egy másik speciális esetben, amikor egy központi erőforrás vár el sok klienstől aláírt dokumentum beküldését (ilyenek például a különböző bevallások hatóság részére való beküldése), pont ellenkezőleg, ahhoz fűződhet érdek, hogy az aláírás legyen minél egyszerűbb.

Az alábbi táblázat az aláíró számára elérhető legelőnyösebb aszimmetrikus viszonyt jellemzi.

	Aláíró	Ellenőrző
szükséges funkciók	• aláírás-létrehozás • utólagos ellenőrzés	• kezdeti ellenőrzés kivárási idő letelte előtt, • kezdeti ellenőrzés kivárási idő után, • utólagos ellenőrzés
szükséges erőforrások	• token (aláíró magánkulccsal) • hálózati kapcsolat	• hálózati kapcsolat
szükséges ismeretek	• az aktivizálandó 2 funkció ismerete • hitelesítő adat	• az aktivizálandó 3 funkció ismerete
Egyéb előfeltételek	• szerződés hitelesítés-szolgáltatóval,	• szerződés időbélyegzés-szolgáltatóval

3. táblázat: munkamegosztás kliens/szerver alapú aláíró/ellenőrző kapcsolat esetén

Az aláíró számára igen előnyös fenti eredmény csak akkor érhető el, ha az aláíró és az ellenőrző a következőket hajtják végre:

1. az aláíró létrehozza az aláírást (XAdES-EPES formátumban),
2. az aláíró elküldi az aláírt dokumentumot az ellenőrzőnek,
3. az ellenőrző a fogadás után haladéktalanul időbélyeget kér és helyez el az aláíráson,
4. az ellenőrző közvetlenül ez után –opcionálisan- kezdeti ellenőrzést hajt végre (kivárási idő letelte előtt),
5. az ellenőrző a kivárási idő letelte után minél hamarabb –kötelezően- kezdeti ellenőrzést hajt végre,
6. az ellenőrző az utólagos ellenőrzésre alkalmassá tett aláírt dokumentumot visszaküldi az aláírónak.

Ennek a munkamegosztásnak van egy hátránya is az aláíró szempontjából: az utólagos ellenőrzésre alkalmas dokumentumot az ellenőrző féltől kapja vissza (a kivárási idő letelte után). Ha ez nem történik meg, nem rendelkezik vitás esetekben használható jogi bizonyítékkal. Ha ez a hátrány nem vállalható, ez a munkamegosztás nem jöhet szóba.

8. Hivatkozások

[1] ETSI TS 101 733 Electronic Signature Formats, v1.5.1, 2003-12

[2] ETSI TS 101 903 XML Advanced Electronic Signatures (XAdES), v1.2.2, 2004-04

[3] RFC 3275 XML-Signature Syntax and Processing (XMLDSIG), March 2002

[4] RFC 3369 XML Cryptographic Message Syntax (CMS), August 2002

[5] CWA 14171:2004 General guidelines for electronic signature verification, 2004-05

9. Rövidítések

CRL	**C**etification **R**evocation **L**ist
MELASZ	**M**agyar **El**ektronikus **A**láírás **Sz**övetség
MMM	**M**ELASZ **M**unkacsoport **M**egállapodás
OCSP	Online Certificate Status Protocol
XAdES	**X**ML **Ad**vanced **E**lectronic **S**ignatures
XML	E**x**tensible **M**arkup **L**anguage