

DigiDoc formaadi kirjeldus

Dokumendi versioon: 1.2,17.03.2003

Kirjeldatav formaadi versioon: 1.2

Käesolev dokument kirjeldab dokumendiformaati, mida kasutavad DigiDoc süsteemi rakendused (edaspidi: DIGIDOC-XML). DigiDoc dokument on esitatud XML kujul ning põhineb rahvusvahelistel standarditel XML-DSIG ja ETSI TS 101 903.

Viited

- RFC2560 Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, C., X.509 Internet Public Key Infrastructure: Online Certificate Status Protocol - OCSP. June 1999.
- RFC3275 Eastlake 3rd D., Reagle J., Solo D., (Extensible Markup Language) XML-Signature Syntax and Processing. (XML-DSIG) March 2002.
- ETSI TS 101 903 XML Advanced Electronic Signatures (XAdES). February 2002.
- XML Schema 2 XML Schema Part 2: Datatypes. W3C Recommendation 02 May 2001 <http://www.w3.org/TR/xmlschema-2/>
- DAS Eesti Digitaalallkirja Seadus

Sissejuhatus

XML Advanced Electronic Signatures (XAdES) [ETSI TS 101 903] defineerib vormingu, mis võimaldab struktuurselt esitada signeeritud andmeid, signatuuri ning muid digitaalallkirjaga seotud turvaatribuute (näiteks kehtivuskinnitusi).

Käesolevas dokumendis kirjeldatakse DigiDoc dokumendiformaati, mis põhineb XAdES standardil ning on alamtüüp standardist.

DigiDoc dokumendiformaadil (DIGIDOC-XML) on järgmised olulised omadused:

- On verifitseeritav vallasresiidimis ilma lisainfot omamata
- Võimaldab anda allkirja mitmele algdokumendile korraga
- Kaitseb formaadiründe vastu – signeeritava dokumendi tüüp sisaldub signatuuris
- Algdokument võib olla kas konteineri sees või sellest eraldi
- Algdokument võib olla XML või suvaline binaarfail (Word, Excel, PDF, RTF jne.)
- Toetatud on mitme signatuuri andmine
- Kehtivuskinnitusi on üks iga signatuuri kohta

Spetsifikatsioonis XAdES kirjeldatud kohustuslikud elemendid ja atribuudid on muudatusteta üle võetud. Mittekohustuslikest XAdES elementidest on DIGIDOC-XML

formaadis esindatud sellised, mis võimaldavad esitada DAS-le vastavat digitaalselt allkirjastatud dokumenti selliselt, et ta oleks verifitseeritav ilma lisainformatsiooni vajamata. Kirjeldataud on nende elementide ja atribuutide väärtustamine.

DIGIDOC-XML dokumendi üldstruktuur

DIGIDOC-XML failiformaadi struktuur on järgmine (kasutatud on [RFC3275] peatükis 2 defineeritud notatsiooni):

```
<?xml version="1.0" encoding="UTF-8" ?>
<SignedDoc format="DIGIDOC-XML" version="1.2">
  <!-- algandmefailid -->
  <DataFile />
  <!-- kliendi allkirjad koos kehtivuskinnitustega -->
  <Signature />
</SignedDoc>
```

Seega kujutab DIGIDOC-XML endast konteinerit <SignedDoc />, milles sisalduvad algandmefailid ja signatuurid.

Algandmed – Üks või enam algandmefaili või viide välisele failile.

Allkirjad – Üks või enam allkirja, mis kinnitavad kõigi dokumendis sisalduvate või välise algandmefaili andmete muutumatus. Kui dokumendis sisaldub (või viitab välisele failile) enam kui üks algandmefaili, siis peab iga allkiri kinnitama kõigi nende dokumentide kontrollkoode ja andmetüüpe. Allkiri sisaldab ka kehtivuskinnituse. Üks kehtivuskinnitus kinnitab täpselt ühe kliendi allkirja kehtivuse soovitud ajamomendil. Ilma kehtivuskinnitusega allkirja ei tohi dokumendile lisada.

DIGIDOC-XML formaadi täielik üldstruktuur on järgmine:

```
<SignedDoc format= version=>
  (<DataFile Id= Filename= ContentType= MimeType= Size= DigestType= DigestValue=
  >)+
  <Signature Id= xmlns=>
    <SignedInfo>
      <CanonicalizationMethod Algorithm= >
      <SignatureMethod Algorithm= >
        (<Reference URI= >
          (<Transforms>
            <Transform Algorithm= >
            </Transforms>)?
          <DigestMethod Algorithm= >
          <DigestValue />
        </Reference>)+
    <SignedInfo xmlns=>
    <SignatureValue Id= >
    <KeyInfo>
      <KeyValue>
        <RSAKeyValue>
          <Modulus />
          <Exponent />
        </RSAKeyValue>
      </KeyValue>
```

```

<X509Data>
  <X509Certificate />
</X509Data>
</KeyInfo>
<Object>
  <QualifyingProperties>
    <SignedProperties xmlns= Id= Target= >
      <SignedSignatureProperties>
        <SigningTime />
        <SigningCertificate>
          <Cert Id= >
            <CertDigest>
              <DigestMethod Algorithm= />
              <DigestValue />
            </CertDigest>
            <IssuerSerial />
          </Cert>
        </SigningCertificate>
        <SignaturePolicyIdentifier>
          <SignaturePolicyImplied/>
        </SignaturePolicyIdentifier>
        (<SignatureProductionPlace>
          <City />
          <StateOrProvince />
          <PostalCode />
          <CountryName />
        </SignatureProductionPlace>)?
        (<SignerRole>
          <ClaimedRoles>
            <ClaimedRole />
          </ClaimedRoles>
        </SignerRole>)?
      </SignedSignatureProperties>
      <SignedDataObjectProperties />
    </SignedProperties>
    <UnsignedProperties Target= >
      <UnsignedSignatureProperties>
        <CompleteCertificateRefs>
          <Cert Id= >
            <CertDigest>
              <DigestMethod Algorithm= />
              <DigestValue />
            </CertDigest>
            <IssuerSerial />
          </Cert>
        </CompleteCertificateRefs>
        <CompleteRevocationRefs>
          <OCSPRefs>
            <OCSPRef>
              <OCSPIdentifier URI= >
                <ResponderID />
                <ProducedAt />
              </OCSPIdentifier>
              <DigestAlgAndValue>
                <DigestMethod Algorithm= />
                <DigestValue />
              </DigestAlgAndValue>
            </OCSPRef>
          </OCSPRefs>
        </CompleteRevocationRefs>
        <CertificateValues>
          <EncapsulatedX509Certificate Id= />
        </CertificateValues>
      </UnsignedProperties>
    </Object>
  </Object>
</Object>

```

```

    <RevocationValues>
      <EncapsulatedOCSPValue Id= >
    </RevocationValues>
  </UnsignedSignatureProperties>
</UnsignedProperties>
</QualifyingProperties>
</Object>
</Signature>
</SignedDoc>

```

Elemendid ja nende parameetrid

Juurelement (SignedDoc)

Iga DigiDoc faili juur-elementiks on **<SignedDoc>**, mis omab järgmisi atribuute:
format – DigiDoc faili formaadi nimi. Aktuaalseks formaadiks on "DIGIDOC-XML". Tuntud on ka vanem formaat "SK-XML" ning formaadi "DIGIDOC-XML" eelmine versioon "1.1"

version – DigiDoc faili formaadi versioon. Aktuaalseks versiooniks on "1.2". Vanema formaadi "SK-XML" puhul on versiooniks "1.0". Versioonide erinevuse kohta on teave saadaval selle dokumendi lisas. Käesolev dokument kirjeldab versiooni 1.2.

Algandmefailid (DataFile)

DigiDoc fail sisaldab ühe või enama algandmefaili või viite väliselt salvestatud failile. Iga faili kohta tehakse kirje **<DataFile>**, mis omab järgmisi atribuute:

- **Id** – faili sisemine unikaalne tunnus. Andmefailide tunnused algavad sümboliga 'D', millele järgneb faili järjekorranumber.
- **Filename** – faili tegelik (väline) nimi ilma teekonnata.
- **ContentType** – dokumendi salvestamise meetod (DETACHED, EMBEDDED_BASE64 või EMBEDDED)
 - **EMBEDDED** - faili andmed on sisestatud algkujul antud kirjes. Kasutatav vaid XML kujul algandmete jaoks. Tähelepanu tuleb osutada sellel, et algandmete XML fail ei sisaldaks XML päist (<?xml ... ?>) ega DTD-d. Siin kirjeldatud XML elemendid ei ole keelatud. Võimalik on ühe faili sisse salvestada algkujul teist DigiDoc faili.
 - **EMBEDDED_BASE64** - faili andmed on sisestatud Base64 kujul antud kirjes.
 - **DETACHED** – algandmed sisalduvad failis, mille nimi on salvestatud atribuudis Filename.
- **MimeType** – algandmete andmetüüp.
- **Size** – tegeliku algandmefaili suurus baitides.
- **DigestType** - algandmefaili räsikoodi tüüp. Esialgu toetatakse vaid sha1 tüüpi. Nõutud vaid DETACHED tüüpi faili puhul.
- **DigestValue** – algandmefaili räsikoodi väärtus Base64 kujul. Räsi arvutatakse

algandmete üle nende originaalkujul. Nõutud vaid DETACHED tüüpi faili puhul.

- Suvaline hulk muid atribuute (metaandmed) kujul <nimi>="<väärtus>".

Allkirjad (Signature)

DIGIDOC-XML fail võib sisaldada suvalisel arvul signatuure. Iga signatuur on esitatud blokis <Signature>, mille põhistruktuuri elementideks on:

- <SignedInfo> - XML-DSIG standardi kohane blokk, mis koondab endasse signeeritava info
- <SignatureValue> - signatuur ise
- <KeyInfo> - sertifikaat, mille alusel signatuur on antud ja selles sisalduv RSA avalik võti.
- <Object> + <QualifyingProperties> - XAdES standardi kohane laienduste blokk, mis omakorda sisaldab:
 - <SignedProperties>+<SignedSignatureProperties> - andmed, mis lähevad täiendavalt signatuuri sisse. Nendeks on:
 - signeerimise aeg (<SigningTime>)
 - sertifikaadiinfo, mille alusel on signatuur antud (<SigningCertificate>)
 - signeerimispoliitika (<SignaturePolicyIdentifier>)
 - signeerimise koht (<SignatureProductionPlace>)
 - signeerija roll (<SignerRole>)
 - <UnsignedProperties>+<UnsignedSignatureProperties> - signeerimata andmed. Nendeks on:
 - kehtivuskinnituse serveri (OCSP responderi) sertifikaadiinfo (<CompleteCertificateRefs>)
 - kehtivuskinnituse enda info (<CompleteRevocationRefs>+<OCSPRefs>)
 - kehtivuskinnituse serveri (OCSP responderi) sertifikaat (<CertificateValues>)
 - kehtivuskinnitus ise (<RevocationValues>)

Signatuuri (Signature) parameetrid

Kliendi allkiri sisaldub elemendis <Signature>, mis omab järgmisi atribuute:

- **Id** – kliendi allkirja unikaalne tunnus. Kliendi allkirjade tunnused algavad sümboliga 'S', millele järgneb allkirja järjekorranumber.
- **xmlns** – XML allkirja namespace. Peab omama väärtust: "http://www.w3.org/2000/09/xmldsig#".

Signeeritava info blokk (SignedInfo)

Kõik kliendi poolt allkirjastatavad andmed sisalduvad <SignedInfo> blokis. Element <SignedInfo> võib omada atribuuti "xmlns" sama sisuga kui <Signature> element. Kui

seada atribuuti ei ole, siis teek lisab ta automaatselt räsi arvutamise ajaks. DigiDoc faili allkirjad on kanoniseeritud ja allkirja meetodiks on alati SHA1 + RSA, mida peegeldab bloki alguse elemendid <CanonicalizationMethod> ja <SignatureMethod> ning mille sisu on fikseeritud kujul järgmine:

```
<CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
```

Järgneb iga algandmefaili kohta üks <Reference> blokki. Iga allkirja kohta tuleb ka üks <Reference> blokk salvestamiseks XAdES laiendustest ettenähtud signeeritava info räsikoodi. Algandmefaili kohta käiv <Reference> blokk sisaldab <DataFile> räsikoodi ja XAdES laiendustes ette nähtud signeeritava info <SignedProperties> bloki räsikoodi. Iga <Reference> element omab atribuuti URI, mille väärtuseks on vastavalt bloki tüübile:

- URI="#<dok-id>" - Algandmete räsikood.
- URI="#<signature-id>-SignedProperties" – XAdES laienduses ette nähtud signatuurile lisatava lisainfo bloki räsikood. Element <Reference>, mis sisaldab elemendi <SignedProperties> räsikoodi peab omama atribuuti "Type" väärtusega "http://uri.etsi.org/01903/v1.1.1#SignedProperties".

Räsikoodi tüübiks on alati sha1 ja allkirja tüübiks detached-signature, s.o. elemendid <Reference> sisaldavad antud XML dokumendis sisalduvate <DataFile> elementide räsikoode. Seda näitavad bloki <Reference> atribuudid URI="<xml-bloki-id>".

Element <Transforms> on vajalik vaid sel juhul kui <Reference> blokk sisaldab DETACHED tüüpi andmefaili räsikoodi. Sel juhul on nimelt elemendis <DataFile> ainult viide välisele failile ja tema räsikood atribuudis DigestValue. Igas <Reference> blokkis on ka <DigestValue> element, mille sisuks on elemendi <DataFile> räsikoodi väärtus Base64 kujul. Siis on vajalik elemendi <Reference> alamelemendid <DigestMethod> ja <Transforms>, mis peavad olema järgmisel kujul:

```
<Transforms>
  <Transform
Algorithm="http://www.sk.ee/2002/10/digidoc#detached-document-
signature"/>
  </Transforms>
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
```

Signatuur ise (SignatureValue)

Allkirja koostamiseks tuleb koguda allkirjastatavate andmete räsikoodid ja salvestada nad ülalkirjeldatud formaadis <SignedInfo> blokki. Viimase üle (ASCII kujul) koostatakse ka allkiri, mille väärtus salvestatakse Base64 kujul <SignatureValue> elemendis. Viimane omab atribuuti id kujul: id="<allkirja-tunnus>-SIG".

Signeerija sertifikaat (KeyInfo)

Kliendi allkirja lõpus on veel allkirjastaja sertifikaat Base64 kujul (PEM formaadis) kahes esituses:

- <KeyValue>/<RSAKeyValue> mis sisaldab identifitseerivaid alamelemente <Modulus> ja <Exponent>
- <X509Data>/<X509Certificate> mis sisaldab sertifikaati ennast BASE64 vormingus.

XAdES laienduste blokk – allkirjastatavad parameetrid (SignedSignatureProperties)

Allkirjale järgnevad allkirja omadused, mis on pakitud <Object> ja <QualifyingProperties> elementide sisse. Omadusi on kahte tüüpi – allkirjastatud ja allkirjastamata omadused.

Allkirjastatud omadused on salvestatud elemendis <SignedProperties> ja kaitstud muudatuste vastu kliendi poolt allkirjastatud räsikoodiga tunnusega URI="#<allkirja-tunnus>-SignedProperties". <SignedProperties> parameetriteks on:

- Id** – allkirjale vastav identifikaator vormingus „<allkirja-tunnus>-SignedProperties“
- Target** – allkirjale viitav väärtus vormingus „#<allkirja-tunnus>“

Allkirjastatud omadustest toetab hetkeline formaat vaid allkirja omadusi, mis on salvestatud elemendi <SignedSignatureProperties> sees:

- Arvuti aeg allkirja koostamisel - <SigningTime>. Salvestatakse dateTime (vt. XML Schema2 p.3.2.7) formaadis: "YYYY.MM.DDTHH24:MM:SS(+/-)TZ:00". Versioonist DIGIDOC-XML 1.1 alates salvestatakse kõik ajad UTC ajas ning ajavööndi väärtuseks on 'Z'.
- Kliendi sertifikaadi info – takistamaks kliendi sertifikaadi vahetamist salvestame elemendis <Cert Id="<allkirja-tunnus>-CERTINFO"> ja allkirjastame kliendi sertifikaadi järgmised atribuudid:
 - sertifikaadi räsikood - <CertDigest>/<DigestValue> - alati SHA1 kood.
 - sertifikaadi tunnus - <IssuerSerial>
- Allkirja kasutamise kord - <SignaturePolicyIdentifier>. Selles XAdES standardi kohustuslikus elemendis kasutab DIGIDOC-XML väärtust <SignaturePolicyImplied />, mis tähendab, et signeerimispoliitika määrab ära mingi välimine kord - näiteks ID-kaardi kontekstis DAS ja SK sertifitseerimispoliitika.
- Mittekohustuslik element allkirjastamise koht - <SignatureProductionPlace>. Siia salvestatakse vabas vormis järgmised andmed:
 - Linna nimi - <City>
 - Maakond - <StateOrProvince>

- Postiindeks - <PostalCode>
- Riigi nimi - <CountryName>
- Mittekohustuslik element allkirjastaja roll – <SignerRole> - allkirjastaja kinnitamata (enda väite kohased) rollid. DIGIDOC-XML kasutab vaid väidetava rolli elementi <ClaimedRoles>. Neid võib olla üks või enam. DIGIDOC-SK vorming lubab seda välja interpreteerida ka kui resolutsiooni, mis dokumendi allkirjastamisel lisatud.

XAdES laienduste blokk – allkirjastamata parameetrid (UnsignedSignatureProperties)

Allkirjastamata omadused on salvestatud elemendis <UnsignedProperties>. Hetkeline formaat toetab vaid allkirjastamata kliendi allkirja omadusi - <UnsignedSignatureProperties>, mille abil salvestatakse viide OCSP responderi sertifikaadile, viide kehtivuskinnitusele ning vastav sertifikaat ja kehtivuskinnitus ise.

Kõigepealt salvestatakse kehtivuskinnituse andja sertifikaadi omadused elemendis <CompleteCertificateRefs>. Elemendis <Cert Id="<allkirja-tunnus>-RESPONDER_CERTINFO"> salvestatakse järgmised sertifikaadi omadused:

- sertifikaadi räsikood - <CertDigest>/<DigestValue> - alati SHA1 kood.
- sertifikaadi tunnus - <IssuerSerial>

Järgneb element <CompleteRevocationRefs>, mis sisaldab kehtivuskinnituse väljaandjat, ajatempli ning räsikoodi alamelemendis /<OCSPRefs>/<OCSPRef>. Kehtivuskinnituse väljaandja andmed on salvestatud elemendis <OCSPIdentifier> atribuudiga:

- URI – viide tegelikule kehtivuskinnitusele kujul #<kehtivuskinnituse-tunnus>

ja alamelementidega:

- ResponderID – OCSP serveri tunnus, näiteks:
"/C=EE/O=ESTEID/OU=OCSP/CN=ESTEID-SK OCSP RESPONDER/emailAddress=pki@sk.ee".
- ProducedAt – Kehtivuskinnituse väljastamise aeg dateTime formaadis:
"YYYY.MM.DDTHH24:MM:SS(+/-)TZ:00".

Kehtivuskinnituse räsikood on salvestatud elemendi <OCSPRef> alamelemendis <DigestAlgAndValue> alamelementidega:

- DigestMethod – määrab räsikoodi arvutamise algoritmi atribuudiga:
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"
- DigestValue – räsikoodi väärtus Base64 kujul.

Seejärel salvestatakse ka kehtivuskinnituse andja kogu sertifikaat Base64 (PEM) kujul

elemendis <CertificateValues> <EncapsulatedX509Certificate Id="<allkirja-tunnus>-RESPONDER_CERT">.

Lõpuks salvestatakse ka kehtivuskinnituse andmed elemendis <RevocationValues>. Sellele järgneb element <EncapsulatedOCSPValue> atribuudiga Id – kehtivuskinnituse tunnus.

Elemendi <EncapsulatedOCSPValue> sisuks ongi OCSP kehtivuskinnitus Base64 kujul.

Erinevused DIGIDOC-XML versioonide 1.0 ja 1.1 vahel

Dokumendiformaadi versioon 1.1 tagab suurema interoperabluse olemasolevate XML-DSIG rakendustega ning parandab mõningaid näpuvigasid võrreldes versiooniga 1.0:

- Eelmises versioonis oli formaadi tunnuseks "SK-XML" ja ainsaks lubatud versiooniks "1.0". Uemas formaadis on tunnuseks "DIGIDOC-XML" ja esialgu ainsaks lubatud versiooniks "1.1".
- Versioonis 1.0 oli elemendil <DataFile> alati nõutud atribuudid DigestType ja DigestValue ning nad sisaldasid elemendi sisu (ilma XML tagideta) räsikoodi. Versioonis 1.1 on nimetatud atribuudid nõutud vaid välisele failile viitava <DataFile> elemendi (atribuut ContentType="DETACHED") puhul ja nad sisaldavad atribuudis Filename määratud välise faili räsikoodi.
- Versioonis 1.0 arvutati räsikoodi elemendi <DataFile> sisu üle selle algkujul. Versioonis 1.1 arvutatakse räsikoodi terve elemendi <DataFile> üle selle kanoniseeritud kujul. Sellest tulenevalt ei ole versioonis 1.1 enam olemas allkirjastamata <DataFile> atribuute. Versioonis 1.0 oli allkirjastatud atribuudiks MimeType ja algandmete endi räsi.
- Versioonis 1.0 pidi signeeritava info <SignedInfo> blokk sisaldama kaks <Reference> blokki iga <DataFile> elemendi kohta. Versioonis 1.1 peab <SignedInfo> sisaldama vaid ühe <Reference> bloki iga <DataFile> elemendi kohta ja selles salvestatud räsikood on arvutatud terve vastava <DataFile> üle knoniseeritud kujul. Kuna versioonis 1.1 on terve <DataFile> allkirjastatud siis pole vaja eraldi <Reference> blokki maimitüübi räsiga.
- Versioonis 1.0 sisaldas iga <Reference> blokk ka XML-DSIG enveloped-signature transformatsioonikirjet. Versioonis 1.1 seda kirjet ei kasutata. Erandiks on välistele failidele viitavad <DataFile> elemendid. (ContentType="DETACHED"). Sel juhul peab vastav <Reference> blokk sisaldam transformatsiooni algoritmiga:
<http://www.sk.ee/2002/10/digidoc#detached-document-signature>.
- Versioonis 1.0 sisaldas element <X509Certificate> atribuuti Id. Versioonis 1.1 seda atribuuti ei kasutata.
- Versioonis 1.1 salvestatakse ka allkirjastaja sertifikaadis sisalduva avaliku võtme moodulus ja eksponent erald elemendi <KeyInfo>/<KeyValue>/<RSAKeyValue> alamelementides <Modulus> ja <Exponent>.
- Versioonis 1.0 oli dateTime struktuuris (kasutusel <SigningTime> ja <ProducedAt>

elementides) oli ajatsooni tähistamisel pluss miinusega vahetuses. Versioonis 1.1 salvestatakse aeg alati GMT ajatsooni ajaks ümberarvutatuna.

- Süntaksiviga <SignedProperties> atribuutil „Id“ – sisaldas ekslikult alguses #-märki
- Süntaksiviga <SigningCertificate> alamelemendil <Cert>, täpsemalt selle atribuudil „Id“ - sisaldas ekslikult alguses #-märki
- DIGIDOC-XML versioon 1.1 on täielikult vastav XML-DSIG standardile aga versioon 1.0 seda ei olnud. Testimiseks sobib näiteks Apache XML Security pakett. Probleeme tekib vaid väliste failide spetsiifilise detached-document-signature transformatsiooniga ja Apache ei kontrolli kehtivuskinnitust jms.

Erinevused DIGIDOC-XML versioonide 1.1 ja 1.2 vahel

- Eelmises versioonis oli formaadi tunnuseks "DIGIDOC-XML" ja ainsaks lubatud versiooniks "1.1". Uemas formaadis on tunnuseks "DIGIDOC-XML" ja lubatud versioonideks "1.1" ja "1.2".
- Eelmises versioonis oli elementidel <SignedInfo> ja <SignedProperties> nõutud atribuudi "xmlns" olemasolu väärtusega: <http://www.w3.org/2000/09/xmldsig#>. Versioonis "1.2" on see atribuut nõutud vaid elemendil <Signature> ja kõigil tema alamelementidel on see lubatud aga mitte nõutud. Räsikoodi arvutamise hetkeks genereeritakse see atribuut xml bloki peamisele elemendile (<SignedInfo> ja <SignedProperties>). See langeb kokku ka Apache XML Security ja .NET käitumisega.
- Versioonis "1.2" on elemendil <Reference>, mis sisaldab elemendi <SignedProperties> räsikoodi nõutud atribuudi: "Type" olemasolu väärtusega: <http://uri.etsi.org/01903/v1.1.1#SignedProperties>.
- Versioonis "1.1" ei hoitud OCSP kehtivuskinnitust tervikuna mälus vaid ainult olulisemaid atribuute. Seetõttu tekkis viga faili salvestamisel ning kehtivuskinnituse ja tema räsi väärtused ei langenud kokku. Versioonis "1.2" on see viga parandatud.

DigiDoc formaadi definitsioon XML Schemas

DigiDoc formaat baseerub standardile ETSI TS 101 903 – XML Advanced Electronic Signatures (XAdES), mis omakorda on standardi XML-DSIG – XML -Signature Syntax and Processing laiendus. DigiDoc profileerib XAdES standardit esitades mõningaid lisanõudmisi. Järgnevalt ka kogu DigiDoc formaadi kirjeldus XML Schema kujul koos selgitustega DigiDoc ja XAdES formaatide erinevuste kohta.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xsd:schema targetNamespace="http://www.sk.ee/DigiDoc/v1.2.0#"
  xmlns="http://www.sk.ee/DigiDoc/v1.2.0#">
```

```

xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
elementFormDefault="qualified" >

<xsd:import namespace="http://www.w3.org/2000/09/xmldsig#"
  schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd" />
<xsd:import namespace="http://uri.etsi.org/01903/v1.1.1#"
  schemaLocation="http://uri.etsi.org/01903/v1.1.1/XAdES.xsd" />

<!-- Root element for SignedDoc -->

<xsd:element name="SignedDoc" type="SignedDocType"/>
<xsd:complexType name="SignedDocType">
  <xsd:sequence>
    <xsd:element name="DataFile" type="DataFileType"
      minOccurs="1" maxOccurs="unbounded"/>
    <xsd:element name="Signature" type="SignatureType"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="format" type="xsd:string" fixed="DIGIDOC-XML"/>
  <xsd:attribute name="version" type="xsd:string" fixed="1.2"/>
</xsd:complexType>

<!-- payload data - DataFile -->

<xsd:complexType name="DataFileType">
  <xsd:attribute name="Id" type="xsd:ID" use="required"/>
  <xsd:attribute name="Filename" type="xsd:string" use="required"/>
  <xsd:attribute name="ContentType">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">

```

```

        <xsd:enumeration value="EMBEDDED"/>
        <xsd:enumeration value="EMBEDDED_BASE64"/>
        <xsd:enumeration value="DETACHED"/>
    </xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
<xsd:attribute name="MimeType" type="xsd:string" use="required"/>
<xsd:attribute name="Size" type="xsd:decimal" use="required"/>
<!-- but required for DETACHED files -->
    <xsd:attribute name="DigestType" type="xsd:string" use="optional"/>
    <xsd:attribute name="DigestValue" type="xsd:string" use="optional"/>
</xsd:complexType>

<!-- Signature according to XML-DSIG and ETSI TS 101 903 -->

<xsd:element name="Signature" type="SignatureType" />
<xsd:complexType name="SignatureType">
    <xsd:sequence>
        <xsd:element name="SignedInfo" type="SignedInfoType"/>
        <xsd:element name="SignatureValue" type="SignatureValueType"/>
        <xsd:element name="KeyInfo" type="KeyInfoType"/>
        <xsd:element name="Object" type="SignatureAttributesType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SignatureValueType">
    <xsd:simpleContent>
<!-- Difference from XML-DSIG: attribute Id is required -->
        <xsd:extension base="xsd:base64Binary">
            <xsd:attribute name="Id" type="xsd:ID" use="required"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

```

```

<xsd:complexType name="SignedInfoType">
  <xsd:sequence>
    <!-- Difference from XML-DSIG: we support only this one canonicalization method -->
    <xsd:element name="CanonicalizationMethod">
      <xsd:complexType>
        <xsd:attribute name="Algorithm" type="xsd:string"
          fixed="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      </xsd:complexType>
    </xsd:element>
    <!-- Difference from XML-DSIG: we support only this one signature method -->
    <xsd:element name="SignatureMethod">
      <xsd:complexType>
        <xsd:attribute name="Algorithm" type="xsd:string"
          fixed="http://www.w3.org/2000/09/xmlsig#rsa-sha1"/>
      </xsd:complexType>
    </xsd:element>
    <xsd:sequence>
      <xsd:element name="Reference" type="ReferenceType"
        minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:element name="Reference" type="SignedPropertiesReferenceType"
      minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ReferenceType">
  <xsd:sequence>
    <!-- Difference from XML-DSIG: we support only one proprietary Transform and only
    for detached datafiles -->
    <xsd:element name="Transforms" type="TransformsType"

```

```

        minOccurs="0" />
<!-- Difference from XML-DSIG: we support only this one digest method -->
    <xsd:element name="DigestMethod">
        <xsd:complexType>
            <xsd:attribute name="Algorithm" type="xsd:string"
                fixed="http://www.w3.org/2000/09/xmldsig#sha1"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="DigestValue" type="ds:DigestValueType"/>
</xsd:sequence>
<!-- Difference from XML-DSIG: URI is required since we use only detached signatures
-->
    <xsd:attribute name="URI" type="xsd:anyURI" use="required"/>
</xsd:complexType>

<xsd:complexType name="SignedPropertiesReferenceType">
    <xsd:complexContent>
        <xsd:restriction base="ReferenceType">
<!-- Difference from version 1.1: attribute Type is required -->
            <xsd:attribute name="Type" type="xsd:anyURI" use="required"
                fixed="http://uri.etsi.org/01903/v1.1.1#SignedProperties"/>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="TransformsType">
    <xsd:sequence>
        <xsd:element name="Transform">
            <xsd:complexType>
                <xsd:attribute name="Algorithm" type="xsd:string"
                    fixed="http://www.sk.ee/2002/10/digidoc#detached-document-signature"/>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

```

```
        </xsd:complexType>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
```

```
<!-- signers public sertificate data -->
```

```
<xsd:complexType name="KeyInfoType">
<!-- Difference from XML-DSIG:
no support for KeyName, RetrievalMethod, PGPDData, SPKIDData, MgmtData
-->
    <xsd:sequence>
        <xsd:element name="KeyValue" type="KeyValue" />
        <xsd:element name="X509Data" type="X509DataType" />
    </xsd:sequence>
</xsd:complexType>
```

```
<xsd:complexType name="KeyValue">
<!-- Difference from XML-DSIG: no support for DSAKeyValue -->
    <xsd:sequence>
        <xsd:element name="RSAKeyValue" type="ds:RSAKeyValue" />
    </xsd:sequence>
</xsd:complexType>
```

```
<xsd:complexType name="X509DataType">
<!-- Difference from XML-DSIG:
no support for X509IssuerSerial, X509SKI, X509SubjectName, X509CRL
-->
    <xsd:sequence>
        <xsd:element name="X509Certificate" type="X509Certificate" />
    </xsd:sequence>
</xsd:complexType>
```

```

<xsd:complexType name="X509CertificateType">
<!-- Difference from XML-DSIG: attribute Id is required -->
  <xsd:simpleContent>
    <xsd:extension base="xsd:base64Binary">
      <xsd:attribute name="Id" type="xsd:ID" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<!-- signed and unsigned signature properties -->

<xsd:complexType name="SignatureAttributesType">
  <xsd:sequence>
    <xsd:element name="QualifyingProperties" type="QualifyingPropertiesType" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="QualifyingPropertiesType">
  <xsd:sequence>
    <xsd:element name="SignedProperties"
      type="SignedPropertiesType" />
    <xsd:element name="UnsignedProperties"
      type="UnsignedPropertiesType" />
  </xsd:sequence>
</xsd:complexType>

<!-- signed properties -->

<xsd:complexType name="SignedPropertiesType">
  <xsd:sequence>
    <xsd:element name="SignedSignatureProperties"

```

```

type="SignedSignaturePropertiesType" />
<!-- Difference from ETSI: we use only empty element, not content for now -->
    <xsd:element name="SignedDataObjectProperties" />
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="required"/>
  <xsd:attribute name="Target" type="xsd:anyURI" use="required"/>
</xsd:complexType>

<xsd:complexType name="SignedSignaturePropertiesType">
  <xsd:sequence>
    <xsd:element name="SigningTime" type="xsd:dateTime" />
    <!-- Difference from ETSI: no list, just one cert info here -->
    <xsd:element name="SigningCertificate"
      type="SigningCertificateType" />
    <!-- Difference from ETSI: we use only empty element, no content for now -->
    <xsd:element name="SignaturePolicyIdentifier"
      type="SignaturePolicyIdentifierType" />
    <xsd:element name="SignatureProductionPlace"
      type="etsi:SignatureProductionPlaceType" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SigningCertificateType">
  <!-- Difference from ETSI: no list, just one cert info here -->
  <xsd:sequence>
    <xsd:element name="Cert" type="CertType" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CertType">
  <xsd:sequence>
    <xsd:element name="CertDigest" type="etsi:DigestAlgAndValueType" />
    <xsd:element name="IssuerSerial" type="ds:X509IssuerSerialType" />
  </xsd:sequence>
</xsd:complexType>

```

```

        </xsd:sequence>
<!-- Difference from ETSI: Id attribute is required -->
        <xsd:attribute name="Id" type="xsd:ID" use="required"/>
</xsd:complexType>

<xsd:complexType name="SignaturePolicyIdentifierType">
    <xsd:sequence>
        <xsd:element name="SignaturePolicyImplied" />
    </xsd:sequence>
</xsd:complexType>

<!-- unsigned signature properties - this contains the OCSP response! -->

<xsd:complexType name="UnsignedPropertiesType">
    <xsd:sequence>
        <xsd:element name="UnsignedSignatureProperties"
            type="UnsignedSignaturePropertiesType" />
    </xsd:sequence>
    <xsd:attribute name="Target" type="xsd:anyURI" use="required"/>
</xsd:complexType>

<xsd:complexType name="UnsignedSignaturePropertiesType">
<!-- Difference from ETSI: we don't support elements CounterSignature,
    SignatureTimeStamp, SigAndRefsTimeStamp, RefsOnlyTimeStamp,
    ArchiveTimeStamp
-->
    <xsd:sequence>
        <xsd:element name="CompleteCertificateRefs"
            type="CompleteCertificateRefsType" />
        <xsd:element name="CompleteRevocationRefs"
            type="CompleteRevocationRefsType" />
        <xsd:element name="CertificateValues"

```

```

        type="CertificateValueType" />
    <xsd:element name="RevocationValues"
        type="RevocationValueType" />
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CompleteCertificateRefsType">
<!-- Difference from ETSI: no list, just one cert info here -->
    <xsd:sequence>
        <xsd:element name="Cert" type="CertType" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CertificateValueType">
    <xsd:sequence>
        <xsd:element name="EncapsulatedX509Certificate">
            <xsd:complexType>
                <xsd:simpleContent>
                    <xsd:extension base="xsd:base64Binary">
<!-- Difference from ETSI: Id attribute is required -->
                        <xsd:attribute name="Id"
                            type="xsd:ID" use="required"/>
                    </xsd:extension>
                </xsd:simpleContent>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="RevocationValueType">
    <xsd:sequence>
        <xsd:element name="EncapsulatedOCSPValue">
            <xsd:complexType>

```

```
        <xsd:simpleContent>
            <xsd:extension base="xsd:base64Binary">
<!-- Difference from ETSI: Id attribute is required -->
                <xsd:attribute name="Id"
                    type="xsd:ID" use="required"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CompleteRevocationRefsType">
<!-- Difference from ETSI: we don't support elements: CRLRefs and OtherRefs -->
    <xsd:sequence>
        <xsd:element name="OCSPRefs" type="etsi:OCSPRefsType" />
    </xsd:sequence>
</xsd:complexType>

</xsd:schema>
```