< IH Elektronische Handtekening met UZI-pas

Datum: 16 december 2016
Publicatie: AORTA 2015 (V6.14.0.0)

Inhoudsopgave

1 Inleiding

1.1 Doel en scope

Dit document heeft tot doel het gebruik van de elektronische handtekening in de zorgsector nader te specificeren. Dit document beschrijft de samenstelling van de te ondertekenen gegevens, wat nader door een zorgtoepassing in te vullen is en de randvoorwaarden die aan het ondertekenen worden gesteld.

1.2 Doelgroep voor dit document

Dit document is vooral bedoeld voor softwareontwikkelaars van zorgapplicaties en zorg-infrastructurele applicaties, die op grond van de HL7v3 communicatiestandaard en op grond van dit document berichten willen ondertekenen.

1.3 Documenthistorie

Versie	Datum	Omschrijving
v6.10.0.0	12-okt-2011	RfC 46142: SOAP Headers van tokens worden uitgebreid met SOAP:actor. RfC 46183: soap:actor moet zijn gbx ipv gbz. Verwijzing naar [IH Transport] opgenomen en tekst aangepast.
v6.11.0.0	12-okt-2012	Ongewijzigde herpublicatie als onderdeel van AORTA-Infrastructuur v6·11
V6.12.15.0	14-dec-2015	Ongewijzigd overgenomen in documentset 6.12.15.0
V6.14.0.0	16-dec-2016	Ongewijzigd overgenomen in documentset 6.14.0.0

2 Het handtekeningtoken

Bij het ondertekenen van gegevens met een elektronische handtekening ligt de inhoud van die gegevens niet op generieke wijze vast, maar wordt de inhoud van die gegevens bepaald per zorgtoepassing. In deze handleiding worden daarom alleen vormvereisten gesteld aan de ondertekende gegevens. Iedere zorgtoepassing die binnen AORTA gebruik wil maken van een elektronische handtekening, moet aan deze vormvereisten voldoen. Omdat de inhoud van de ondertekende gegevens per zorgtoepassing bepaald wordt, wordt in dit hoofdstuk met een voorbeeld gewerkt. We gebruiken hiervoor een fictief voorbeeld: zorginstellingen verstrekken maaltijden aan opgenomen patiënten, en het maaltijdvoorschrift wordt door een zorgverlener ondertekend.

Stel dat een zorgtoepassing een elektronisch maaltijdenvoorschrift wil hanteren met de volgende inhoud:

- voorschrijvend arts: naam, UZI-nummer;
- moment van ondertekenen;
- identificatie maaltijd;
- voorgeschreven maaltijd;
- instructie gebruik;
- patiëntgegevens: naam, BSN, geboortedatum, geslacht.

2.1 Structuur

De te ondertekenen gegevens worden dit in een signedDataMeal blok geplaatst, bijvoorbeeld als hieronder.

```
<signedDataMeal xmlns="http://www.aortarelease.nl/805/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd"
  wsu:Id="id_2.16.840.1.113883.2.4.99.1.2.3_123456">
  <signatureMetaData>
    <signatureVersion>http://www.aortarelease.nl/805/meal/1</signatureVersion>
    <ds:X509IssuerSerial>
      <ds:X509IssuerName>CN=TEST UZI-register...C=NL</ds:X509IssuerName>
      <ds:X509SerialNumber>2754...999</ds:X509SerialNumber>
    </ds:X509IssuerSerial>
  </signatureMetaData>
  <meal>
    <id>
      <root>2.16.840.1.113883.2.4.99.3.4.5</root>
      <extension>0123456789</extension>
    </id>
    <dateTime>20090319144010</dateTime>
    <patient>
      <name>J.M. Breed</name>
      <gender>M</gender>
      <birthdate>19680816</birthdate>
      <id>
        <root>2.16.840.1.113883.2.4.6.3</root>
        <extension>012345672</extension>
      </id>
    </patient>
    <author>
      <name>Hendrikus Rudolf Testzorgverlener30</name>
      <id>
        <root>2.16.840.1.113883.2.4.6.3</root>
        <extension>000005489</extension>
      </id>
          </author>
    <dinner>
      <code>
                <codeSystem>2.16.840.1.113883.2.4.99.2.3.4</codeSystem>
        <code>999999</code>
      </code>
      <text>Fettucine met verse wintertruffel, parelhoen gevuld
        met appel en walnoot, salade van wintergroenten</text>
    </dinner>
    <usage>Avondeten, innemen met een glas goede wijn</usage>
  </meal>
</signedDataMeal>
```

2.2 Namespaces

De elektronische handtekening maakt gebruik van de volgende namespaces. De prefixen zijn niet normatief maar worden in dit document als voorbeelden gebruikt.

Tabel AORTA.STK.t3700 – Namespaces

Prefix	Namespace URI
ao	http://www.aortarelease.nl/805/
ds	http://www.w3.org/2000/09/xmldsig#
soap	http://schemas.xmlsoap.org/soap/envelope/
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd

2.3 Inhoud

2.3.1 signedData<naam> element

```
<signedDataMeal xmlns="http://www.aortarelease.nl/805/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd"
  wsu:Id="id_2.16.840.1.113883.2.4.99.1.2.3_123456">
```

Er moet een naam worden toevoegd aan de "signedData" tagnaam. Een los <signedData> element wordt al gebruikt voor berichtauthenticatie met de UZI pas, een andere naam maakt het makkelijker tokens met XPath te vinden.

Het wsu:Id dient om de relatie te leggen met de feitelijke handtekening (XML Signature in de WS Security Header). Het wsu:Id moet uniek zijn zodat bij samenvoegen van ondertekende gegevens uit diverse bronnen in een enkel XML bestand, de Id's uniek blijven. Bij elektronische handtekeningen is dat een zeer realistisch scenario. In bijvoorbeeld een query zouden meerdere recepten met handtekeningen opgehaald kunnen worden. De relatie tussen ieder recept en de bijbehorende handtekening moet dan uniek zijn; vandaar de eis dat wsu:Id globaal uniek moet zijn.

Er moet een unieke Id opgenomen worden. De toegestane methoden om het Id samen te stellen zijn:

- begin met "id",
- voeg hier een underscore "_" aan toe,
- voeg een hiervoor gekozen OID toe die uniek is voor het verzendende systeem,
- voeg hier een underscore "_" aan toe,
- voeg een (binnen de scope van de OID) uniek nummer voor het uitgegeven handtekeningtoken toe,
- bijvoorbeeld: id_2.16.840.1.113883.2.4.99.1.2.3_123456

of:

- begin met "uuid",
- voeg hier een underscore "_" aan toe,

- voeg een nieuw gegenereerde UUID toe,
- bijvoorbeeld: uuid_8e45bb15-aa1a-4649-a22f-28eefb70b1ed.

2.3.2 Metadata

```
<signatureMetaData>
  <signatureVersion>http://www.aortarelease.nl/805/meal/1</signatureVersion>
  <ds:X509IssuerSerial>
    <ds:X509IssuerName>CN=TEST UZI-register...C=NL</ds:X509IssuerName>
    <ds:X509SerialNumber>2754...999</ds:X509SerialNumber>
  </ds:X509IssuerSerial>
</signatureMetaData>
```

Onder signedData... komt eerst een blok metagegevens. Een verwijzing naar het gebruikte certificaat wordt meegetekend, zodat deze relatie ook hard wordt vastgelegd. Een versienummer voorkomt dat een aanvaller een ontdekt lek kan uitbuiten door te suggereren dat een oudere versie gebruikt is. Deze gegevens moeten opgenomen worden, maar hoeven niet aan de zorgverlener getoond te worden.

Deze gegevens zijn:

De versie van de specificatie. Een zorgtoepassing bepaalt welke waarde hier in moet komen, en welke versie gebruikt mag worden. Deze waarde wijzigt niet met nieuwe versies van het gerelateerde bericht, tenzij de samenstelling of werkwijze van het handtekeningtoken wijzigt. Een ontvanger **moet** een onbekende waarde afkeuren. In de context van de elektronische handtekening en de bijbehorende wettelijke lading is het nooit voldoende onbekende inhoud stilzwijgend te negeren.

De issuer en serialnumber van het gebruikte certificaat, volgens [XMLSIG].

2.3.3 Naam en inhoud elementen

```
<meal>
```

Het volgende element direct onder het signedData element moet een (herkenbare) naam van het *type bericht* hebben. In dit geval wordt *meal* gebruikt. Dit element mag alleen XML elementen bevatten (geen mixed content).

Voor de XML elementen onder het signedData element gelden de volgende regels:

- het element mag alleen tekstuele inhoud óf alleen andere elementen bevatten,
- áls het element alleen andere elementen bevat, dient het alleen ter groepering,
- áls het element tekstuele inhoud bevat, dan is die inhoud een deel van de specifieke inhoud van dit specifieke bericht; een deel van de mededeling van de ene zorgverlener aan de andere, deze inhoud is zoveel mogelijk zoals de ontvangende zorgverlener deze te zien krijgt,
- de naam van het element is vrij te kiezen per zorgtoepassingen en kan bijvoorbeeld in lijn met de HL7v3 payload worden gebracht.

2.3.4 Identificatie en uniekheid

```
<id>
  <root>2.16.840.1.113883.2.4.99.3.4.5</root>
  <extension>0123456789</extension>
</id>
```

Ieder handtekeningtoken moet een unieke identifier bevatten; bijvoorbeeld een receptnummer, dossiernummer of iets dergelijks.

```
<meal>
  <id extension="0123456789" root="2.16.840.1.113883.2.4.99.3.4.5"/>
```

Wanneer dit binnen een bepaalde zorgtoepassing niet mogelijk is, moet een UUID gebruikt worden.

2.3.5 Datum en tijd

```
<dateTime>20090319144010</dateTime>
```

Er moet een datum opgenomen worden; dit is de datum van ondertekenen, gelijk aan deze datum op een papieren ondertekend stuk. Ook (juist) wanneer een bericht een dergelijke datum niet kent, wordt deze opgenomen. In dit geval is dit een lokale tijd. Waar een tijd gebruikt wordt die ook in het bericht voorkomt, wordt het formaat in het bericht gevolgd dus inclusief tijd en/of tijdzone wanneer dat zo in het bericht zit, en zonder tijd en/of tijdzone wanneer dat ook niet in het bericht zit.

Het weten van de datum waarop iets ondertekend is, is juridisch van belang, en technisch wanneer het bijvoorbeeld gaat om de geldigheid van de gebruikte certificaten. Wanneer er geen datum in het bericht staat, moet de datumtijd een precisie in seconden hebben. Wanneer er wel een datum in het bericht staat, moet die minimaal de dag aangeven. Het formaat is als de datums in het HL7v3 bericht (datatype TS).

2.3.6 Patiëntgegevens

```
<patient>
  <id>
    <root>2.16.840.1.113883.2.4.6.3</root>
    <extension>012345672</extension>
  </id>
  <name>J.M. Breed</name>
  <gender>M</gender>
  <birthdate>19680816</birthdate>
</patient>
```

Voor alle aan één enkele patiënt gerelateerde berichten zijn naam, geslacht, geboortedatum en BSN verplicht. Extra benodigde velden kunnen per zorgtoepassing bepaald worden. Het BSN (in element patient/id/extension) moet overeenkomen met het BSN in het bericht zelf:

```
<Patient>
  <id extension="012345672" root="2.16.840.1.113883.2.4.6.3"/>
```

Voor toepassingen (b.v. intramuraal) waar geen BSN bekend hoeft te zijn voor de patiënt (bijvoorbeeld pasgeborenen), mag één en niet meer dan één ander patiënt id gebruikt

worden. Binnen AORTA is gebruik van BSN verplicht, dus dit alternatief geldt alleen voor toepassingen buiten AORTA die voor gebruik van handtekeningen toch van deze standaard gebruik willen maken.

2.3.7 Auteurgegevens

```
<author>
  <name>Hendrikus Rudolf Testzorgverlener30</name>
  <id>
    <root>2.16.528.1.1007.3.1</root>
    <extension>000005489</extension>
  </id>
  </author>
```

Idem voor de zorgverlenernaam en identificatie. De naam is zoals de zorgverlener die op b.v. een papieren voorschrift hanteert. Ook hier moet het UZI-nummer overeenkomen met het UZI-nummer in het bericht:

```
<AssignedPerson>
  <id extension="000005489" root="2.16.528.1.1007.3.1"/>
```

Waar de rolcode relevant is, is het deze verplicht op te nemen volgens de rolcode tabel van het CIBG. Het author blok ziet er dan als volgt uit:

```
<author>
  <name>Hendrikus Rudolf Testzorgverlener30</name>
  <id>
    <root>2.16.528.1.1007.3.1</root>
    <extension>000005489</extension>
  </id>
    <role>
    <codeSystem>2.16.840.1.113883.2.4.15.111</codeSystem>
    <code>01.000</code>
    <name>Arts</name>
  </role>
</author>
```

2.3.8 Representatie van gegevens

Gegevens worden zoveel mogelijk verzonden worden zoals de zorgverlener deze ziet.

- Namen worden als een enkele string getoond: "Jhr. Mr. M. van der Goes van Naters", niet opgesplitst in titels, tussenvoegsels en dergelijke.
- Een eenvoudige representatie van de naam heeft de voorkeur, bijvoorbeeld: voornamen voorvoegsels geslachtsnaam.
- Veel voorkomende en bekende codes, zoals M/V of M/F voor geslacht mogen zonder vertaling in "Mannelijk" en "Vrouwelijk" opgenomen worden. Dit maakt het eenvoudiger de code met het bericht te matchen.
- Datums worden opgenomen volgens het HL7v3 formaat *eejjmmddhhmmss* (de precisie wordt bepaald per zorgtoepassing).
- Het BSN wordt opgenomen, inclusief het OID wat het BSN uniek identificeert.

2.3.9 Codes

Codes die algemeen gangbaar zijn worden opgenomen (d.w.z. coderingssystemen die gebruikt worden in alle bij AORTA aangesloten systemen), met de bijbehorende tekst.

Hieronder wordt bijvoorbeeld een maaltijdcode uit een gangbaar maaltijdcoderingssysteem opgenomen:

```
<dinner>
  <code>
    <codeSystem>2.16.840.1.113883.2.4.99.2.3.4</codeSystem>
    <code>999999</code>
  </code>
  <text>Fettucine met verse wintertruffel, parelhoen gevuld met appel en walnoot,
   salade van wintergroenten</text>
</dinner>
```

Wanneer OID's nodig zijn in het bericht, mogen deze opgenomen worden. In het voorbeeld is de OID codeSystem nodig om de juiste codetabel te identificeren. Zowel de code als het codeSystem zullen de meeste zorgverleners op zich natuurlijk niets zeggen. De zorgverlener moet er in deze gevallen op vertrouwen dat de codes horen bij de getoonde tekst. De zorgtoepassing moet weer borgen dat de codes in het token vergeleken worden met het bericht en de zorgverlener moet de mogelijkheid hebben de getekende tekst in te zien.

Wanneer er geen gangbare code is, wordt een tekst meegetekend.

```
<usage>Avondeten, innemen met een glas goede wijn</usage>
```

In deze gevallen is het een vereiste dat deze tekst octet-voor-octet gelijk is aan een tekst die in het bericht opgenomen wordt:

```
<mealAdministrationRequest>
  <id extension="0030002011" root="2.16.840.1.113883.2.4.6.1.6005465.12.1.1"/>
  <text mediaType="text/plain">Avondeten, innemen met een glas goede wijn</text>
```

Omdat de getekende tekst en de tekst in het HL7v3-bericht in hetzelfde XML bericht zitten, zal de encoding gelijk zijn (UTF-8). De zorgtoepassing moet hier borgen dat de getekende tekst gematcht wordt met het bericht. Mogelijk wordt in het bericht ook een code opgenomen (bij doseringen zoals hierboven is dat het geval). De ontvangende zorgverlener zal in diens applicatie dan waarschijnlijk een vertaling van die codes zien. Het is dan belangrijk dat deze zorgverlener de mogelijkheid heeft de getekende tekst in te zien zoals deze getekend is. Bij twijfel kan op deze manier het "oorspronkelijke receptbriefje" (de getekende gegevens, niet het HL7v3-bericht) geraadpleegd worden.

```
</meal>
</signedData>
```

Afsluitende elementen.

2.4 Uit te werken per zorgtoepassing

Een zorgtoepassing moet op een aantal punten een nadere invulling geven aan de elektronische handtekening. Hier volgt een overzicht van zaken die niet generiek ingevuld kunnen worden, maar die per zorgtoepassing die gebruik wil maken van de elektronische handtekening vastgelegd moeten worden.

Een zorgtoepassing moet vastleggen of een elektronische handtekening getekend moet zijn met een certificaat dat geldig is op het moment van ontvangst. Certificaten die niet geldig zijn, zijn alleen toegestaan bij historische gegevens, nooit bij actuele gegevens. Wanneer een zorgtoepassing een elektronische handtekening inzet bij het verzenden van actuele gegevens, moet de ontvangende partij dus controleren of het certificaat geldig is bij ontvangst van de handtekening en ondertekende gegevens.

Een zorgtoepassing moet bepalen wat er moet gebeuren als de handtekening niet valide is (ook als het certificaat niet (meer) geldig is, of als de match met het bericht onjuist is. Er zijn diverse mogelijkheden:

- een foutmelding retourneren aan zender, en het daarbij laten;
- een foutmelding retourneren aan zender, en het bericht tonen aan de zorgverlener voor wie het bestemd is, met de mededeling dat de handtekening niet geldig is;
- geen foutmelding retourneren aan zender, en het bericht tonen aan de zorgverlener voor wie het bestemd is, met de mededeling dat de handtekening niet geldig is.

Bij medisch belangrijke berichten is het vaak raadzaam de zorgverlener voor wie het bericht bestemd is wel op de hoogte te stellen van de inhoud van het bericht, met de mededeling dat de handtekening niet geldig is. De zorgverlener is meestal beter in staat de juiste actie te bepalen dan een softwaresysteem.

Een zorgtoepassing moet vastleggen of de elektronische handtekening verplicht of optioneel is. In het eerste geval zal een bericht zonder elektronische handtekening tot een foutmelding moeten leiden.

Een zorgtoepassing moet vastleggen waar alleen codes, alleen teksten of zowel code als tekst in het handtekeningtoken opgenomen worden. Waar een code gebruikt wordt, moet de bijbehorende tekst ook opgenomen worden (met als enige uitzondering codes die algemeen bekend kunnen worden verondersteld zoals "M" of "V" voor geslacht). In deze gevallen moet een match tussen code in token en bericht geëist worden door de zorgtoepassing, en mag een dergelijke match tussen de bijbehorende tekst en de tekst in het bericht gevraagd worden door de zorgtoepassing. Het is het algemeen beter alleen een match tussen codes te eisen, en niet een match tussen de bijbehorende teksten. Het matchen van tekstuele gegevens is altijd bijzonder foutgevoelig. Bij teksten kunnen makkelijk verschillen optreden, bijvoorbeeld in "whitespace" die de inhoud niet aantasten, maar wel zorgen dat de teksten verschillen. Codes kennen dit soort problemen niet.

Elektronische gegevens, waaronder elektronische handtekening, zijn zonder verlies van kwaliteit te kopiëren. Een zorgtoepassing moet bepalen of hier maatregelen nodig zijn, en zo ja, welke.

Daarnaast moet een zorgtoepassing leveren:

- een beschrijving van het gebruikte handtekeningtoken;
- een XML Schema voor het gebruikte handtekeningtoken;
- een Schematron bestand voor de matching tussen handtekeningtoken en bericht;
- een XSLT stylesheet voor presentatie als HTML van het handtekeningtoken.

2.5 Algoritmes

Voor het berekenen van hashwaarden voor de elektronische handtekening is SHA-256 verplicht. Voor de elektronische handtekening wordt RSA gebruikt.

In de XML Signature wordt tweemaal gerefereerd aan deze waarden. Eenmaal voor de digest over de getekende data; en eenmaal voor de signature, waarbij een digestmethode plus een encryptie-algoritme. Dit leidt tot de volgende waarden in de inhoud van de XML signature:

Tabel AORTA.STK.t3710 – Algoritmes

Functie	Algoritme	URI
Signature	RSA+SHA-256	`<SignatureMethod Algorithm= "http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>`
Digest	SHA-256	`<DigestMethod Algorithm= "http://www.w3.org/2001/04/xmlenc#sha256"/>`

2.6 Opbouw

De ondertekende gegevens bij de elektronische handtekening worden in een soap Header ingebed. De bijbehorende XML Signature (de feitelijke handtekening) komt om een WS Security 1.0 Soap Header.

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <ao:authenticationTokens

            ...>
    ...
    </ao:authenticationTokens>
    <wss:Security
      soap:mustUnderstand="1"
      soap:actor="http://www.aortarelease.nl/actor/zim"
      ...
    </wss:Security>
    <ao:signatureTokens
      xmlns:ao="http://www.aortarelease.nl/805/"
      soap:mustUnderstand="1"
      soap:actor="http://www.aortarelease.nl/actor/gbx">
      <signedDataMeal xmlns="http://www.aortarelease.nl/805/"
      ...
      </signedDataMeal>
    </ao:signatureTokens>
    <wss:Security
      soap:mustUnderstand="1"
      soap:actor="http://www.aortarelease.nl/actor/gbx"
      ...
    </wss:Security>
  </soap:Header>
  <soap:Body>
  ...
  </soap:Body>
</soap:Envelope>
```

De getekende gegevens worden ingebed in een ao:signatureTokens element, wat de mogelijkheid biedt meerdere handtekeningtokens op te nemen onder dit element.

De volgorde van de soap Headers is bij voorkeur:

- eerst - indien aanwezig - de authenticatietokens;
- vervolgens - indien aanwezig - de overige headers bestemd voor de ZIM;
- het ao:signatureTokens element (met een of meer signedData... elementen hieronder);
- het wss:Security behorend bij de handtekening.

2.6.1 Bestemming van de soap Headers

SOAP headers kunnen verschillende bestemmingen hebben. Het attribuut soap:actor geeft aan welke "actor" een bepaalde header moet verwerken. Bij tokenauthenticatie en de elektronische handtekening speelt dit: tokenauthenticatie gebruikt headers gericht aan de ZIM, de elektronische handtekening headers gericht aan een GBZ of wellicht ooit aan de ZIM. Er moet dus gedifferentieerd worden op actor.

De elektronische handtekening is gericht aan een GBZ. Zie [IH Transport] voor de te gebruiken waarden in soap:actor voor headers gericht aan een GBZ.

Er is één WS-Security header per bestemming. Er zijn er dus maximaal twee: een voor de ZIM en een voor het GBZ. De header gericht aan de ZIM moet bij voorkeur vóór de header gericht aan een GBZ staan; daarmee kunnen ze verwerkt worden in de volgorde waarin ze nodig zijn.

Er moet een `soap:mustUnderstand="1"` vlag opgenomen worden in de header, die aangeeft dat de ontvanger deze header moet verwerken

3 Certificaten

De UZI-pas wordt gebruikt voor de elektronische handtekening. De pas die gebruikt wordt voor het ondertekenen moet een zorgverlenerpas of een medewerkerpas op naam zijn. Hoewel het pastype gecodeerd is opgenomen in het certificaat (in het subjectAltName attribuut), dient een applicatie op basis van de uitgevende CA vast te stellen wat het pastype van de UZI-pas is.

Bij de elektronische handtekening dient getekend te worden met het handtekeningcertificaat (NonRepudiation) (zie ook [UZI]). De ontvanger dient te controleren of het certificaat met de juiste keyUsage gebruikt is. De ontvanger dient te controleren of het certificaat is uitgegeven door het UZI-register (bijvoorbeeld in het geval van zorgverlenerspassen, wat bij de elektronische handtekening meestal het geval zal zijn, of het certificaat getekend is door "UZI-register Zorgverlener CA G2" of de "UZI-register Zorgverlener CA G21").

Om de handtekening te verifiëren, moet de ontvanger over de bijbehorende publieke sleutel beschikken. De ondertekenaar kan deze op verschillende manieren aan de ontvanger ter beschikking stellen:

1) door het certificaat met de publieke sleutel mee te zenden;
2) door een verwijzing naar het certificaat mee te zenden; de ontvanger moet deze dan met bijvoorbeeld het LDAP protocol ophalen in de directory van het UZI-register.

Voor de elektronische handtekening is gekozen voor het meezenden van certificaten in een wss:BinarySecurityToken. De reden is eenvoudig: voor de elektronische handtekening is de bestemming een zorgverlener, die de certificaten vaak niet zal hebben. Meezenden is dan eenvoudiger dan het certificaat ophalen bij het UZI-register. Een tweede overweging is het gemak bij archivering.

Noot: uiteraard mogen in het testtraject alleen UZI-testpassen gebruikt worden. Het gebruik hiervan wordt verder niet uitgewerkt in deze handleiding. De opbouw en werking is identiek.

4 Token afhandeling

4.1 Verwerking van de handtekening

4.1.1 Verificatie van handtekening en handtekeningtoken

Bij ontvangst moet de ontvangende applicatie de elektronische handtekening controleren. Een specificatie van een zorgtoepassing moet aangeven of een handtekeningencertificaat op het moment van ontvangst geldig moet zijn. Normaliter moet dit; de enige uitzondering zijn historische handtekeningen, waarbij niet gegarandeerd kan worden dat het certificaat niet verlopen of ingetrokken is op het moment van opvragen. Bij berichten waarbij ondertekenen deel is van het lopende zorgproces, moet een certificaat echter nog steeds geldig zijn bij ontvangst. Een zorgverlener die gegevens wil tekenen dient er dan ook voor te zorgen dat een gebruikte pas nog enige tijd na verzenden geldig is; m.a.w. men dient een verlopende UZI-pas tijdig te vervangen.

4.1.2 Controles bij ontvangst

Bij ontvangst dient een ontvanger de volgende controles uit te voeren (hieronder dient "deze standaard" gelezen te worden als "deze standaard incluis de standaarden waaraan in deze standaard gerefereerd wordt", zoals bijvoorbeeld [IH tokens generiek], [XMLSIG], [WSS] en [WSX509]):

- voldoet de WS-Security BinarySecurityToken header aan de eisen die er in deze standaard aan gesteld worden;
- voldoet de WS-Security Signature header aan de eisen die er in deze standaard aan gesteld worden;
- voldoet het handtekeningtoken aan de eisen die er in deze standaard aan gesteld worden;
  - klopt de versie van het handtekeningtoken in de metadata;
  - ligt de datum niet in de toekomst;
  - valideert de signature over het handtekeningtoken conform de eisen die er in deze standaard aan gesteld worden;
  - is het gebruikte certificaat correct getekend door het UZI-register;
  - is het gebruikte certificaat niet verlopen of ingetrokken (alleen wanneer een zorgtoepassing dit in bepaalde situaties expliciet toestaat, mag deze controle achterwege gelaten worden);
  - stemmen de X.509 IssuerName en SerialNumber in het handtekeningtoken overeen met die van het gebruikte certificaat;
  - stemt het UZI-nummer in het handtekeningtoken overeen met het nummer in het certificaat;
- is er een match tussen het bericht en het handtekeningtoken conform de eisen die de zorgtoepassing daaraan stelt, er moet daarbij minimaal een match zijn tussen:
  - het gebruikte Id in het handtekeningtoken en in het bericht;
  - BSN, indien gebruikt en relevant in de interactie;
  - UZI-nummer in het token en het relevante (door de zorgtoepassing is te bepalen welk) UZI-nummer in het bericht;
  - codes, waar deze gebruikt worden;
  - teksten, in gevallen waar geen code gebruikt wordt.

Deze controles dienen gedaan te zijn voordat een bevestiging op HL7v3 niveau geretourneerd wordt. Interacties die als respons een HL7v3 Accept Acknowledgement hebben, dienen dus voor retourneren van een HL7v3 Accept Acknowledgement met als waarde "CA" (Commit Ack) deze controles gedaan te hebben. Interacties die geen HL7v3

Accept Acknowledgement kennen, maar een HL7v3 Application Acknowledgement, moeten deze controles doen voordat een HL7v3 Application Acknowledgement met waarde "AA" (Application Acknowledgement Accept) geretourneerd wordt. Interacties die een inhoudelijk antwoord kennen (zoals queries), moeten deze controles doen voordat een dergelijk antwoord gegeven wordt.

4.1.3 Tonen van de ondertekende gegevens

Bij digitaal ondertekenen van gegevens is het belangrijk dat voor de ondertekenaar en de ontvanger duidelijk is wat er ondertekend wordt c.q. is. Een verschil met een handtekening op papier is dat de ondertekenaar/ontvanger niet letterlijk "ziet" wat er ondertekend wordt: op technisch niveau is datgene wat invoer is voor de tekenfunctie een reeks octets: voor de zorgverlener enen en nullen waar geen betekenis aan ontleend kan worden. Dus of deze gegevens een beeld in JPEG formaat vastleggen, of een film in MPEG, geluid in MP3 of leesbare tekens in ASCII of UTF-8, in alle gevallen is software nodig die ervoor zorgt dat de zorgverlener datgene "ziet" wat ondertekend wordt. Omdat deze software bij ondertekenaar en ontvanger anders kan zijn, is het belangrijk vast te leggen wat getoond moet worden: dus hoe de enen en nullen naar iets vertaald worden wat de zorgverlener kan zien. Een deel is vastgelegd in onderliggende standaarden als XML, UTF-8, en netwerk- en schijfopslagprotocollen. Deze worden hier niet nader uitgelegd. In deze gids wordt wel het hoogste niveau beschreven van deze "vertaling".

De tekenende en ontvangende zorgverleners moeten beiden in staat zijn om de ondertekende gegevens precies zo te zien als de ander. Dit "precies" betekent in deze context:

- ieder karakter moet getoond worden zoals het in het signedData blok staat,
- de gebruikte lettertypes mogen variëren per implementatie, zolang het gebruikte lettertype een getrouwe weergave is van het karakter horende bij de onderliggende UTF-8 octet(s),
- de volgorde van de karakters is ongewijzigd.

Een voorbeeld van de getoonde gegevens:

Maaltijd 0123456789

Maaltijdnummer:	0123456789
Tijdstip maaltijdvoorschrift:	19-03-2009, 14:40
Patient:	J.M. Breed
Geslacht:	M
Geboortedatum:	16-08-1968
BSN:	012345672
Voorschrift van:	Hendrikus Rudolf Testzorgverlener30
UZI-nummer:	000005489
Gerecht:	Fettucine met verse wintertruffel, parelhoen gevuld met appel en walnoot, salade van wintergroenten
Gebruik:	Avondeten, innemen met een glas goede wijn

Voor het tonen van de mededeling moet een zorgtoepassing een XSLT stylesheet leveren die het handtekeningtoken vertaalt naar HTML. Deze transformatie dient eenvoudige HTML op te leveren, die in diverse gangbare browsers vergelijkbare resultaten oplevert. Omdat deze transformaties gebaseerd zijn op wijdverbreide en stabiele standaarden (XML, XSLT, HTML) is aan te nemen dat beide zorgverleners de informatie op gelijke wijze tot zich kunnen nemen.

Bij de transformatie gelden de volgende regels:

- de geleverde stylesheet mag gebruikt worden (de transformatie mag ook met andere middelen plaatsvinden),
- een implementatie moet de tekenende én de ontvangende zorgverlener de mogelijkheid bieden de ondertekende informatie in te zien,
- een implementatie moet bij inzage de ondertekende informatie zodanig tonen dat een objectieve derde zal zeggen dat de getoonde informatie gelijk is aan die getoond met de stylesheet.

Deze versoepeling staat dus wel toe dat er geen XSLT-processor gebruikt wordt wanneer deze op het betreffende platform niet aanwezig is, mits de zorgverlener de informatie op soortgelijke wijze in kan zien.

Een implementatie mag dus wel enkel de informatie uit de payload van het bijbehorende HL7v3 bericht tonen aan de ontvanger, maar moet de zorgverlener in staat stellen de ondertekende gegevens (het handtekeningtoken) direct in te zien, zowel aan tekenende als ontvangende zijde. Het is essentieel dat datgene wat ondertekend is, ook ingezien kan worden door diegene die de ondertekende informatie ontvangt. Wanneer een implementatie enkel gegevens uit de payload van het bijbehorende HL7v3 bericht zou tonen, of codes zou vertalen met lokale vertaaltabellen, is er geen garantie meer dat

tekenende en ontvangende zorgverlener naar dezelfde informatie kijken, en wordt de essentie van de elektronische handtekening tenietgedaan.

4.2 Foutafhandeling

Foutafhandeling kan op de volgende manieren plaatsvinden:

1. door middel van een HTTP fout, bijvoorbeeld status 403 Forbidden,
2. door middel van een SOAP Fault die een probleem rond het handtekeningtoken of het tekenen aangeeft.

Met een HTTP status 4xx – 5xx moet overigens altijd gerekend worden; deze kan om meerdere redenen optreden.

4.2.1 HTTP fouten en SOAP Faults

De volgende SOAP Faults worden onderkend (zie ook de tabellen in [IH tokens generiek]).

Tabel AORTA.STK.t3760 – AORTA SOAP Faults

Omschrijving (faultstring)	Faultcode
Handtekeningtoken en bericht stemmen niet overeen	ao:SigTokenMessageMismatch
Handtekeningtoken is niet valide of compleet	ao:SigTokenInvalid

Bij fouten in de elektronische handtekening moet de waarde van faultactor gevuld worden met de geadresseerde, bijvoorbeeld: http://www.aortarelease.nl/actor/gbx (in [IH Transport] wordt aangegeven welke soap:actor attributen gebruikt moeten worden per bestemming). Het element detail in een SOAP Fault wordt bij voorkeur gevuld met aanvullende informatie, bijvoorbeeld een XPath expressie die de lokatie van de fout aangeeft of een nadere tekstuele toelichting.

Om te voorkomen dat deze informatie misbruikt wordt om de beste aanval te bepalen, dient eerst de handtekening geverifieerd te worden, en pas daarna op de "AORTA" fouten gecontroleerd te worden.

Een applicatie dient er ook rekening mee te houden dat er fouten door een ontvanger op HTTP-niveau worden afgehandeld, met name door het retourneren van de HTTP statuscode 403 "Forbidden" of 404 "Not Found".

Bijlage A Referenties

Referentie	Document	Versie
[IH tokens generiek]	Implementatiehandleiding security tokens generiek	6.14.0.0
[IH Transport]	Implementatiehandleiding berichttransport	6.14.0.0
[HL7v3 IH Wrp]	HL7v3 implementatiehandleiding berichtwrappers	6.14.0.0
[XML]	Extensible Markup Language (XML) 1.0 (Fifth Edition)	
[Namespaces]	Namespaces in XML 1.0 (Third Edition)	
[UZI]	http://www.uziregister.nl/veelgesteldevragen/certificaten/hoegebruikikvandedriecertificatendejuisteomteauthenticeren.asp	
[XMLSIG]	XML-Signature Syntax and Processing (Second Edition), W3C Recommendation, 12 February 2002 http://www.w3.org/TR/xmldsig-core/	
[WSS]	Web Services Security: SOAP Message Security 1.0, OASIS Standard Specification, March 2004 http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf	
[WSX509]	Web Services Security X.509 Certificate Token Profile 1.0, OASIS Standard Specification, March 2004 http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf	