Kanta CDA R2 -asiakirjojen sähköinen allekirjoitus

Määrittely ja soveltamisopas

4.10.2024
Versio 2.1

Kela
Kanta-palvelut
PL 450, 00056 Kela

FPA
Kanta tjänsterna
PL 450, 00056 FPA

www.kanta.fi
kanta@kanta.fi

Versiohistoria

Versio	Päivämäärä	Kuvaus
1.0	23.6.2014	Ensimmäinen yhdistetty versio oppaasta ja määrityksestä. SHA256 menetelmän käyttöönotto, muiden kuin XML-muotoisten asiakirjojen allekirjoitus
2.0	4.10.2018	SHA512- ja ECC-menetelmien käyttöönotto, sosiaalihuollon asiakirjojen allekirjoitus (Luonnos)
2.1	4.10.2024	Päivitetty käytettävien varmenteiden ja algoritmien lista (luku 1.4) ECC-menetelmien käyttöönotto vuoden 2025 alussa (luku 1.4)

Sisällysluettelo

1 Johdanto

1.1 Dokumentin rakenne

Tämä versio sähköisen allekirjoituksen määräyksestä korvaa CDA R2 -asiakirjojen sähköisistä allekirjoituksista aiemmin annetut määritykset ja soveltamisoppaat.

Luvut 2 ja 3 ovat normatiivisia. Luku 2 käsittelee yleisemmin CDA-rakennetta ja luku 3 määrittää XML-allekirjoitusrakennetta koskevat asiat.

Luvut 4-6 eivät ole normatiivisia.

Tämän määritysversion keskeiset erot määrityksen aikaisempiin versioihin ovat:

- Listattu erilliseen taulukkoon sallitut varmennealgoritmit
    - Allekirjoituksissa käytettyjen varmenteiden tulee olla Digi- ja Väestötietoviraston (jatkossa lyhennettynä DVV) myöntämiä Sosiaali- ja terveydenhuollon järjestelmäallekirjoitusvarmenteita
- Varmenteen tulee olla vähintään RSA3072 SHA512. Muut varmenteen tarvittavat tiedot löytyvät DVV:n Palveluvarmenteiden tekniset tiedot -dokumentin luvusta. 2.7. ECC-varmenteet eivät ole vielä tuettuja Kanta-palveluissa
- Lisätty tuki SHA384 ja SHA512 tiivistefunktiolle
- Asiakirjojen allekirjoituksissa sallitut tiivistefunktiot ovat SHA256, SHA384 ja SHA512
- Valmistaudutaan ECDSA allekirjoitusalgoritmien käyttöönottoon
- Järjestelmien tulee tukea ECC-menetelmän varmenteiden käyttöä asiakirjojen allekirjoituksessa sekä allekirjoitusten tarkastamisessa (ECDSAwithSHA384 ja ECDSAwithSHA512 varmenteet) vuoden 2025 alusta lähtien.
- Määritys kattaa myös sosiaalihuollon asiakirjojen sähköiset allekirjoitukset.

1.2 Sähköisen allekirjoituksen yleiset periaatteet

CDA-asiakirjojen allekirjoitukset perustuvat XML-allekirjoitusstandardiin siten että allekirjoituksen ympärille on toteutettu lisäksi lisätoiminnallisuutta CDA-tason laajennuksina. Laajennuksina toteutetut toiminnot ovat allekirjoitusaika ja moniallekirjoitus. Allekirjoitusaika liittää allekirjoituksen tapahtumahetkeen.

Moniallekirjoitus toteuttaa laissa kuvatun toiminnallisuuden, jossa yksi allekirjoitus allekirjoittaa monta lääkemääräystä yhdellä kertaa[1].

Yksittäinen allekirjoitus ja moniallekirjoitus sisältävät molemmat XML-allekirjoitusrakenteen, joka sisältää kaksi kohdistusta allekirjoitettavaan tietoon. Yksi kohdistuksista osoittaa aikaleimarakenteeseen, toinen asiakirjan tietosisältöön.

Yksittäinen allekirjoitus kohdistuu XML-allekirjoituksesta suoraan asiakirjan tietosisältöön. Moniallekirjoituksessa XML-allekirjoitus kohdistuu moniallekirjoitusrakenteeseen. Moniallekirjoitusrakenne kohdistuu jokaisen samalla kertaa moniallekirjoitetun asiakirjan tietosisältöön.

Kun asiakirjan tietosisältö on CDA-muotoista, on asiakirjan tietosisältö ***cda:structuredBody***-rakenteen alla.

Kun asiakirjan tietosisältö on muussa muodossa kuin CDA (PDF/A tai muu hyväksytty muoto), on asiakirjan tietosisältö ***cda:nonXMLBody***-rakenteen alla.

1.3 Sähköisen allekirjoituksen käyttö eri osa-alueissa

Sähköisen allekirjoituksen käyttö eroaa toisistaan seuraavissa eri käyttötapauksessa:

- KT1) Resepti-palvelun asiakirjat allekirjoitettuna nipussa eli moniallekirjoitettuna
- KT2) Terveydenhuollon XML-muotoinen asiakirja yksittäin allekirjoitettuna
- KT3) Terveydenhuollon PDF-muotoiset asiakirjat
- KT4) Sosiaalihuollon asiakirjat

Näiden käyttötapausten sähköisiin allekirjoituksiin liittyvät erot on koostettu alla olevaan taulukkoon:

Taulukko 1

Ominaisuus	KT1	KT2	KT3	KT4
Allekirjoituksen sijainti osiossa ***hl7fi:localHeader***	x	x	x	
Allekirjoituksen sijainti osiossa ***hl7fi:localSocialHeader***				x
Moniallekirjoitus on käytössä	x			
Allekirjoitus kohdistuu osioon ***hl7fi:signatureTimestamp***	x	x	x	x
Allekirjoitus kohdistuu osioon ***hl7fi:multipleDocumentSignature***	x			
Allekirjoitus kohdistuu osioon ***cda:structuredBody***		x		
Allekirjoitus kohdistuu osioon ***cda:nonXMLBody***			x	x

Taulukon mukaiset vaatimukset esitetään alla kohdassa 3.2

[1] Laki sähköisestä lääkemääräyksestä 2.2.2007/61, 7§ ”Kaikki samaan potilaskäyntiin liittyvät lääkemääräykset voi allekirjoittaa yhdellä allekirjoitustoiminnolla.”

1.4 Allekirjoitukset Kanta-palveluissa

Tietojärjestelmä jossa muodostetaan Kanta-palveluun tallennettava asiakirja lisää asiakirjaan tarvittavat sähköiset allekirjoitukset ennen Kantaan tallentamista.

Sähköiset allekirjoitukset voidaan tuottaa seuraavilla algoritmeilla. Uudet varmenteet tulee olla vähintään kansallisen TL IV kryptovahvuusluokan mukaisia.

Taulukko 2

Algoritmi	Tarkenne/Kuvaus
RSA[2048]	Algoritmin käyttöä uusien asiakirjojen allekirjoittamiseen ei suositella. Uusia allekirjoitusvarmenteita ei tälle algoritmivahvuudelle ei enää myönnetä. Toteutukset VOIVAT käyttää allekirjoitusalgoritmia allekirjoitusten tuottamiseen 2024 käytössä olevien allekirjoitusavainten vanhenemiseen saakka. Toteutusten PITÄÄ tukea allekirjoitusalgoritmia allekirjoitusten tarkastamisessa.
RSA[3072]	**Minimivaatimus** Kansallinen turvallisuusluokka/kryptovahvuus TL IV Allekirjoitus epäsymmetrisellä menetelmällä (esim. RSA) - "vaativat allekirjoitukset" Tuettu algoritmi RS256, RS384 ja RS512 (RSASSA-PKCS1-v1_5) Toteutusten PITÄÄ tukea allekirjoitusalgoritmia allekirjoitusten tuottamiseen. Toteutusten PITÄÄ tukea allekirjoitusalgoritmia allekirjoitusten tarkastamisessa.
RSA[4096]	Allekirjoitus epäsymmetrisellä menetelmällä (esim. RSA) - "vaativat allekirjoitukset" Tuettu algoritmi RS256, RS384 ja RS512 (RSASSA-PKCS1-v1_5) Toteutusten PITÄÄ tukea allekirjoitusalgoritmia allekirjoitusten tuottamiseen. Toteutusten PITÄÄ tukea allekirjoitusalgoritmia allekirjoitusten tarkastamisessa.

Vuoden 2025 aikana Kanta-palveluissa otetaan käyttöön myös ECC-menetelmiin pohjautuvat varmenteet ja niillä toteutetut allekirjoitukset.

ECC-menetelmien käyttöönoton tarkempi aikataulu tiedotetaan erikseen. Käytetyt ECC-menetelmän avainpituudet on esitetty taulukossa 3.

Taulukko 3

Algoritmi	Tarkenne/Kuvaus
ECDSA[256]	**Minimivaatimus** Kansallinen turvallisuusluokka/kryptovahvuus TL IV Elliptic Curve Digital Signature Algorithm, allekirjoitus epäsymmetrisellä menetelmällä (esim. RSA) - "vaativat allekirjoitukset" Tuettu algoritmi ECDSA ES256 (EC P-256 DSA with SHA-256) Toteutusten PITÄÄ tukea allekirjoitusalgoritmia allekirjoitusten tuottamiseen. Toteutusten PITÄÄ tukea allekirjoitusalgoritmia allekirjoitusten tarkastamisessa.
ECDSA[384]	Elliptic Curve Digital Signature Algorithm, allekirjoitus epäsymmetrisellä menetelmällä (esim. RSA) - "vaativat allekirjoitukset" Tuettu algoritmi ECDSA ES384 (EC P-384 DSA with SHA-384) Toteutusten PITÄÄ tukea allekirjoitusalgoritmia allekirjoitusten tuottamiseen. Toteutusten PITÄÄ tukea allekirjoitusalgoritmia allekirjoitusten tarkastamisessa.

Allekirjoituksessa voidaan käyttää seuraavia tiivistealgoritmeja:

Taulukko 4

Algoritmi	Tarkenne/Kuvaus
SHA-2: SHA-256	**Minimivaatimus** Kansallinen turvallisuusluokka/kryptovahvuus TL IV. Tiivistefunktio, käyttötarkoitus digitaaliset allekirjoitukset ja "hash-only" -sovellukset. Toteutusten PITÄÄ tukea tiivistealgoritmia allekirjoitusten tuottamiseen. Toteutusten PITÄÄ tukea tiivistealgoritmia allekirjoitusten tarkastamisessa.
SHA-2: SHA-384	Tiivistefunktio, käyttötarkoitus digitaaliset allekirjoitukset ja "hash-only" -sovellukset. Toteutusten PITÄÄ tukea tiivistealgoritmia allekirjoitusten tuottamiseen. Toteutusten PITÄÄ tukea tiivistealgoritmia allekirjoitusten tarkastamisessa.
SHA-2: SHA-512	Tiivistefunktio, käyttötarkoitus digitaaliset allekirjoitukset ja "hash-only" -sovellukset. Toteutusten PITÄÄ tukea tiivistealgoritmia allekirjoitusten tuottamiseen. Toteutusten PITÄÄ tukea tiivistealgoritmia allekirjoitusten tarkastamisessa.

Kun Kanta vastaanottaa tallennettavan asiakirjan, se tarkistaa allekirjoitukset. Tätä seuraava toimintatapa riippuu käytettävästä Kanta-palvelusta:

- **Resepti-palvelu** allekirjoittaa asiakirjan omalla järjestelmäallekirjoituksellaan joka lisätään asiakirjaan ennen sen tallentamista (Kanta-allekirjoitus).
- **Potilastiedon tietovaranto** ei lisää uusiin asiakirjoihin Kanta-allekirjoituksia. Vanhoissa asiakirjoissa on Kanta-allekirjoituksia.
- **Sosiaalihuollon asiakastiedon tietovaranto** ei lisää asiakirjoihin Kanta-allekirjoituksia.

Kun tietojärjestelmä noutaa asiakirjan Kanta-palvelusta, riippuu allekirjoituksien tarkastusvelvollisuus käytettävästä Kanta-palvelusta seuraavasti:

- **Resepti-palvelu** - asiakirjassa on rinnakkain alkuperäinen allekirjoitus ja Kanta-allekirjoitus. Noudetun asiakirjan allekirjoituksista tarkistetaan vain Kanta-allekirjoitus.
- **Potilastiedon tietovaranto** - asiakirjassa on joko pelkkä alkuperäinen allekirjoitus tai rinnakkain alkuperäinen allekirjoitus ja Kanta-allekirjoitus. Noudetun asiakirjan allekirjoituksia ei tarkisteta.
- **Sosiaalihuollon asiakastiedon tietovaranto** - asiakirjassa on pelkkä alkuperäinen allekirjoitus. Noudetun asiakirjan allekirjoitusta ei tarkisteta.

Jos järjestelmä tarkistaa Potilastiedon tietovarannosta noudetun asiakirjan allekirjoituksen, tulee järjestelmän varautua siihen, että asiakirjan allekirjoituksessa käytetty varmenne ei ole enää voimassa.

1.5 Dokumentissa käytetyt merkinnät

Sähköiseen allekirjoitukseen liittyvät osuudet CDA-dokumentissa ovat kolmen eri nimiavaruuden (namespace) alla. Lisäksi allekirjoituksiin liittyy rakenteita, joiden tietotyypit on määritelty XML Schemassa. Tässä määrityksessä käytetään selvyyden vuoksi elementeistä ja attribuuteista etuliitteitä sen mukaan missä nimiavaruudessa ne ovat. Käytetyt etuliitteet ja näitä vastaavat nimiavaruudet ovat:

Taulukko 5

Etuliite (prefix)	Nimiavaruus (namespace)
hl7fi	urn:hl7finland
ds	http://www.w3.org/2000/09/xmldsig#
cda	urn:hl7-org:v3
xs	http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

Tässä määrityksessä käytetään esimerkeissä pelkistettyä CDA-rakennetta, jolla pyritään korostamaan allekirjoitukseen vaikuttavia keskeisiä rakenteita. Selkeyden vuoksi muut osat on piilotettu ...-merkin taakse.

Määritysosiossa käytetään Internet Engineering Task Forcen RFC 2119 suosituksen[2] terminologiaa seuraavasti:

- PITÄÄ, PAKOLLINEN (MUST, REQUIRED, SHALL): Määritys tai sääntö ilmaisee ehdottoman vaatimuksen.
- EI SAA (MUST NOT, SHALL NOT): Määritys tai sääntö ilmaisee ehdottoman kiellon.
- PITÄISI (SHOULD, RECOMMENDED): Määritys tai sääntö ilmaisee käytännön, jota tulee noudattaa, ellei ole hyvää syytä toimia toisin. Suosituksesta poikkeavan ratkaisun vaikutukset on syytä ymmärtää ennen poikkeavan ratkaisun tekemistä.
- EI PITÄISI (SHOULD NOT, NOT RECOMMENDED): Määritys tai sääntö ilmaisee, ettei määritellyllä tavalla tule toimia, ellei siihen ole hyvää syytä. Jos määritellyllä tavalla kuitenkin perustellusti toimitaan, ratkaisun vaikutukset on syytä ymmärtää ennen ratkaisun tekemistä.
- SAA, VALINNAINEN (MAY, OPTIONAL): Määritys tai sääntö ilmaisee toimintatavan, joka on sallittu mutta ei pakollinen. Tietojärjestelmän toteuttaja voi harkintansa mukaan noudattaa toimintatapaa tai olla noudattamatta sitä.

[2] RFC 2119: Key words for use in RFCs to Indicate Requirement Levels. Scott Bradner, maaliskuu 1997. *https://www.ietf.org/rfc/rfc2119.txt*

2 CDA-asiakirjojen sähköinen allekirjoitus

2.1 CDA-allekirjoituksen rakenne

Suomessa terveydenhuollossa käytettävät CDA R2 -dokumentin paikalliset laajennukset ovat CDA Headerin lopussa ***hl7fi:localHeader***-elementin alla. Sosiaalihuollossa vastaava elementti on ***hl7fi:localSocialHeader***. Sähköiset allekirjoitukset ovat molemmissa tapauksissa **hl7fi:signatureCollection**-elementin alla.

hl7fi:signatureCollection-elementti sisältää nolla tai useampia ***hl7fi:signature***-elementtejä joista kukin sisältää yhden allekirjoituksen tiedot. Kaikki erityyppiset allekirjoitukset sisältävät elementit ***hl7fi:signatureDescription***, ***hl7fi:signatureTimestamp*** ja ***ds:Signature***. Moniallekirjoitus sisältää lisäksi elementin ***hl7fi:multipleDocumentSignature***.

ds:Signature-rakenne sisältää kaksi ***ds:Reference***-elementtiä, joista yksi kohdistuu aina aikaleimaan (***hl7fi:signatureTimestamp***-elementti). Toinen ***ds:Reference***-elementti kohdistuu yksittäisissä allekirjoituksissa ***cda:structuredBody***-elementtiin tai ***cda:nonXMLBody***-elementtiin, ja moniallekirjoituksissa ***hl7fi:multipleDocumentSignature***-elementtiin.

```
<cda:ClinicalDocument xmlns:cda="urn:hl7-org:v3">
  ...
  <cda:id root="1.2.246.10.21.93.2014.1"/>
  ...
  <hl7fi:localHeader>
  ...
    <hl7fi:signatureCollection>
      <hl7fi:signature ID="CDA-allekirjoitus">
        <hl7fi:signatureDescription code="1" codeSystem="1.2.246.537.5.40127.2006"/>
        <hl7fi:signatureTimestamp ID="CDA-aikaleima">2014-05-14T09:30:01+02:00</hl7fi:signatureTimestamp>
        <ds:Signature Id="XML-allekirjoitus" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
          ...
        </ds:Signature>
      </hl7fi:signature>
    </hl7fi:signatureCollection>
  </hl7fi:localHeader>
  <cda:component>
    <cda:structuredBody id="CDA-allekirjoituksen-kohde">...</cda:structuredBody>
  </cda:component>
</cda:ClinicalDocument>
```

Kuva 1 Pelkistetty esimerkki sähköisestä allekirjoituksesta CDA R2-asiakirjassa

CDA-allekirjoituksen rakenne ("?" tarkoittaa nolla tai yksi ja "*" nolla tai useampi):

```
<hl7fi:signatureCollection>
 (<hl7fi:signature ID>
   <hl7fi:signatureDescription/>
   <hl7fi:signatureTimestamp ID/>
   (<hl7fi:multipleDocumentSignature ID>)?
   <ds:Signature/>
 </hl7fi:signature>)*
</hl7fi:signatureCollection>
```

(***ds:Signature*** on XML-allekirjoituksen rakenteen mukainen)

hl7fi:signatureDescription-elementti kuvaa allekirjoituksen tyypin. Tyypin kuvaamiseen käytettävä koodisto on: ”Kanta-palvelut - Sähköisen

allekirjoituksen tyyppi" ja sen OID-tunnus on 1.2.246.537.5.40127.2006. Koodisto jaellaan kansallisen koodistopalvelun kautta muiden vastaavien koodistojen tavoin osoitteessa http://koodistopalvelu.kanta.fi/.

Esimerkki yksittäisen allekirjoituksen ***hl7fi:signatureDescription***-elementistä:

```
<hl7fi:signatureDescription code="1"
        codeSystem="1.2.246.537.5.40127.2006"
        codeSystemName="Kanta-palvelut - Sähköisen allekirjoituksen tyyppi"
        displayName="Ammattihenkilön allekirjoitus"/>
```

Esimerkki koodiston 1.2.246.537.5.40127 (Kanta-palvelut - Sähköisen allekirjoituksen tyyppi) arvolistasta:

Id	Short name
1	Ammattihenkilön allekirjoitus
2	Ammattihenkilön moniallekirjoitus
3	Järjestelmäallekirjoitus
4	Kanta-järjestelmäallekirjoitus
5	Asiakkaan sähköinen allekirjoitus

hl7fi:signatureTimestamp-elementti sisältää kellonajan sekunnin tarkkuudella. Elementti on tyyppiä ***xs:dateTime***[3] ja sillä on pakollinen attribuutti ***ID***. Aikaleiman muodostaminen on kuvattu yksityiskohtaisemmin luvussa 4.2.

Esimerkki ***hl7fi:signatureTimestamp***-elementistä:

```
<hl7fi:signatureTimestamp ID="TSid001">2008-11-21T12:18:06Z</hl7fi:signatureTimestamp>
```

hl7fi:multipleDocumentSignature-elementti sisältää viittaukset moniallekirjoituksen kohteena oleviin CDA-asiakirjoihin joista jokaiseen liitetään kopio samasta moniallekirjoituksesta. Elementillä on attribuutti ***ID***. Kukin viittaus on oma ***hl7fi:Ref***-elementtinsä jonka ***OID***-attribuutti on kohteena olevan CDA-asiakirjan yksilöintitunnus (OID, elementistä ***cda:ClinicalDocument/cda:id***) ja ***hash***-attribuutissa kyseisen asiakirjan tietosisällöstä (***cda:structuredBody***- tai ***cda:nonXMLBody***-rakenne) laskettu tiiviste. Tiivisteen laskemisessa käytetään samoja kanonikalisointi- ja tiivistealgoritmeja kuin moniallekirjoitusrakenteeseen kohdistuvassa allekirjoituksessa.

Esimerkki ***hl7fi:multipleDocumentSignature***-elementistä:

```
<hl7fi:multipleDocumentSignature ID="MDSid001">
   <hl7fi:Ref OID="1.2.246.10.98765432.93.2009.1" hash="ii2inzvingiirkmGQXiiWj72ggRg/jYhWizy0M8CzIE="/>
   <hl7fi:Ref OID="1.2.246.10.98765432.93.2009.2" hash="RicmXiX1iik0iHiAnGXq94+Lieijq9y+gf0s6ofcs1Y="/>
</hl7fi:multipleDocumentSignature>
```

Sähköisen allekirjoituksen skeematiedosto on osa CDA R2 Header -määrittelyä terveydenhuollossa ja osa asiakirjastandardin määrittelyä sosiaalihuollossa.

[3] XML Schema Part 2: Datatypes Second Edition. W3C Recommendation 28 October 2004, *http://www.w3.org/TR/xmlschema-2/#dateTime*

Sähköisen allekirjoituksen rakenne on skeematiedostossa hl7fi_extensions_cdar2header.xsd (terveydenhuollossa) tai hl7fi_extensions_cdar2header_sos.xsd (sosiaalihuollossa).

Kansainvälisessä CDA-standardissa ***cda:structuredBody***-elementillä ei ole ***ID***-attribuuttia. Suomen HL7-yhdistyksen viralliseen CDA R2-skeemaan on lisätty ***ID***-attribuutti (tyyppiä ***xs:id***). Vastaavasti myös ***cda:nonXMLBody***-elementille on määritetty ***ID***-attribuutti kansallisena laajennuksena.

XML-allekirjoitusstandardi määrittää kolme erilaista allekirjoitustyyppiä sen mukaan miten sähköinen allekirjoitus sijoittuu suhteessa allekirjoituksen kohteena olevaan sisältöön. CDA R2 -asiakirjoissa käytettävä allekirjoitustyyppi on detached[4].

2.2 Allekirjoituksen aikaleima

hl7fi:signatureTimestamp-elementin tietosisältö sisältää allekirjoituksen ajankohdan sekunnin tarkkuudella. Käytetty ajan esitystapa noudattaa tyyppiä ***xs:dateTime***[5]

Esimerkkejä hl7fi:signatureTimestamp-elementistä:

```
<hl7fi:signatureTimestamp ID="TSid001">2018-09-11T20:18:06Z</hl7fi:signatureTimestamp>
<hl7fi:signatureTimestamp ID="TSid002">2018-09-11T22:18:06+02:00</hl7fi:signatureTimestamp>
<hl7fi:signatureTimestamp ID="TSid003">2018-09-07T07:07:07+03:00</hl7fi:signatureTimestamp>
<hl7fi:signatureTimestamp ID="TSid004">2018-09-07T04:07:07Z</hl7fi:signatureTimestamp>
```

Esimerkin ensimmäinen ja toinen sekä kolmas ja neljäs rivi kuvaavat keskenään samaa aikaa.

Aikaleimassa voidaan ilmaista myös sekunnin murto-osat tai aikavyöhyke. Aikavyöhyke ilmoitetaan erotuksena UTC-aikaan.

On suositeltavaa, että järjestelmien kello on synkronoitu NTP-protokollan avulla oikeaan aikaan. NTP-palvelimia on tarjolla sekä ilmaiseksi että kaupallisten toimijoiden toimesta. Mittatekniikan keskus MIKES tarjoaa Suomen viralliseen aikaan synkronoitua NTP-palvelua eri tasoilla.

Moniallekirjoituksessa kaikilla yhdellä kertaa allekirjoitetuilla asiakirjoilla on sama aikaleima.

Sähköinen allekirjoitus kohdistuu aikaleimarakenteeseen. Tästä seuraa, että aikaleima pitää muodostaa ennen allekirjoittamista ja että aikaleiman sisältöä ei saa muokata allekirjoittamisen jälkeen.

Kun asiakirjassa on useampi kuin yksi sähköinen allekirjoitus, on asiakirjassa myös enemmän kuin yksi ***hl7fi:signatureTimestamp***-elementti. Jos allekirjoituksen kohdistuksessa ei rajoiteta allekirjoitettavaa aikaleimaa käyttäen

[4] detached-muoto sallisi allekirjoituksen sijoittamisen eri tiedostoon kuin missä allekirjoitettava tietosisältö on, mutta tämä ominaisuus ei ole käytössä CDA R2-asiakirjojen allekirjoittamisessa. detached-muodon määritelmä: *http://www.w3.org/TR/xmldsig-core/#def-SignatureDetached*

[5] XML Schema Part 2: Datatypes Second Edition. W3C Recommendation 28 October 2004, *http://www.w3.org/TR/xmlschema-2/#dateTime*, luku 3.2.7

ID-elementin arvoa, ei allekirjoitus ole enää eheä sen jälkeen, kun asiakirjaan lisätään mahdollisesti muita allekirjoituksia.

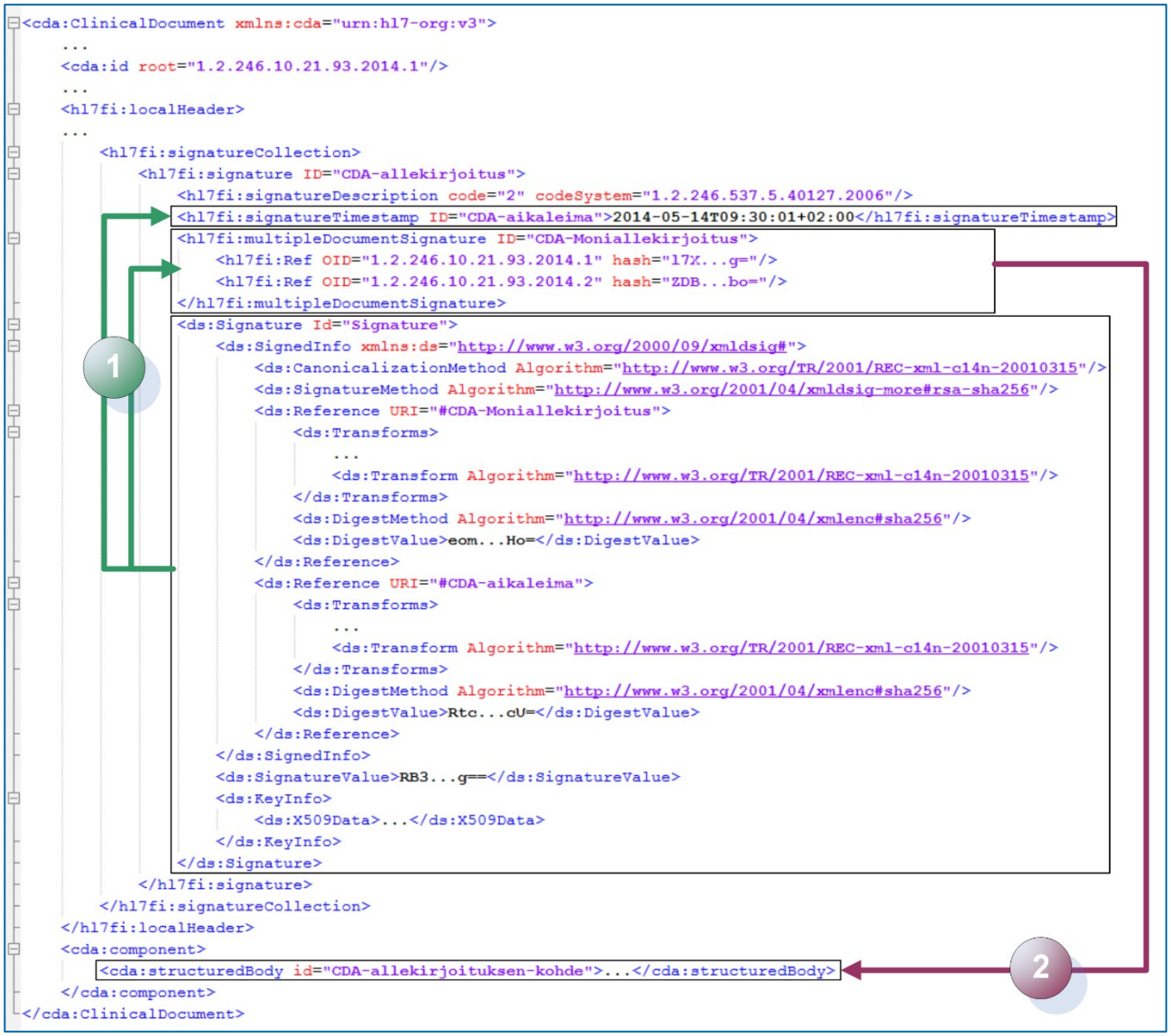
2.3 Moniallekirjoituksen rakenne

Moniallekirjoitus eroaa yksittäisestä allekirjoituksesta seuraavilta osin:

- ***hl7fi:signatureDescription***-elementissä määritelty allekirjoituksen tyyppi on arvoltaan 2, eli ammattihenkilön moniallekirjoitus.
- käytössä on ***hl7fi:multipleDocumentSignature***-elementti
- toinen ***ds:Reference***-elementeistä ei kohdistu asiakirjan tietosisältöön (***cda:structuredBody***- tai ***cda:nonXMLBody***-rakenne), vaan ***hl7fi:multipleDocumentSignature***-elementtiin
- kaikki yhdellä kertaa moniallekirjoitetut asiakirjat sisältävät saman ***hl7fi:signatureCollection***-elementin. Erityisesti on huomioitavaa, että ***hl7fi:signatureTimestamp***-elementti ja sen sisältämä aika on sama kaikissa asiakirjoissa.

Tästä seuraa se, että allekirjoituksen XML-allekirjoitusosuus ei enää takaa suoraan varsinaisen tietosisällön eheyttä. Tietosisällön eheyden takaaminen tapahtuu ***hl7fi:multipleDocumentSignature***-elementin sisältämän ***hl7fi:Ref***-elementin kautta vastaavilla menetelmillä kuin yksittäisessä allekirjoituksessa. Kohteena olevan sisällön muuttumattomuuden takaa moniallekirjoitusrakenteeseen muodostettu tiiviste, jonka muuttamattomuuden takaa XML-allekirjoitus.

Alla (Kuva 2) on esitetty allekirjoituksen kohdistuminen ja eheyden takaaminen moniallekirjoituksessa. 1-Nuolet kuvaavat XML-allekirjoituksen sisältämiä kohdistuksia. 2-Nuoli kuvaa moniallekirjoitusrakenteen sisältämää kohdistusta.



Kuva 2 Moniallekirjoituksen kohdistumiset - allekirjoitus takaa nuolen kohteina olevien alueiden muuttumattomuuden

2.4 XML-allekirjoituksen kohdistuminen

Seuraavassa on esimerkki kahdesta sallitusta kohdistustavasta:

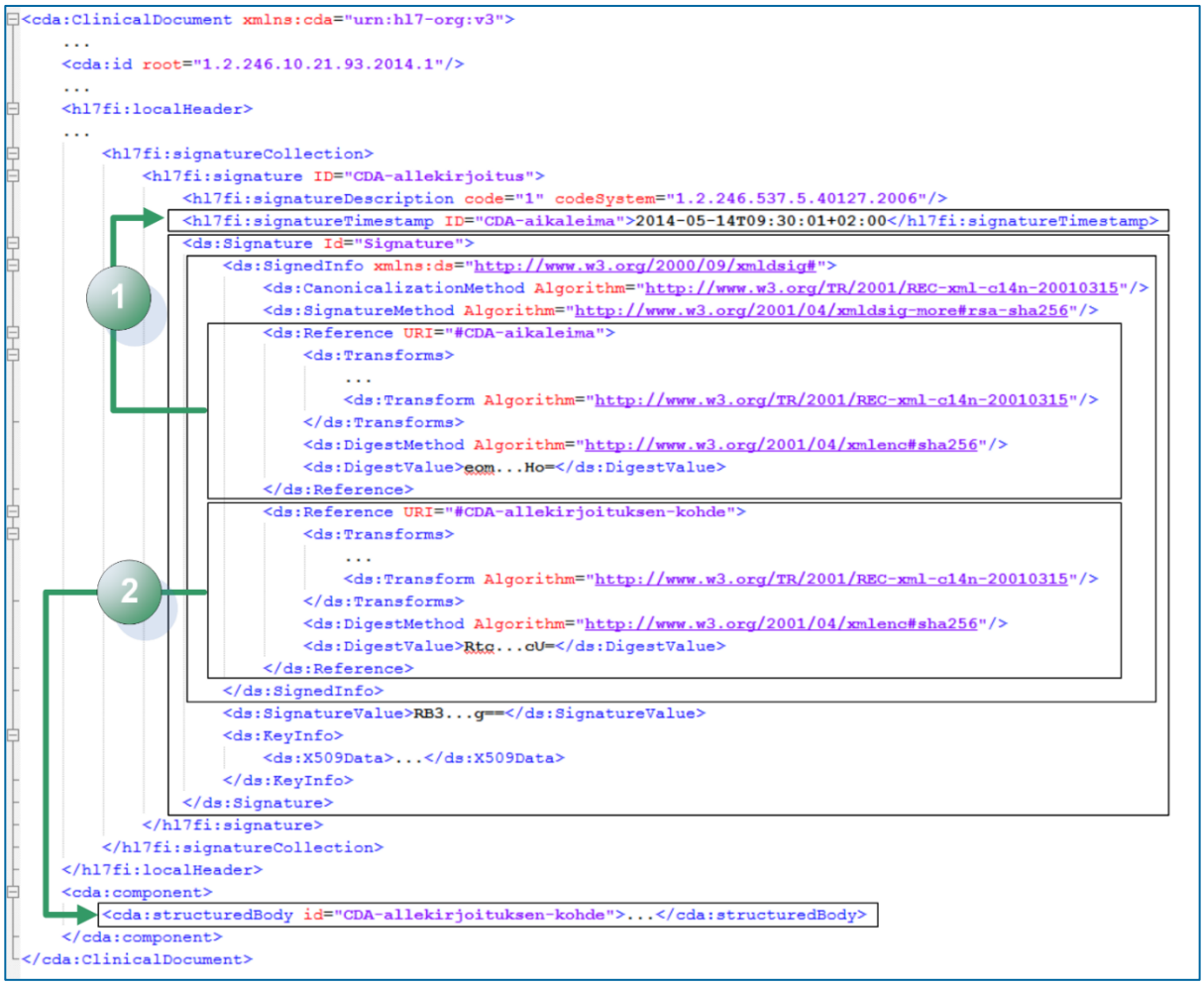
- Suora kohdistus ds:Reference-elementillä ID-attribuuttiin (jatkossa Reference-kohdistus)

```
<ds:Reference URI="#TSid001">…</ds:Reference>
<ds:Reference URI="#MDSid001">…</ds:Reference>
```

- Kohdistus juureen ja allekirjoitettavan tiedon suodattaminen Filter2-suodatuksella (jatkossa Filter2 -kohdistus)

```
<ds:Reference URI="">
  <ds:Transforms>
    <ds:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
      <dsig-xpath:XPath xmlns:dsig-xpath=”http://www.w3.org/2002/06/xmldsig-filter2” Filter="intersect">//*[local-name()='ClinicalDocument']/*[local-name()='localHeader']/*[local-name()='signatureCollection']/*[local-name()='signature']/*[local-name()='signatureTimestamp'][@ID=' TSid001']</dsig-xpath:XPath>
    </ds:Transform>
  </ds:Transforms>…
</ds:Reference>
<ds:Reference URI="">
  <ds:Transforms>
    <ds:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
      <dsig-xpath:XPath xmlns:dsig-xpath=”http://www.w3.org/2002/06/xmldsig-filter2” Filter="intersect">//*[local-name()='ClinicalDocument']/*[local-name()='component']/*[local-name()='structuredBody']</dsig-xpath:XPath>
    </ds:Transform>
  </ds:Transforms>…
</ds:Reference>
```

Alla on esitetty yksittäisen sähköisen allekirjoituksen kohdistuminen (Kuva 3).



Kuva 3 yksittäisessä allekirjoituksessa XML-allekirjoitus kohdistuu aikaleimaan (1) ja asiakirjan sisältöön (2)

2.5 Moniallekirjoituksen kohdistuminen

Moniallekirjoitusrakenteen ***hl7fi:Ref***-rakenne vastaa käyttötarkoitukseltaan XML-allekirjoituksen ***ds:Reference***-rakennetta. ***ds:Reference***-rakenteessa käytetty kohdistaminen erilaisine vaihtoehtoisine parametreineen on kuvattu luvussa 4.2.

Tässä esitetään moniallekirjoitusrakenne siten että se kohdistuu CDA-sisältöisiin asiakirjoihin. Eli kohteena on ***cda:structuredBody***, ei ***cda:nonXMLBody***-rakenne.

hl7fi:Ref-elementin osoittaman rakenteen sijainti, käytettävä tiivistefunktio ja käytettävät suodatukset määräytyvät seuraavasti:

hl7fi:Ref-elementin kohteena oleva XML-rakenne on ***OID***-attribuutin arvoa vastaavan CDA R2 -asiakirjan ***cda:structuredBody*** ja tämän alipuu. Kohteesta muodostettu tiiviste tallennetaan ***hash***-attribuutin arvoksi.

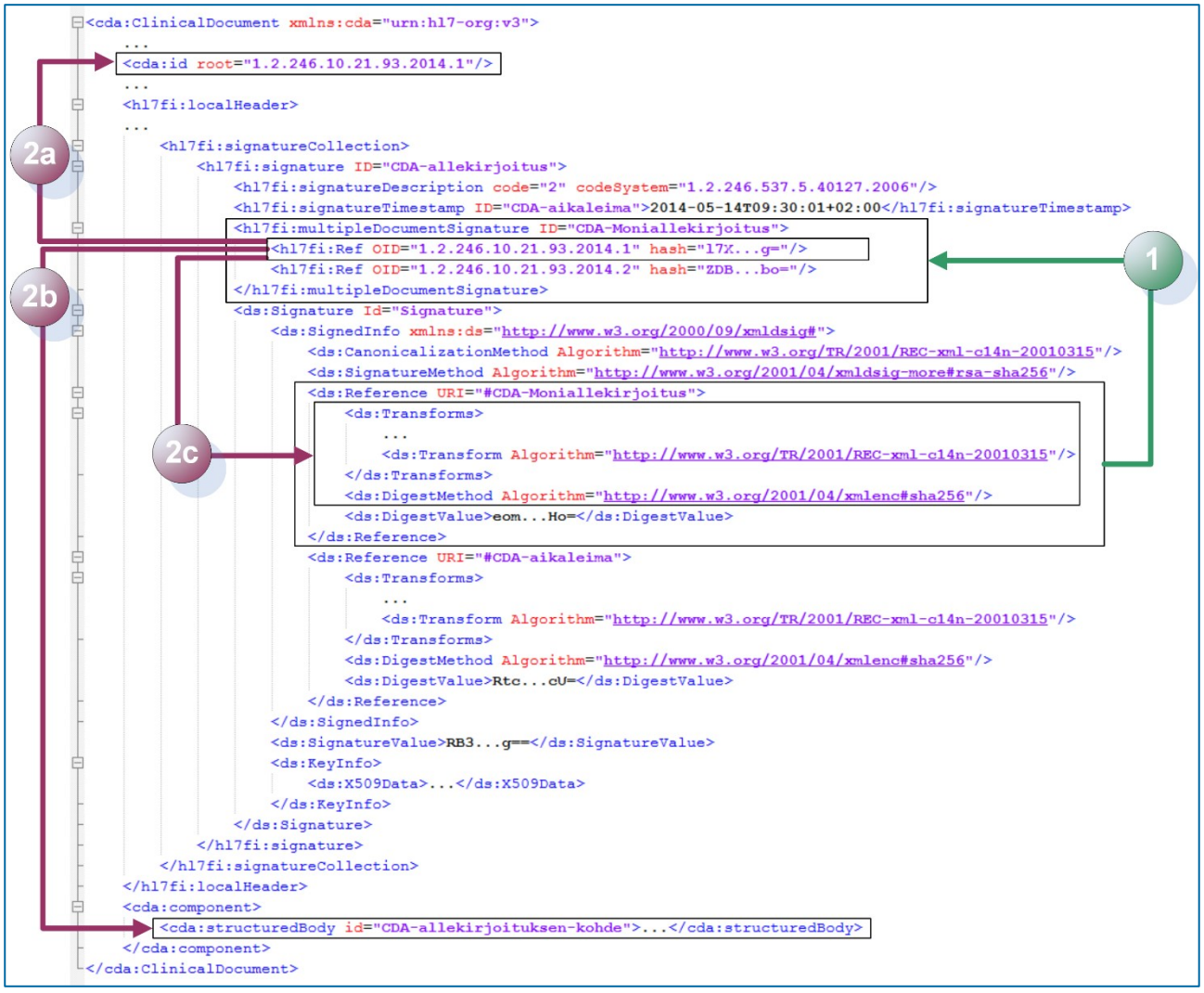
CDA-dokumentin yksilöintitunnuksena käytetty OID sijaitsee asiakirjan ***cda:id***-solmussa. OID muodostetaan attribuuttien ***root*** ja ***extension*** arvoista, jotka erotetaan pisteellä. Mikäli ***extension***-attribuuttia ei ole käytetty, OID on sama kuin attribuutin ***root*** arvo. Oikean dokumentin valinta ja ***cda:structuredBody***-rakenteen kohdistamiseen käytettävä menetelmä ovat toteutuskohtaisesti vapaasti valittavissa sallittujen menetelmien joukosta.

cda:structuredBody-rakenteen suodattamiseen käytetään ***hl7fi:multipleDocumentSignature***-elementtiin kohdistuneen ***ds:Reference***-rakenteen mukaisia ***ds:Transform***-solmujen menetelmiä. Luvussa 3.4 on esitetty ne menetelmät, joita tämä koskee (Taulukko 7).

ds:Transform-solmujen menetelmiä sovellettaessa tulee huomioida se, että menetelmien järjestyksellä on merkitystä[6]. Menetelmät tulee soveltaa samassa järjestyksessä kuin ne sovelletaan XML-allekirjoituksessa. Erityisesti tulee huolehtia siitä, että kanonikalisointi suoritetaan menetelmistä viimeisenä ennen tiivisteen laskemista.

Moniallekirjoituksen kohdistuminen on esitetty alla (Kuva 4). Nuoli 2a kuvaa ***hl7fi:Ref***-solmun ***OID***-elementin mukaista viittausta dokumentin ***cda:id***-solmuun. Nuoli 2b kuvaa edellisen nuolen mukaista viittausta ***cda:structuredBody***-rakenteeseen. Nuoli 2c kuvaa moniallekirjoituksen riippuvuutta ***ds:Reference***-rakenteen ***ds:Transform***-rakenteista.

[6] Juuri ennen tiivisteen laskemista tehtynä kanonikalisointi takaa yhdenmukaisen rakenteen esitystavan. Muiden menetelmien osalta rakenteen esitystapa eri ympäristöissä voi vaihdella.



Kuva 4 moniallekirjoitusrakenne on riippuvainen punaisten nuolten kohteista

2.6 Yksittäisen CDA-asiakirjan allekirjoituksen muodostaminen ja tarkastaminen

Yksittäisen allekirjoituksen muodostaminen tapahtuu seuraavasti:

1. Muodostetaan uusi hl7fi:signature-elementti jonka sisältö on seuraava:

   - hl7fi:signatureDescription-elementti on yksittäisen allekirjoituksen mukainen:

```
<hl7fi:signatureDescription code="1"
        codeSystem="1.2.246.537.5.40127.2006"
        codeSystemName="Kanta-palvelut - Sähköisen allekirjoituksen tyyppi"
        displayName="Ammattihenkilön allekirjoitus"/>
```

   - ***hl7fi:signatureTimestamp***-elementti muodostetaan vähän ennen allekirjoittamista (korkeintaan sekunteja ennen)

- ***ds:Signature***-elementti sisältää sähköisen allekirjoituksen joka kohdistuu
  - Aikaleimarakenteeseen
    - Aina ***hl7fi:signatureTimestamp***
  - Asiakirjan sisältöön
    - ***cda:structuredBody***- tai ***cda:nonXMLBody***-***rakenne***.

2. Lisätään muodostettu hl7fi:signature-elementti allekirjoitettuun CDA-asiakirjaan

Allekirjoituksen tarkistaminen tapahtuu seuraavasti:

1. Tarkistetaan CDA-asiakirja joka sisältää allekirjoituksen XML-allekirjoitusstandardin toteuttavalla validaattorilla.

2.7 Moniallekirjoituksen muodostaminen ja tarkastaminen

Moniallekirjoituksen muodostamisessa ja tarkistamisessa on yksi lisäkerros välissä verrattuna tavalliseen allekirjoitukseen.

Moniallekirjoituksen muodostaminen tapahtuu seuraavasti:

1. Muodostetaan uusi ***hl7fi:signature***-elementti jonka sisältö on seuraava:

- ***hl7fi:signatureDescription***-elementti on moniallekirjoituksen mukainen:

```
<hl7fi:signatureDescription code="2"
        codeSystem="1.2.246.537.5.40127.2006"
        codeSystemName="Kanta-palvelut - Sähköisen allekirjoituksen tyyppi"
        displayName="Ammattihenkilön moniallekirjoitus"/>
```

- ***hl7fi:signatureTimestamp***-elementti on samanlainen kaikissa eri allekirjoituksissa.
- ***hl7fi:multipleDocumentSignature***-elementti (tarkempi kuvaus alla)
- ***ds:Signature***-elementti sisältää sähköisen allekirjoituksen joka kohdistuu **hl7fi:signatureTimestamp**- ja ***hl7fi:multipleDocumentSignature***-elementteihin.

2. ***hl7fi:multipleDocumentSignature***-elementti sisältää kutakin allekirjoitettavaa CDA R2 -asiakirjaa kohden ***hl7fi:Ref***-elementin seuraavasti:

- ***OID***-attribuutin arvona on CDA R2 -asiakirjan tunniste (***cda:ClinicalDocument/id***-elementin ***root*** ja ***extension***-attribuuttien mukainen arvo, jossa ***root***- ja ***extension***-arvot on erotettu pisteellä).

- ***hash***-attribuutin arvona on CDA R2 -asiakirjan allekirjoitettavasta sisällöstä muodostettu tiiviste. Tiiviste muodostetaan asiakirjan ***cda:structuredBody***-elementin sisällöstä käyttäen samoja menetelmiä ja parametreja kuin moniallekirjoitusrakenteeseen itseensä kohdistuvassa XML-allekirjoituksessa. ***cda:structuredBody***-elementin tiivisteen laskemisessa sovelletaan siis kaikki muut Transform-rakenteilla kuvatut muunnokset (ml. kanonikalisointi) paitsi kohdistamiseen liittyvät.

```
<hl7fi:multipleDocumentSignature ID="MDSid001">
   <hl7fi:Ref OID="1.2.246.10.98765432.93.2007.16" hash="bFEFUCL6NjvIw4tlwCTAvfYsWLM="/>
   <hl7fi:Ref OID="1.2.246.10.98765432.93.2007.2" hash="MZIz+sdPtKCORLFvyxf6IAInXt0="/>
   <hl7fi:Ref OID="1.2.246.10.98765432.93.2007.3" hash="B9/F5tBIs5o/xOGQmkQ4MjEXYxU="/>
</hl7fi:multipleDocumentSignature>
```

3. Lisätään muodostettu ***hl7fi:signature***-elementti kuhunkin allekirjoitettuun CDA-asiakirjaan

Moniallekirjoituksen tarkistaminen tapahtuu seuraavasti:

1. Tarkistetaan CDA-asiakirja joka sisältää allekirjoituksen XML-allekirjoitusstandardin toteuttavalla validaattorilla.
2. Tarkistetaan moniallekirjoitusrakenteen ja moniallekirjoitetun asiakirjan välinen liitos.

Moniallekirjoituksen muodostamisessa ja tarkistamisessa tarvittava ***cda:structuredBody***-elementin tiivisteen laskeminen edellyttää XML-allekirjoituksen mukaista toiminnallisuutta. Käytännön toteutuksissa voidaan hyödyntää XML-allekirjoitustoteutusta esimerkiksi siten, että asiakirja allekirjoitetaan palvelinvarmenteella mutta tätä allekirjoitusta ei tallenneta vaan pelkästään sen sisältämä tiiviste otetaan talteen moniallekirjoitusrakenteen muodostamista varten[7].

[7] Rakenteen muodostamisessa tehtävässä apuallekirjoituksessa käytettävällä varmenteella ei ole väliä koska itse allekirjoitusta ei tallenneta. Apuallekirjoitukseen käytettävälle varmenteelle ei ole mitään laatuvaatimuksia.

3 Sähköisen allekirjoituksen vaatimukset

3.1 Sähköisessä allekirjoituksessa sallitut menetelmät

Seuraavissa taulukossa on esitetty elementtikohtaisesti mitkä arvot ovat sallittuja parametreja CDA R2 -asiakirjan XML-allekirjoitusrakenteessa (***ds:Signature***). Elementit ovat kaikki ***ds:SignedInfo***-elementin lapsia. Vaihtoehtoisista arvoista suositeltu algoritmi on alleviivattu.
Toteutuksissa käytettävien arvojen PITÄÄ olla tämä tässä taulukossa sallituksi määritetty. Toteutusten PITÄISI käyttää alleviivattuja vaihtoehtoja aina kun mahdollista.

Taulukko 6

Elementti	Sallitut menetelmät
ds:CanonicalizationMethod	Exclusive XML Canonicalization version 1.0 (without comments) *[http://www.w3.org/2001/10/xml-exc-c14n#]* Canonical XML version 1.0 (without comments) *[http://www.w3.org/TR/2001/REC-xml-c14n-20010315]* Exclusive XML Canonicalization version 1.0 (with comments) *[http://www.w3.org/2001/10/xml-exc-c14n#WithComments]*
ds:SignatureMethod	RSAwithSHA256 *[http://www.w3.org/2001/04/xmldsig-more#rsa-sha256]* RSAwithSHA512 *[http://www.w3.org/2001/04/xmldsig-more#rsa-sha512]* ECDSAwithSHA256 *[http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256]* ECDSAwithSHA512 *[http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha512]*
ds:Reference/ ds:Transforms/ ds:Transform	Enveloped Signature Transform *[http://www.w3.org/2000/09/xmldsig#enveloped-signature]* XSLT Transform *[http://www.w3.org/TR/1999/REC-xslt-19991116]* XPath Filter-2 *[http://www.w3.org/TR/2002/REC-xmldsig-filter2-20021108/]* Exclusive XML Canonicalization version 1.0 (without comments) *[http://www.w3.org/2001/10/xml-exc-c14n#]* Canonical XML version 1.0 (without comments) *[http://www.w3.org/TR/2001/REC-xml-c14n-20010315]* Exclusive XML Canonicalization version 1.0 (with comments) *[http://www.w3.org/2001/10/xml-exc-c14n#WithComments]*
ds:Reference/ ds:DigestMethod	SHA256 *[http://www.w3.org/2001/04/xmlenc#sha256]* SHA512 *[http://www.w3.org/2001/04/xmlenc#sha512]*

3.2 Sähköisiä allekirjoituksia koskevat sitovat vaatimukset

Terveydenhuollon asiakirjoissa allekirjoituksen PITÄÄ olla osiossa ***cda:ClinicalDocument/hl7fi:localHeader/hl7fi:signatureCollection***

Sosiaalihuollon asiakirjoissa allekirjoituksen PITÄÄ olla osiossa ***cda:ClinicalDocument/hl7fi:localSocialHeader/hl7fi:signatureCollection***

Kaikissa terveydenhuollon ja sosiaalihuollon asiakirjoissa PITÄÄ käyttää aina kahta allekirjoituksen kohdistusta. Toinen näistä kohdistuksista PITÄÄ aina kohdistaa aikaleimarakenteeseen osiossa ***hl7fi:signatureTimestamp***

Sosiaalihuollon asiakirjoissa allekirjoitus PITÄÄ kohdistaa aina osioon ***cda:nonXMLBody***

Terveydenhuollon asiakirjoissa joiden varsinainen tietosisältö ei ole CDA-muotoinen allekirjoitus PITÄÄ kohdistaa aina osioon ***cda:nonXMLBody***

Moniallekirjoitusta SAA käyttää reseptiasiakirjojen allekirjoittamiseen.

Yksittäisen asiakirjan allekirjoituksessa EI SAA käyttää moniallekirjoitusrakennetta.

Moniallekirjoitusta EI SAA käyttää muiden asiakirjatyyppien kuin reseptiasiakirjojen allekirjoittamiseen.

Moniallekirjoituksessa allekirjoitus PITÄÄ kohdistaa osioon ***hl7fi:multipleDocumentSignature***

Terveydenhuollon asiakirjoissa joiden tietosisältö on CDA-muodossa ja joita ei moniallekirjoiteta allekirjoitus PITÄÄ kohdistaa osioon ***cda:structuredBody***

Allekirjoituksissa käytettävien varmenteiden PITÄÄ olla voimassaolevan lain ja asetusten mukaisia[8].

Kaikki järjestelmäallekirjoitukset PITÄÄ tehdä yksittäisen asiakirjan allekirjoituksina.

Allekirjoittajan allekirjoitusvarmenne PITÄÄ liittää osaksi allekirjoitusta sekä yksittäin allekirjoitetuissa että moniallekirjoitetuissa dokumenteissa. Käytettävä rakenne on ***ds:Signature/ds:KeyInfo/ds:X509Data/ds:X509Certificate***. Muita ***ds:KeyInfo***-rakenteita EI SAA käyttää.

Allekirjoitettavan CDA R2 -asiakirjan PITÄÄ olla kulloinkin voimassa olevan virallisen CDA R2 -skeeman mukainen sekä ennen allekirjoitusta, että allekirjoituksen jälkeen.

CDA R2 -asiakirjan sisältämän XML-allekirjoituksen PITÄÄ olla validi XML-allekirjoitusstandardin kokonaisuudessaan toteuttavaa allekirjoitusvalidaattoria vastaan, esimerkiksi Oraclen tai Apachen Santuario-implementaatio.

Käytettäessä Reference- kohdistusta, allekirjoituksen kohteena olevalle rakenteelle PITÄÄ antaa ***ID***-attribuutti[9] ja tälle arvo.

Aikaleimarakenteen allekirjoituksen kohdistamisessa PITÄÄ aina hyödyntää ***ID***-attribuutin arvoa avaimena.

[8] Terveydenhuollon ja sosiaalihuollon varmenteita toimittavan Digi- ja väestötietovirasto varmenteisiin liittyvät määrittelyt löytyvät osoitteesta *https://dvv.fi/palveluvarmenteet* ja *https://dvv.fi/varmenteet-sosiaali-ja-terveydenhuollolle*
[9] ***ID***-attribuutin kirjoitusasu on CDA-määrityksissä ***ID*** ja XML-allekirjoituksen määrityksissä ***Id***.

Kaikissa kohdistuksessa PITÄÄ käyttää Filter2-kohdistusta tai Reference-kohdistusta.

Käytettäessä Filter2-kohdistusta, PITÄÄ käytetyn suodatuksen olla suojattu ”XML Signature Wrapping”-hyökkäykseltä. Määrityksen lopussa on esimerkki sallitusta suodatustavasta.

Allekirjoitusrakenteissa käytettävät ajat PITÄÄ määrittää sekunnin tarkkuudella.

Allekirjoitusrakenteissa PITÄÄ käyttää ***hl7fi:signatureDescription*** -elementtiä ja tässä PITÄÄ käyttää oikeaa allekirjoitustyyppiä.

Kanta-allekirjoitus PITÄÄ tarkastaa Reseptikeskuksesta haettavissa asiakirjoissa. Tarkastuksessa on otettava huomioon, että allekirjoituksessa käytetty varmenne voi olla vanhentunut.

3.3 Sähköisiä allekirjoituksia koskevat suositukset

Kanonikalisoinnissa PITÄISI käyttää menetelmää ”Exclusive XML Canonicalization version 1.0 (without comments)”.

Tietojärjestelmää kehitettäessä PITÄÄ varmistaa käytettävän kohdistuksen oikeellisuus, eli varmistua että:

- allekirjoitus kohdistuu haluttuun rakenteeseen ja vain siihen
- allekirjoituksen kohdistus on sama myös sen jälkeen, jos asiakirjaan myöhemmin lisätään toinen allekirjoitus.

ds:Signature-elementille PITÄISI asettaa ***Id***-attribuutti ja tälle yksilöivä arvo, vaikka allekirjoitusta tuottava järjestelmä ei tätä arvoa itse käyttäisi mihinkään.

Tyhjätilamerkkien suodatusta PITÄISI käyttää (kuvattu luvussa 4.3.1).

Tietojärjestelmien PITÄÄ toteuttaa tuki sekä RSA- että ECC[10]-menetelmillä tuotettaville allekirjoituksille ja varautumaan siihen, että käytössä on molempia menetelmiä samaan aikaan.

3.4 Moniallekirjoituksessa käytettävät menetelmät

Moniallekirjoitusrakenteen sisältämien hajautussummien muodostamisessa PITÄÄ hyödyntää samoja menetelmiä kuin moniallekirjoitusrakenteeseen itseensä kohdistuvassa ***ds:Reference***-elementissä on käytetty. Moniallekirjoitusrakenteen allekirjoittamiseen käytettyjä kohdistamisen menetelmiä ei voi käyttää suoraan tiivisteen muodostamisessa, vaan kohdistus pitää tehdä kohteen mukaan.

Ne algoritmit, joita PITÄÄ tukea moniallekirjoituksia muodostettaessa ja tarkistettaessa on eritelty seuraavassa taulukossa:

[10] RSA:n nimi tulee algoritmin kehittäjien sukunimistä, ECC tulee englanninkielen sanoista Elliptic Curve Cryptography.

Taulukko 7

Elementti	Moniallekirjoitukseen periytyvät algoritmit
ds:Reference/ ds:Transforms/ ds:Transform	*http://www.w3.org/2000/09/xmldsig#enveloped-signature* *http://www.w3.org/TR/1999/REC-xslt-19991116* *http://www.w3.org/2001/10/xml-exc-c14n#* *http://www.w3.org/2001/10/xml-exc-c14n#WithComments* *http://www.w3.org/TR/2001/REC-xml-c14n-20010315*
ds:Reference/ ds:DigestMethod	*http://www.w3.org/2001/04/xmlenc#sha256* *http://www.w3.org/2001/04/xmlenc#sha512*

4 Taustaa (ei normatiivinen)

4.1 XML-allekirjoitus

XML-allekirjoitus on XML-muotoinen tietorakenne, jonka sisältämä sähköinen allekirjoitus kohdistuu XML-muotoiseen tietoon. XML-allekirjoitus on mahdollista liittää osaksi allekirjoitettua tietoa siten, että allekirjoituksen automaattinen tarkistaminen on mahdollista eri ympäristöissä.

XML-allekirjoitusstandardi määrittää joukon erilaisia menetelmiä joita voidaan käyttää allekirjoituksen muodostamisessa. Allekirjoituksen tarkistaminen edellyttää tukea samoille menetelmille joita on käytetty allekirjoituksen muodostamisessa.

XML-allekirjoitusstandardin mukaiseen sähköiseen allekirjoitukseen liittyviä parametreja ovat:

- kanonikalisointimenetelmä (canonicalization, c14n)
- allekirjoitusmenetelmä (signature)
- viittaus allekirjoitettavaan tietoon (reference URI)
- tiedon muutokset ja suodatus (transforms, filtering)
- tiivistefunktio (digest)

Kanonikalisoinnissa allekirjoitettava XML yhtenäistetään esitystavaltaan aina täsmälleen samaan muotoon. Allekirjoituksella osoitetaan tiedon muuttumattomuus ja liityntä allekirjoituksen muodostaneeseen tahoon. Viittauksella, muutoksilla ja suodatuksella osoitetaan allekirjoitettavasta asiakirjasta allekirjoitettavat kohdat ja voidaan muuntaa allekirjoitettavaa muotoa. Tiivistefunktiolla tarkoitetaan menetelmää, jolla allekirjoitettavasta kohdasta muodostetaan tiedon muuttumattomuuden osoittava tiiviste (hajautussumma).

XML allekirjoituksen rakenne on esitetty alla ("?" tarkoittaa nolla tai yksi, "+" tarkoittaa yksi tai useampi ja "*" nolla tai useampi):

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

4.2 XML-allekirjoituksen kohdistuminen allekirjoitettavaan sisältöön

XML-allekirjoitus muodostuu kahdesta päällekkäisestä kerroksesta. Sisempänä on ***ds:SignedInfo***-rakenne ja sen sisältämät ***ds:Reference***-solmut, jotka sisältävät viittauksen allekirjoitettavaan sisältöön. Ulompana on varsinaisen julkisen avaimen allekirjoituksen kerros.

Julkisen avaimen kerroksen allekirjoituksessa allekirjoitettava sisältö on ***ds:SignedInfo***-rakenne. Ennen allekirjoittamista ***ds:SignedInfo***-rakenne kanonikalisoidaan ***ds:CanonicalizationMethod***-solmun mukaisella menetelmällä. Allekirjoituksessa käytetty algoritmi määritetään ***ds:SignatureMethod***-solmussa. Allekirjoituksessa käytetyn avaimen tiedot esitetään ***ds:KeyInfo***-solmussa. Allekirjoituksen arvo tallennetaan ***ds:SignatureValue***-solmuun.

XML-allekirjoitus kohdistuu allekirjoitettavaan sisältöön ***ds:Reference***-rakenteella siten että kohteesta muodostettu tiiviste tallennetaan ***ds:DigestValue***-solmun arvoksi. Kohdistuminen tapahtuu määrittämällä kohteena olevan XML-rakenteen sijainti suhteessa allekirjoitukseen ja suodatukset jotka rakenteelle tehdään ennen tiivisteen laskemista.

Kohteen sijainti voidaan esittää ***ds:Reference***-elementin ***URI***-attribuutissa URI-viittauksella (Reference-kohdistus). Vaihtoehtoisesti ***URI***-attribuutti voi viitata XML-rakenteen juureen ja tarkka sijainti määritetään Filter-suodatuksella (Filter2-kohdistus). Tämän määrityksen esimerkeissä käytetään tilan säästämiseksi Reference-kohdistusta.

URI-attribuutin avulla viittaaminen tapahtuu XPointer-standardin[11] mukaisesti. XML-allekirjoitusstandardin mukaisissa ympäristöissä tuettuja XPointereita ovat ainakin dokumentin juureen viittaava tyhjä arvo (***URI***=””) ja dokumentin sisäinen viittaus elementin ***ID***-attribuuttiin (***URI***=”#attribuutinarvo”). XPointer kohdistuu tiettyyn elementtiin ja kaikkiin sen alasolmuihin.

Filter-suodatuksen avulla XPointerin tekemää kohdistusta on mahdollista rajata yksityiskohtaisesti. Yleisesti tuettuja Filter-suodatuksia on kaksi erilaista; XML Path Language Version 1.0 (XPath) ja XML-Signature XPath Filter 2.0 (Filter2). Suodatukset ovat kuvailuvoimaltaan vastaavia, mutta Filter2-toteutukset ovat useimmissa ympäristöissä tehokkaampia kuin XPath-toteutukset.

Tässä määrityksessä kuvataan sallituiksi kohdistamistavoiksi Reference-kohdistus ja Filter2-kohdistus.

Kohdistuksen kohteesta laskettavan tiivisteen muodostamisessa käytettävä algoritmi määritetään ***ds:DigestMethod***-solmussa.

Ennen tiivisteen laskemista kohteena oleva XML-rakenne suodatetaan ***ds:Transform***-solmujen mukaisilla menetelmillä. Suodatusmenetelmät ovat jaettavissa neljään osajoukkoon käyttötarkoituksen mukaisesti. Käyttötarkoitukset ja näitä vastaavat algoritmit on esitetty alla taulukossa:

[11] XML Pointer Language (XPointer) Version 1.0, W3C Candidate Recommendation 11 September 2001, *http://www.w3.org/TR/2001/CR-xptr-20010911/*

Taulukko 8

Käyttötarkoitus	Algoritmi
XML-allekirjoitusten suodattaminen	*http://www.w3.org/2000/09/xmldsig#enveloped-signature*
Kohdistaminen / kohteen rajaaminen	*http://www.w3.org/2002/06/xmldsig-filter2*
Suodattaminen XSLT-merkintäkielen avulla	*http://www.w3.org/TR/1999/REC-xslt-19991116*
Kanonikalisointi	*http://www.w3.org/2001/10/xml-exc-c14n#* *http://www.w3.org/2001/10/xml-exc-c14n#WithComments* *http://www.w3.org/TR/2001/REC-xml-c14n-20010315*

XML-allekirjoitusten suodattaminen -toiminnallisuudella XML-allekirjoitukset suodatetaan pois allekirjoituksen kohteena olevasta XML-rakenteesta. Tämänhetkisissä CDA-asiakirjojen allekirjoituksissa allekirjoitusten suodattaminen ei ole tarpeen, mutta tästä ei myöskään ole mitään haittaa.

Kohdistaminen / kohteen rajaaminen -toiminnallisuudella allekirjoituksen kohdistus XML-rakenteeseen voidaan rajata yksityiskohtaisesti suodatusmenetelmälle annettujen parametrien mukaisesti.

Suodattaminen XSLT-merkintäkielen avulla -toiminnallisuudella allekirjoituksen kohteena oleva XML-rakennetta voidaan suodattaa yksityiskohtaisesti menetelmälle annettujen parametrien mukaisesti.

Kanonikalisointi-toiminnallisuudella allekirjoituksen kohteena oleva XML-rakenne voidaan yhdenmukaistaa ennen allekirjoituksen muodostamista. Kanonikalisointi tulee tehdä suodatusmenetelmistä viimeisenä, jotta muut sen jälkeen sovellettavat menetelmät eivät sotke yhdenmukaistettua järjestystä.

4.3 XML-allekirjoituksen haasteet

XML-allekirjoitus on kahden eri maailman kohtaamispiste. XML ja sen päälle tehdyt määritykset, esimerkiksi CDA-dokumenttirakenne, ovat luonteeltaan semanttisia. Ne perustuvat merkityksiin ja niiden yksikäsitteiseen ilmaisemiseen. Sähköinen allekirjoitus taas perustuu bittijonoihin kohdistuviin algoritmisiin operaatioihin. Koska XML-standardit sallivat samojen merkitysten ilmaisemisen useilla eri tavoilla, syntyy tästä väistämättä ongelmia.

Näiden ongelmien ratkaisemiseksi on kehitetty XML-allekirjoitusstandardi, jota ylläpitää W3C (World Wide Web Consortium). Ongelman lähtökohtaisen hankaluuden ja kentällä olevien lukuisten toimijoiden takia kyseisestä standardista on muodostunut varsin mutkikas.

Keskeisimmät tulkintakohdat XML-allekirjoitustandardissa liittyvät suodatukseen ja kanonikalisointiin. Standardi tarjoaa lukuisia eri vaihtoehtoja päästä samaan lopputulokseen. Eri tilanteissa onkin usein perusteltua käyttää eri vaihtoehtoja. Tietyn kontekstin sisällä toimittaessa on perusteltua yhtenäistää käytäntöjä eri toimijoiden kesken.

Seuraavissa alaluvuissa käsitellään XML-allekirjoituksiin liittyviä ongelmakohtia ja niiden välttämiseen käytettävissä olevia keinoja.

4.3.1 Tyhjätilamerkit

Tyhjätilamerkkejä (white space) ovat välilyönnit, rivivaihdot ja sarkainmerkit.

Erilaiset XML-työkalut käsittelevät tyhjätilamerkkejä eri tavoin, minkä seurauksena allekirjoitusten eheys saattaa rikkoontua. Erityisesti rivinvaihdot ovat ongelmallisia eivätkä eri sovellukset käsittele niitä yhdenmukaisesti[12].

Tyhjätilamerkkien yhtenäistäminen on mahdollista toteuttaa XML-allekirjoituksen tukemien menetelmien avulla käyttämällä XSL-suodatusta joka poistaa allekirjoitettavasta asiakirjasta ylimääräiset tyhjät merkit ennen allekirjoituksen laskemista. Käytännössä tämä on mahdollista XSLT-kielen *normalize-space()*-funktion avulla.

normalize-space()-funktio korvaa kaikki yhden tai useamman tyhjätilamerkin ilmentymät yhdellä välilyönnillä. Erityisesti on huomionarvoista, että allekirjoitus ei tällöin ota huomioon rivinvaihtoja, joilla saattaa olla joissain erikoistapauksissa merkitystä tietosisällölle. Rivinvaihdot jäävät huomioimatta kuitenkin vastaavasti myös esitettäessä CDA-asiakirja HTML-muodossa.

Esimerkki XSLT-suodatuksesta joka yhtenäistää XML-sisällön tyhjätilamerkit:

```
<ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xslt-19991116">
  <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
    <xsl:template match="*|@*|comment()">
      <xsl:copy>
        <xsl:apply-templates select="*|@*|text()|comment()" />
      </xsl:copy>
    </xsl:template>
    <xsl:template match="text()">
      <xsl:value-of select="normalize-space(.)" />
    </xsl:template>
  </xsl:stylesheet>
</ds:Transform>
```

4.3.2 Kommentit

XML:n semanttisen luonteen takia asiakirjan kommentteihin ei pitäisi sisällyttää merkitsevää tietoa. Niiden sisällyttäminen allekirjoitukseen puolestaan saattaa aiheuttaa lisäongelmia tyhjien merkkien takia sekä siksi, että käytetyt työkalut saattavat ennalta-arvaamattomasti "kuoria" ne pois käsittelyketjuissa.

Kommentit eivät ole ongelma CDA-asiakirjojen allekirjoituksissa käytettäessä kanonikalisointi-algoritmeja, jotka suodattavat kommentit pois ennen allekirjoituksen muodostamista ja tarkistamista.

4.3.3 Nimiavaruudet

XML:n siirtäminen esimerkiksi SOAP-kääreessä ja muu käsitteleminen saattaa lisätä rakenteeseen ennalta-arvaamattomasti nimiavaruuksien lyhenteitä, kuten "hl7fi:" tai "cda:".

[12] Erilaisten rivinvaihtomerkkien historiaan voi tutustua esimerkiksi wikipedian artikkelista: *http://en.wikipedia.org/wiki/Newline*

CDA-asiakirjojen allekirjoituksissa hyväksyttäviksi on määritetty kaksi nimiavaruuksia eri tavalla käsittelevää kanonikalisointialgoritmia. Inclusive-kanonikalisointia käytettäessä mukaan otetaan kaikki allekirjoitettavassa asiakirjassa käytetyt nimiavaruudet, vaikka niitä ei käytettäisi itse allekirjoituksen kohteena olevassa XML:ssä. Exclusive-kanonikalisointia käytettäessä mukaan otetaan vain ne nimiavaruudet jotka ovat käytössä allekirjoituksen kohteena olevassa XML:ssä.

Inclusive-kanonikalisointia käytettäessä tulee varmistua, että asiakirjassa esiintyy vain tarvittavat nimiavaruudet ja puhdistaa asiakirja ylimääräisistä nimiavaruuksista tarvittaessa.

Nimiavaruuksien käyttöä XML-allekirjoituksessa käytettävissä XPatheissa tulee pyrkiä välttämään samoista syistä.

4.3.4 Merkistöt ja erikoismerkit

Jotta merkistömuunnoksissa tapahtuvat virheet havaitaan mahdollisimman aikaisessa vaiheessa, on suositeltavaa käyttää testiaineistoa joka sisältää erikoismerkkejä. UTF-8 merkistössä on syytä käyttää sekä kaksi- että useampitavuisia erikoismerkkejä. Esimerkiksi €-merkki on UTF-8:ssa kolmitavuinen.

Vastaavasti myös ääkkösiä sisältävien testivarmenteiden käyttö on suositeltavaa.

4.3.5 Reference-kohdistus

Suora Reference-kohdistus ***ID***-elementin arvoon edellyttää käytettävän XML-ympäristön tunnistavan kohteena olevan attribuutin ***xs:ID***-tyyppiseksi arvoksi. Käytännössä tämä edellyttää joko DTD- tai XML Schema -tiedoston liittymistä käsiteltävään XML-asiakirjaan siten että se on allekirjoituksen muodostamiseen ja tarkistamiseen käytettävän ympäristön hyödynnettävissä. Eri XML-ympäristöt eroavat standardin noudattamistavoiltaan eikä XPointerin yhdenmukainen toimiminen ole aina taattua.

Kaikissa eri XML-allekirjoitusalustoissa Reference-kohdistus ei ole enää oletuksena tuettu. Tämä johtuu nimellä ”XML signature wrapping” tunnetusta tietoturvahaavoittuvuudesta, ja tältä suojautumisesta.

Kanta-palvelussa on suojauduttu tältä haavoittuvuudelta ja Kanta-palvelun liittyvä järjestelmä voi jatkossakin muodostaa allekirjoituksia, joissa käytetään paikallisia viittauksia XPointerilla.

4.4 Tiivistefunktiot ja allekirjoitusmenetelmät

Sähköisissä allekirjoituksissa käytettävät menetelmät ovat elinkaareltaan pitkiä ja muutokset yleensä hitaita.

Tiivistefunktioissa on ollut käynnissä siirtyminen SHA1-menetelmästä SHA2-menetelmiin jo useita vuosia - tämä siirtymä on näkynyt myös Kanta-ympäristössä. Käytännössä SHA1-menetelmä on poistumassa täysin käytöstä.

Allekirjoitusmenetelmissä siirtyminen RSA-menetelmästä elliptisten käyrien (ECC) menetelmiin on vielä alkuvaiheessa[13]. ECC-menetelmä otetaan Kanta-ympäristössä käyttöön RSA:n rinnalle.

DVV on uudistanut varmennehierarkiaansa siten, että kaikki varmentajat uudistuivat vuosien 2021 ja 2022 aikana. Uusien varmenteiden sisältämät allekirjoitukset käyttävät SHA384 ja SHA512-tiivistefunktiota (varmenteilla/korteilla tehtävät allekirjoitukset voivat käyttää muitakin tiivistefunktioita).

Siirtyminen käyttämään SHA384 ja SHA512-tiivistefunktioihin ei edellytä muutoksia käytettäviin kortteihin tai käytettävään kortinlukijaohjelmistoon. Allekirjoituksen toteuttavaan sovellukseen SHA 384 ja SHA512-tiivistefunktioiden käyttöönotto vaatii muutoksia[14].

Terveydenhuollossa käytössä olevat DVV:n myöntämät sosiaali- ja terveydenhuollon varmennekortit ovat uudistuneet. Aikaisemmin toimikortit olivat aina RSA-pohjaisia, mutta jatkossa on saatavilla myös ECC-pohjaisia kortteja / varmenteita.

Siirtyminen käyttämään ECC-pohjaisia kortteja voi vaatia myös muita muutoksia toimintaympäristöön kuin pelkän kortin vaihtamisen. Käytetyn kortinlukijan ja kortinlukijaohjelmistoversion tulee tukea ECC:tä. Allekirjoituksen toteuttavan sovelluksen pitää myös tietää / tunnistaa ECC-pohjaisen varmenteen käyttö ja ottaa tämä huomioon allekirjoitusta muodostettaessa.

4.5 Tiivistefunktiot XML-allekirjoituksessa

Tiivistefunktiota käytetään kolmessa eri kohdassa XML-allekirjoitusta:

1. **Allekirjoituksen kohteesta lasketun tiivisteen muodostamisessa käytettävä tiivistefunktio**
   Käytettävä funktio määritetään elementissä
   ds:Signature/ds:SignedInfo/ds:Reference/ds:DigestMethod
   Käytettävä funktio on teknisesti allekirjoituksen muodostajan valittavissa allekirjoitushetkellä. Tämä määritys antaa käytettävän arvon.

2. **XML-allekirjoitusrakenteen allekirjoituksessa käytettävä tiivistefunktio**
   Käytettävä funktio määritetään elementissä
   ds:Signature/ds:SignedInfo/ds:SignatureMethod
   Käytettävä funktio on teknisesti allekirjoituksen muodostajan valittavissa allekirjoitushetkellä. Tämä määritys antaa käytettävän arvon.

3. **Allekirjoituksessa käytetyn varmenteen allekirjoitus**
   Käytetty menetelmä on allekirjoituksessa käytetyn varmenteen sisäisessä kentässä. Käytetty funktio on varmentajan asettama eikä sitä voida vaihtaa jälkikäteen.

Näistä kohdista 1 ja 2 ovat tämän määrityksen alaisia. Tässä määrityksessä ei oteta kantaa kohtaan 3, joka on varmentajan hallinnassa.

[13] Siirtyminen pois RSA:n käytöstä johtuu ensisijaisesti suorituskyvystä joka muodostuu haasteeksi avainpituuksien kasvaessa.
[14] Muutoksien laajuus riippuu sovelluksesta – tarvittava muutos on pienimmillään pelkkä konfiguraatio.

5 Allekirjoituksen prosessit (ei normatiivinen)

5.1 Henkilökohtaisen yksittäisen allekirjoituksen muodostaminen

Henkilökohtaisesti allekirjoitettavan yksittäisen allekirjoituksen muodostamisen prosessi on seuraava:

1. **Käyttäjä** valitsee asiakirjan allekirjoituksen suoritettavaksi (voi tapahtua myös implisiittisesti)
2. **Sovellus** muodostaa asiakirjan tiedoista CDA R2 -asiakirjan
3. **Sovellus** muodostaa aikaleimarakenteen ja liittää tämän asiakirjaan
4. **Sovellus** päättelee tehdäänkö allekirjoitus ECC- vai RSA-avaimilla[15]
5. **Sovellus** muodostaa XML-allekirjoituksen, joka kohdistuu aikaleimarakenteeseen ja asiakirjan tietosisältöön. Rakenne sisältää tiedon käytettävästä allekirjoitusmenetelmästä.
   - **Sovellus** välittää allekirjoitettavasta sisällöstä muodostetun tiivisteen käyttäjän toimikortille allekirjoitettavaksi
   - **Käyttäjä** syöttää PIN2-koodin ja kortti tekee allekirjoituksen.
6. **Sovellus** muodostaa ja liittää XML-allekirjoituksen asiakirjaan

5.2 Järjestelmäallekirjoitetun yksittäisen allekirjoituksen muodostaminen

Järjestelmäallekirjoitettavan asiakirjan allekirjoituksen muodostamisen prosessi on seuraava:

1. **Sovellus** käynnistää allekirjoituksen (voi tapahtua käyttäjän toiminnan seurauksena tai taustaprosessina, tms.)
2. **Sovellus** muodostaa asiakirjan tiedoista CDA R2 -asiakirjan
3. **Sovellus** muodostaa aikaleimarakenteen ja liittää tämän asiakirjaan
4. **Sovellus** tietää konfiguraatioista käytetäänkö ECC- vai RSA-avaimia
5. **Sovellus** muodostaa XML-allekirjoituksen, joka kohdistuu aikaleimarakenteeseen ja asiakirjan tietosisältöön. Rakenne sisältää tiedon käytettävästä allekirjoitusmenetelmästä.
   - **Sovellus** käyttää allekirjoittamiseen sen käytettäväksi konfiguroitua järjestelmäallekirjoitusvarmennetta.
6. **Sovellus** muodostaa ja liittää XML-allekirjoituksen asiakirjaan

5.3 Yksittäisen allekirjoituksen tarkistaminen

Yksittäisen allekirjoituksen tarkistamisen prosessi on seuraava:

1. **Sovellus** tarkistaa XML-allekirjoituksen eheyden. XML-allekirjoituksen tarkistaminen tarkistaa allekirjoituksen kohteena olevien tietojen sisällön muuttumattomuuden.
2. **Sovellus** tarkistaa XML-allekirjoituksen sisältämän varmenteen eheyden ja luotettavuuden (varmenteen tulee olla luotetun ja hyväksytyn varmentajan

[15] Tämä päättely voi tapahtua esimerkiksi tekemällä varmenteen valinta -operaatio ja lukemalla varmenteen tiedoista onko kyseessä ECC- vai RSA-pohjainen kortti. Toinen mahdollinen tapa on hyödyntää käyttäjän tekemää korttikirjautumista ja päätellä käytettävä menetelmä tästä.

myöntämä). Jos sovellus samalla tarkastaa varmenteen voimassaolon, tulee sen varautua siihen, että varmenne ei ole enää voimassa.
3. **Sovellus** tarkistaa allekirjoituksen muodostamisajan ja vertaa tätä varmenteen voimassaoloaikaan. Aikaleima ei saa olla nykyhetkestä katsottuna tulevaisuudessa eikä varmenteen voimassaoloajan ulkopuolella (ennen varmenteen voimassaolon alkamista tai voimassaolon päättymisen jälkeen tehty)

Lähetettäessä asiakirja Kanta-järjestelmään, liittyy allekirjoituksen muodostamiseen vielä seuraavat vaiheet:

4. **Sovellus** lähettää allekirjoitetun asiakirjan tietovarantoon
5. **Kanta** tarkistaa asiakirjassa olevan allekirjoituksen oikeellisuuden
6. **Kanta** allekirjoittaa asiakirjan Kanta-järjestelmäallekirjoituksella (jos kyseessä on Resepti-palvelu)
7. **Kanta** tallentaa asiakirjan

Allekirjoituksesta voidaan haluttaessa tarkistaa myös seuraavia osioita:

1. **Sovellus** tarkistaa että allekirjoitus kohdistuu määritysten mukaisesti aikaleimaan ja ***cda:structuredBody***-osioon.

2. **Sovellus** tarkistaa, että allekirjoituksessa käytetyt menetelmät ovat tämän määrityksen mukaisia.

Kanta suorittaa aina myös viimeksi mainitut tarkastukset.

5.4 Moniallekirjoituksen muodostaminen

Moniallekirjoituksen muodostamisen prosessi on seuraava:

1. **Käyttäjä** valitsee tai merkitsee allekirjoitettavat asiakirjat käyttämänsä sovelluksen käyttöliittymästä
2. **Käyttäjä** valitsee moniallekirjoituksen suoritettavaksi (voi tapahtua myös implisiittisesti)
3. **Sovellus** muodostaa moniallekirjoitusrakenteen
4. **Sovellus** muodostaa kutakin asiakirjaa vastaavan rivin moniallekirjoitusrakenteeseen
   - Asiakirjan tietosisällöstä lasketaan tiiviste. Tiivisteen laskemisessa käytettävät menetelmät ovat samat kuin moniallekirjoitusrakenteeseen itseensä kohdistuvassa ***ds:Reference***-elementissä (6)
   - Poikkeuksena kohdistamisessa käytettävät menetelmät (URI, filter2). Näiden osalta ei käytetä ***ds:Reference***-elementin arvoja.
   - Menetelmien soveltamisjärjestys on sama kuin ***ds:Reference***-elementissä
   - Muodostettu tiiviste tallennetaan Base64-muodossa hash-attribuutin arvoksi.
   - Asiakirjan tunniste (OID) ja tiiviste liitetään yhteen ***hl7fi:Ref***-elementin arvoiksi.
5. **Sovellus** muodostaa yhden aikaleimarakenteen
6. **Sovellus** päättelee tehdäänkö allekirjoitus ECC- vai RSA-avaimilla

7. **Sovellus** muodostaa XML-allekirjoituksen, joka kohdistuu aikaleimarakenteeseen ja moniallekirjoitusrakenteeseen. Rakenne sisältää tiedon käytettävästä allekirjoitusmenetelmästä.
   - **Sovellus** välittää allekirjoitettavasta sisällöstä muodostetun tiivisteen käyttäjän toimikortille allekirjoitettavaksi
   - **Käyttäjä** syöttää PIN2-koodin ja kortti tekee allekirjoituksen.
8. **Sovellus** muodostaa yhden allekirjoitusrakenteen joka sisältää XML-allekirjoituksen, moniallekirjoitusrakenteen ja aikaleiman, sekä kopioi tämän saman rakenteen jokaiseen moniallekirjoituksen kohteena olleeseen asiakirjaan

Lähetettäessä moniallekirjoitettu asiakirja Kanta-järjestelmään, liittyy allekirjoituksen muodostamiseen vielä seuraavat vaiheet:

9. **Sovellus** lähettää allekirjoitetun asiakirjan tietovarantoon
10. **Kanta** tarkistaa asiakirjassa olevan moniallekirjoituksen oikeellisuuden
11. **Kanta** allekirjoittaa asiakirjan Kanta-järjestelmäallekirjoituksella
12. **Kanta** tallentaa asiakirjan

5.5 Moniallekirjoituksen tarkistaminen

Moniallekirjoituksen tarkistamisen prosessi on seuraava:

1. **Sovellus** tarkistaa asiakirjan sisältämien XML-allekirjoitusten eheyden. XML-allekirjoituksen tarkistaminen tarkistaa allekirjoituksen kohteena olevien tietojen sisällön muuttumattomuuden.
2. **Sovellus** tarkistaa XML-allekirjoituksen sisältämien varmenteiden eheyden ja luotettavuuden (varmenteen tulee olla luotetun ja hyväksytyn varmentajan myöntämä)
3. **Sovellus** tarkistaa kunkin allekirjoituksen muodostamisajan ja vertaa tätä kyseisen allekirjoituksen varmenteen voimassaoloaikaan. Aikaleima ei saa olla nykyhetkestä katsottuna tulevaisuudessa eikä varmenteen voimassaoloajan ulkopuolella (ennen varmenteen voimassaolon alkamista tai voimassaolon päättymisen jälkeen tehty)
   - **Sovellus** muodostaa asiakirjan tietosisällöstä tiivisteen ja vertaa tätä asiakirjan moniallekirjoitusrakenteessa vastaavan rivin ***hash***-attribuutin arvoon.
   - **Sovellus** valitsee tarkastettavan rivin siten että OID-attribuutti vastaa tarkistettavan CDA R2 asiakirjan tunnistetta (***cda:ClinicalDocument/cda:id***-elementin ***root***- ja ***extension***-attribuuttien mukainen arvo)
   - Tiivisteen laskemisessa käytetään soveltuvin osin samoja menetelmiä samassa järjestyksessä kuin tarkistettavaan moniallekirjoitusrakenteeseen kohdistuvassa ***ds:Reference***-elementissä käytetään.
     - Poikkeuksena kohdistamisessa käytettävät menetelmät (URI, filter2). Näiden osalta ei käytetä ***ds:Reference***-elementin arvoja.
     - Menetelmien soveltamisjärjestys on sama kuin ***ds:Reference***-elementissä
     - Muodostettua tiivistettä verrataan Base64-muodossa ***hash***-attribuutin arvoon.

Tiivisteen muodostamisessa suositellaan käytettävän hyödyksi XML-allekirjoitustoteutusta siten, että asiakirjasta muodostetaan järjestelmäallekirjoitus käyttäen asiakirjan tietosisältöön kohdistuvassa ***ds:Reference***-elementissä samoja menetelmiä ja näiden parametreja kuin tarkistettavassa allekirjoituksessa käytetään moniallekirjoitusrakenteeseen kohdistuvassa allekirjoituksessa. Poikkeuksena tähän kuitenkin kohdistaminen, jotta kohteena on asiakirjan tietosisältö eikä moniallekirjoitusrakenne.

Allekirjoituksesta voidaan haluttaessa tarkistaa myös seuraavia osioita:

4. **Sovellus** tarkistaa että allekirjoitus kohdistuu määritysten mukaisesti aikaleimaan ja moniallekirjoitusrakenteeseen.
5. **Sovellus** tarkistaa että allekirjoituksessa käytetyt menetelmät ovat tämän määrityksen mukaisia.

Sovelluksen ei ole välttämätöntä tarkistaa moniallekirjoitusta. Riittää, että sovellus tarkistaa Kanta-järjestelmän tekemän järjestelmäallekirjoituksen. Tämä tarkoittaa implisiittisesti sitä, että sovellus ja käyttäjä luottavat Kanta-järjestelmän tarkistaneen moniallekirjoituksen oikein (luku 5.4, kohdat 9-11).

6 Esimerkit (ei normatiivinen)

CDA-asiakirja voidaan allekirjoittaa esimerkiksi alla kuvatuilla tavoilla. Nämä esimerkit eivät ole sitovia eivätkä ainoa toimiva tapa allekirjoittaa CDA-asiakirja. Esimerkkien tarkoitus on täydentää määritystä.

6.1 Henkilön allekirjoittamayksittäinen potilasasiakirja

Tiedostossa *EsimerkkiAllekirjoitus2018_1_yksiresepti.xml* on esimerkki Filter2-menetelmän avulla kohdistetusta CDA-allekirjoitusrakenteesta käytettäessä SHA256-tiivistefunktiota ja RSAwithSHA256-allekirjoitusalgoritmia.

Filter2-kohdentamisessa on käytetty seuraavia XPath-parametreja

```
//*[local-name()='ClinicalDocument']/*[local-name()='localHeader']/*
[local-name()= 'signatureCollection']/*[local-name()= 'signature']/*
[local-name()='signatureTimestamp'][@ID='esimerkkiAika1']

//*[local-name()='ClinicalDocument']/*[local-name()='component']/*[local-name()='structuredBody']
```

Kaikki kolme kanonikalisointia (***SignedInfo***-, ***signatureTimestamp***- ja ***StructuredBody***-rakenteet) ovat esimerkissä Inclusive-kanonikalisointimenetelmän (Canonical XML version 1.0 (without comments)) mukaisia. Esimerkissä käytetään tyhjän tilan siistimiseen XSLT-transformaatiota.

6.2 Järjestelmäallekirjoitettu sosiaalihuollon asiakirja

Tiedostossa *EsimerkkiAllekirjoitus2018_2_yksiPDF.xml* on esimerkki Filter2-menetelmän avulla kohdistetusta CDA-allekirjoitusrakenteesta kun sisältö on PDF-muotoinen ja käytetään SHA512-tiivistefunktiota ja ECCwithSHA256-allekirjoitusalgoritmia.

Filter2-kohdentamisessa on käytetty seuraavia XPath-parametreja

```
//*[local-name()='ClinicalDocument']/*[local-name()='localHeader']/*
[local-name()= 'signatureCollection']/*[local-name()= 'signature']/*
[local-name()='signatureTimestamp'][@ID='esimerkkiAika2']

//*[local-name()='ClinicalDocument']/*[local-name()='component']/*[local-name()='nonXMLBody']
```

Kaikki kolme kanonikalisointia (***SignedInfo***-, ***signatureTimestamp***- ja ***nonXMLBody***-rakenteet) ovat esimerkissä Inclusive-kanonikalisointimenetelmän (Canonical XML version 1.0 (without comments)) mukaisia. Esimerkissä käytetään tyhjän tilan siistimiseen XSLT-transformaatiota.

6.3 Moniallekirjoitettu lääkemääräys

Tiedostoissa *EsimerkkiAllekirjoitus2018_3_moniallekirjoitus1.xml* ja *EsimerkkiAllekirjoitus2018_4_moniallekirjoitus2.xml* on esimerkki käytettäessä Reference- kohdistusta kun kohteena on moniallekirjotusrakenne ja käytetään SHA256-tiivistefunktiota ja ECCwithSHA256-allekirjoitusalgoritmia.

Reference-kohdistuksessa on käytetty seuraavia XPointer-arvoja

#esimerkkiMoniallekirjoitusRakenne1 #esimerkkiAika3

Kaikki kolme kanonikalisointia (***SignedInfo***-, ***signatureTimestamp***- ja ***nonXMLBody***-rakenteet) ovat esimerkissä Inclusive-kanonikalisointimenetelmän (Canonical XML version 1.0 (without comments)) mukaisia. Esimerkissä käytetään tyhjän tilan siistimiseen XSLT-transformaatiota.