Specifikation av säker elektronisk kommunikation mellan aktörer i försäkringsbranschen

(Version 1.1)

Version	Datum	Beskrivning
V.1.0	2002-09-10	Originalversion
V.1.1	2003-04-28	- Omstrukturering av specifikationen - Definierade säkerhetsnivåer - Registrerat namespace, ssek.org, för scheman - Anpassning av SSEK mot WS-Security - Definierat kvitto - Definierad TxHeader - Stöd för versionshantering av SSEK genom namespace på TxHeader - Definierad felhantering genom soap fault

1 Inledning

1.1 Bakgrund

SSEK (Specifikation av säker elektronisk kommunikation mellan aktörer i försäkringsbranschen) uppfyller det affärsbehov som finns av en standard för säker elektronisk kommunikation i försäkringsbranschen. Genom att följa specifikationen minskar risken för att organisationer måste bygga flera olika kommunikationslösningar eller att osäkra kommunikationslösningar byggs.

1.2 Syfte

Dokumentet är en specifikation av hur elektronisk kommunikation skall utföras säkert över Internet. Dokumentet kan med fördel användas som underlag och ingå i avtal om hur kommunikationen skall utföras mellan den egna organisationen och motparten.

1.3 Notation

Nyckelorden BÖR, KAN, SKALL och SKALL INTE har i detta dokument följande betydelse:

Nyckelord	Betydelse
BÖR	Det texten syftar till rekommenderas men är inte ett krav för att uppfylla specifikationen.
KAN	Det texten syftar till kan användas om affären kräver det men är inte ett krav för att uppfylla specifikationen.
SKALL	Det texten syftar till är ett krav för att uppfylla specifikationen.
SKALL INTE	Det texten syftar till är otillåtet enligt specifikationen.

1.4 Förvaltning av specifikationen

SSEK förvaltas inom SFM (Svenska Försäkringsmäklares Förening). Gruppen som hanterar utgivandet av specifikationen har representanter från organisationerna Skandia Liv, SEB Trygg-Liv, Länsförsäkringar, Alecta, Aspispronia, Danica, Folksam, SPP samt SFM.

1.5 Versioner och bakåtkompatibilitet

Specifikationen är inte statisk utan kan komma att förändras på grund av anpassning till nya eller förändrade standarder, t.ex. avseende signaturhantering. Implementationer av specifikationen måste kunna hantera olika versioner av specifikationen, genom att programmatiskt avgöra version av SSEK för ett specifikt inkommet meddelande. Detta hanteras genom att TxHeader får en ny namespace för varje version av SSEK, där förändringarna är av sådan karaktär att de påverkar implementationerna. Detta gäller oavsett om själva innehållet eller strukturen i TxHeader har förändrats eller inte.

För version 1.1 av SSEK gäller följande namespace för TxHeader:
http://schemas.ssek.org/txheader/2003-04-03/

För en eventuell version 1.2 av SSEK kan namespace för TxHeader bli:
http://schemas.ssek.org/txheader/2004-01-12/

1.6 Anslutna organisationer

De organisationer som anslutit sig till specifikationen har valt att kommunicera elektroniskt enligt SSEK och har en plattform för SSEK färdig, alternativt är på väg att ta fram en plattform. Alltså, kommunikation med anslutna organisationer kan utföras enligt SSEK eller ska kunna utföras efter en, av båda parterna, affärsmässigt godtagbar tid.

Naturligtvis uppmuntrar vi (gruppen för SSEK inom SFM) att även organisationer som inte anslutit sig till specifikationen utnyttjar SSEK för säker elektronisk kommunikation.

Följande organisationer har anslutit sig till SSEK.
- Skandia Liv
- SEB Trygg-Liv
- Länsförsäkringar

1.7 Standarder

Detta dokument, SSEK, beskriver hur den elektroniska kommunikationen mellan parter i försäkringsbranschen skall se ut. SSEK bygger på ett antal befintliga standarder, specifikationer och rekommendationer från erkända standardiseringsorgan som exempelvis IETF (Internet Engineering Task Force), W3C (World Wide Web Consortium) och OASIS (Organization for the Advancement of Structured Information Standards).

För information om vilka standarder SSEK bygger på, se *Bilaga A Standarder* och *Bilaga B Web Services Security*.

De standarder och specifikationer som uppges i *Bilaga A Standarder* omfattas fullt ut av SSEK med undantag av WS-Security (Web Services-Security) och därigenom även XML-Signature, där valda delar används enligt *Bilaga B Web Services Security*.
WS-Security bygger, liksom SSEK, på en samling standarder och beskriver hur elektronisk kommunikation utförs säkert. SSEK är dock direkt anpassad efter behoven för elektronisk kommunikation i försäkringsbranschen. Därför används endast de delar av WS-Security-specifikationen som det finns affärsbehov av.

En stor fördel med att stödja delar av WS-Security är att de produkter som tas fram med stöd för WS-Security kan utnyttjas. Alltså, SSEK-meddelanden skall kunna skapas och hanteras med hjälp av de ramverk som följer specifikationen WS-Security.

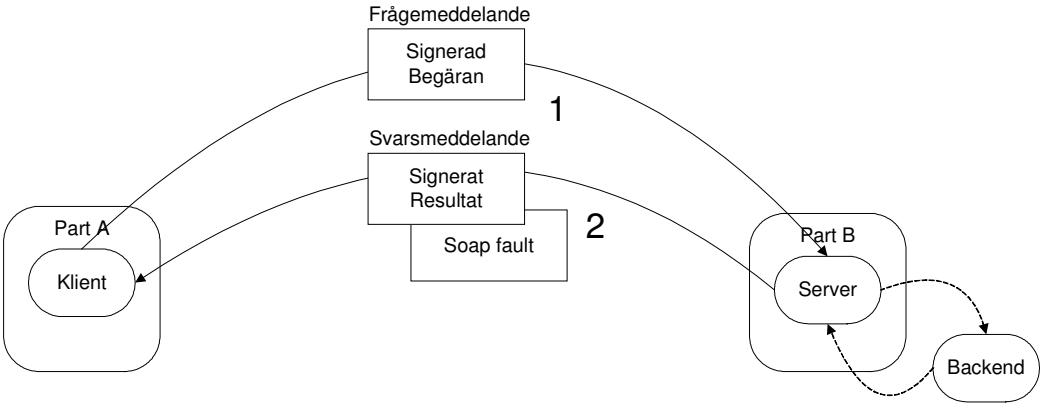
2 Kommunikation

Vid elektronisk kommunikation SKALL flödet vara specificerat mellan de kommunicerande parterna. Denna del beskriver kommunikationsflödet för synkron och asynkron kommunikation.

All kommunikation enligt SSEK sker synkront, med ett **frågemeddelande** och ett **svarsmeddelande** i form av SOAP-meddelanden. Ett asynkront flöde kan åstadkommas genom att kombinera två synkrona sändningar. För att kunna särskilja de synkrona sändningarna i ett asynkront flöde kallar vi dessa fortsättningsvis för **begäran** och **resultat**. För att knyta ihop ett asynkront flöde, d.v.s. koppla ihop begäran med resultatet används TxId enligt avsnitt *3.4 Tekniskt meddelandehuvud*.

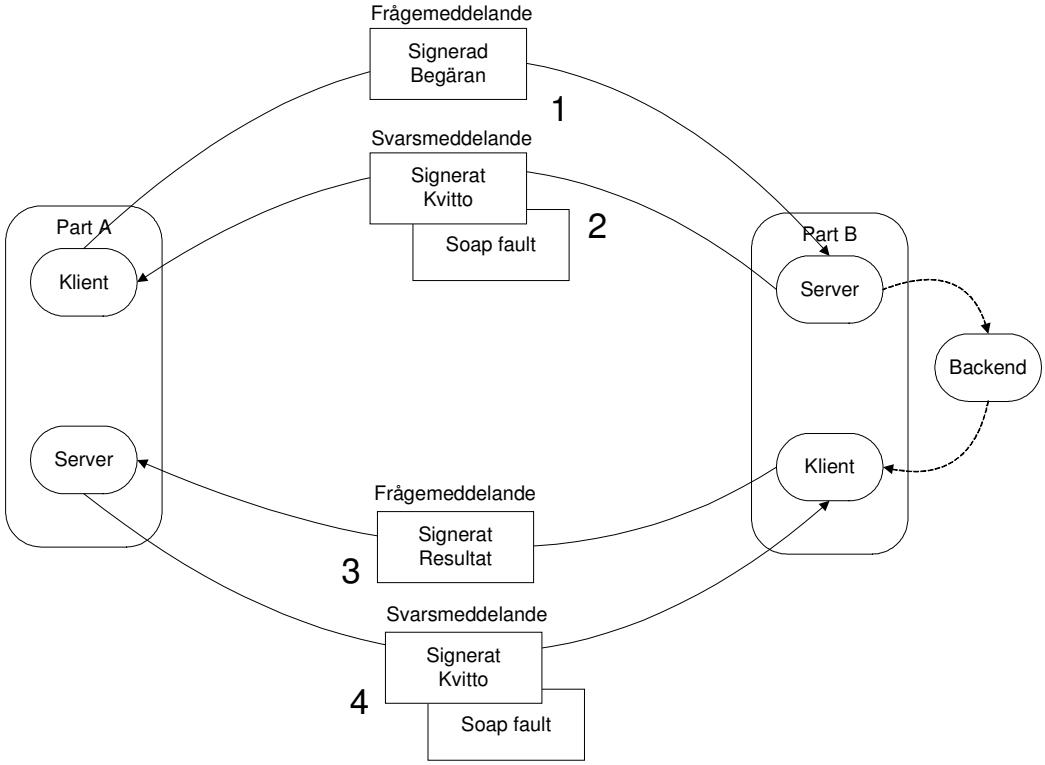
2.1 Kommunikationsflöde

Följande bild och beskrivning visar hur flödet SKALL se ut vid **synkron** kommunikation när signatur används. Signaturen är inte nödvändig utan används beroende på affärskrav.



1. Part A skickar signerat frågemeddelande för en begäran till Part B.
2. Part B verifierar signaturen samt formatet i frågemeddelandet, bearbetar meddelandet och returnerar ett signerat resultat som beskriver hur bearbetningen gått. Vid fel som upptäcks vid mottagandet returneras ett soap fault-meddelande enligt avsnitt *3.6 Soap fault*.

Följande bild och beskrivning visar hur flödet SKALL se ut vid **asynkron** kommunikation när signatur används. Signaturen är inte nödvändig utan används beroende på affärskrav.



1. Part A skickar signerat frågemeddelande med unikt TxId för en begäran till Part B.
2. Part B returnerar ett signerat kvitto efter att ha verifierat signaturen samt formatet i frågemeddelandet. Part A har då erhållit ett bevis på att Part B mottagit meddelandet. Observera att Part B **inte** har givit några garantier på att meddelandet i frågemeddelandet kan bearbetas. Vid fel som upptäcks vid mottagandet returneras ett soap fault-meddelande enligt avsnitt *3.6 Soap fault*.
3. Efter att ha bearbetat begäran skickar Part B ett signerat frågemeddelande med ursprungligt TxId och ett resultat och som beskriver hur bearbetningen gått till Part A.
4. Part A kopplar resultatet till begäran med hjälp av TxId (se avsnitt *3.4 Tekniskt meddelandehuvud)* och returnerar ett signerat kvitto efter att ha verifierat signaturen samt formatet i det emottagna resultatet från Part B. Vid fel som upptäcks vid mottagandet returneras ett soap fault-meddelande enligt avsnitt *3.6 Soap fault*.

2.2 Fel i kommunikationsflödet

I kommunikationen kan situationer uppstå där osäkerhet råder om ett frågemeddelande tagits emot eller bearbetats. Detta kan t.ex. uppstå om den säkra kanalen (SSL-tunneln) går ner efter det att mottagaren tagit emot ett frågemeddelande, men innan avsändaren tagit emot ett svarsmeddelande. Avsändaren av frågemeddelandet kan i detta läge inte veta om mottagaren tagit emot frågemeddelandet eller inte.

Om en sådan situation uppstår SKALL avsändaren av frågemeddelandet ta kontakt med mottagaren för att ta reda på om meddelandet bearbetats eller inte. Avsändaren SKALL INTE skicka meddelandet igen utan att först ha tagit kontakt med mottagaren.

Mottagaren av ett frågemeddelande kan, trots att inget kvitto tagits emot av avsändaren, bearbeta meddelandet och skicka ett resultat och få tillbaka ett kvitto på resultatet. Är det då en tjänst som definierats med att arkivering ska ske har i detta läge avsändaren av begäran en icke komplett transaktion i sitt arkiv. Det saknas ett kvitto på begäran. Mottagaren av frågemeddelandet SKALL på begäran av motparten kunna skicka ett sådant kvitto så att båda parter kan arkivera den kompletta transaktionen.

Då avsändaren kontaktat mottagaren och det visar sig att frågemeddelandet för begäran inte bearbetats eller tagits emot av motparten SKALL avsändaren generera ett nytt unikt TxId och skicka in meddelandet igen.

2.3 Hantering av TxId

Baserat på föregående avsnitt SKALL nedanstående gälla för avsändaren respektive mottagaren för ett frågemeddelande.

2.3.1 Avsändarens ansvar

Avsändaren av ett frågemeddelande för en begäran ansvarar för att generera ett unikt TxId. Vid osäkerhet kring om ett meddelande tagits emot av mottparten SKALL avsändaren kontakta motparten för att utreda status på transaktionen.

2.3.2 Mottagarens ansvar

Mottagaren av ett meddelande ansvarar för att kontrollera om inkommet TxId är unikt. Vid en dubblett SKALL mottagaren returnera ett soap fault-meddelande med faultcode ”Client.TxHeader.TxId.Duplicate”. Vid asynkron bearbetning ansvarar mottagaren av begäran för att TxId flödar med oförändrat genom hela transaktionen.

3 Meddelande

Meddelanden i kommunikation enligt SSEK följer standarden SOAP 1.1.

3.1 Namespaces

Vid skapande av meddelanden enligt SSEK SKALL följande namespaces användas.

Prefix	Namespace	Används vid:
Soap	http://schemas.xmlsoap.org/soap/envelope/	Samtliga fall
Txh	http://schemas.ssek.org/txheader/2003-04-03/	Samtliga fall
Wsu	http://schemas.xmlsoap.org/ws/2002/07/utility	Signering
Wsse	http://schemas.xmlsoap.org/ws/2002/07/secext	Signering
	http://www.w3.org/2000/09/xmldsig#	Signering
R	http://schemas.ssek.org/receipt/2003-04-03/	Kvitto

Prefixen används här för att förtydliga i dokumentets exempel.

3.2 Algoritmer

Följande algoritmer används i detta dokument. Dessa algoritmer SKALL användas.

Use	Algorithm
CanonicalizationMethod	http://www.w3.org/2001/10/xml-exc-c14n#
SignatureMethod	http://www.w3.org/2000/09/xmldsig#rsa-sha1
Transform	http://www.w3.org/2001/10/xml-exc-c14n#
DigestMethod	http://www.w3.org/2000/09/xmldsig#sha1

3.3 Certifikat i signerade meddelanden

I signerade meddelanden SKALL det certifikat som använts för skapande av signaturen bifogas i meddelandet enligt specifikationen för WS-Security, se *Bilaga B Web Services Security*. Bifogat certifikat kan användas av mottagaren för att validera signaturen. Mottagaren av ett signerat meddelande BÖR kontrollera att bifogat certifikat är korrekt.

3.4 Tekniskt meddelandehuvud

I varje meddelande SKALL ett meddelandehuvud med teknisk information, TxHeader, ingå enligt schema i *Bilaga C SSEK-specifika scheman* med följande namespace: http://schemas.ssek.org/txheader/2003-04-03/.

Informationen syftar till att kunna styra meddelanden, hantera asynkron kommunikation samt få en hanterbar logg över kommunicerade dokument.

Fältnamn	Beskrivning	Attribut
SenderId	Id för avsändaren av dokumentet.	**Type:** typ av identitet som används för avsändaren av ett meddelande.
ReceiverId	Id för mottagaren av dokumentet.	**Type:** typ av identitet som används för mottagaren av ett meddelande.
TxId	UUID (Universal Unique IDentifier enligt DCE).	
Timestamp	Tidpunkt då meddelandet skapats enligt formen ssåå-mm-ddThh:mm:ss Där T är konstant	

Fältet TxId i meddelandehuvudet används främst för att hålla samman en asynkron transaktion. TxId är inte obligatorisk utan SKALL överenskommas mellan de kommunicerande parterna för varje tjänst baserat på affärsbehov.

Vilken typ av identitet som används för SenderId och ReceiverId SKALL överenskommas mellan de kommunicerande parterna. Vilka typer som kan användas definieras av schemat för Txheader och förklaras i nedanstående tabell:

Värde	Förklaring
APP	Applikationsnamn, kan vara egna klienter där man vill ha spårbarhet på vilken applikation som används
CN	Common Name, hämtas från klient-cerifikatet
DN	Distinguished Name, hela namnet för klient-certifikatet
ORGNR	Organisationsnummer

Nedan följer ett exempel på ett tekniskt meddelandehuvud, TxHeader:

```
<txh:TxHeader soap:mustUnderstand="1"
      xmlns:txh="http://schemas.ssek.org/txheader/2003-04-03/">
    <txh:SenderId txh:type="CN">Företag A</txh:SenderId>
    <txh:ReceiverId txh:type="CN">Företag B</txh:ReceiverId>
    <txh:TxId>C61B0B07-EF5F-46a1-92B4-6E5FA574E46E</txh:TxId>
    <txh:Timestamp>2003-03-27T12:50:00</txh:Timestamp>
  </txh:TxHeader>
```

3.5 Kvitto

Syftet med kvittot är att avsändaren skall vara säker på att frågemeddelandet har tagits emot av mottagaren. Kvitto SKALL användas vid asynkron kommunikation när ett meddelande tagits emot och godkänts för bearbetning hos mottagaren. En kvittens garanterar dock inte att informationen i det mottagna meddelandet kommer att bearbetas utan fel.

Vid kvittering av ett signerat meddelande SKALL kvittot innehålla det kvitterade meddelandets signatur. Signaturen innehåller en representation av det ursprungliga meddelandet i form av ett hashvärde. Representation i form av signaturen från det mottagna meddelandet kan naturligtvis endast användas om det mottagna meddelandet är signerat.

Kvittot kan även användas för att meddela resultat av bearbetning synkront. Schemat tillåter att bolagsspecifik information kan läggas till.

Namespace för kvitto är http://schemas.ssek.org/receipt/2003-04-03/. Se även schema i *Bilaga C SSEK-specifika scheman*. Innehållet är enligt följande:

Fältnamn	**Beskrivning**
ResponseCode	Svarskod (OK)
ResponseMessage	Svarsmeddelande t.ex. ”Meddelandet mottaget och kommer att bearbetas”
RequestSignatureValue	Det kvitterade frågemeddelandets signatur.

Nedan följer ett exempel på ett osignerat kvitto (för en begäran):

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <txh:TxHeader soap:mustUnderstand="1"
      xmlns:txh="http://schemas.ssek.org/txheader/2003-04-03/">
      <txh:SenderId txh:type="CN">Företag B</txh:SenderId>
      <txh:ReceiverId txh:type="CN">Företag A</txh:ReceiverId>
      <txh:TxId>C61B0B07-EF5F-46a1-92B4-6E5FA574E46E</txh:TxId>
      <txh:Timestamp>2003-03-27T12:50:01</txh:Timestamp>
    </txh:TxHeader>
  </soap:Header>
  <soap:Body>
    <r:Receipt xmlns:r="http://schemas.ssek.org/receipt/2003-04-03/">
      <r:ResponseCode>OK</r:ResponseCode>
      <r:ResponseMessage>Meddelandet är mottaget och kommer att
        bearbetas</r:ResponseMessage>
        <r:RequestSignatureValue>X1os6m3YReQJlk1JBzwvLe5hScdz09uBx4EhnJ
        plKZVl2uScMpj4FPnnyfIPBJ3vkI59aPcbQlzqTEgaHBwiIpcwRNnnOJ4tYe
        Y3/HekPSlBFB9pOFqh73O8qqK0v1/ZK2IYOBT/WBFNdYD6nf8gP8nSjAK
        UPx1QFC8RCUKVeZk=</r:RequestSignatureValue>
    </r:Receipt>
  </soap:Body>
</soap:Envelope>
```

Nedan följer ett exempel på ett signerat kvitto (för en begäran):

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Header>
      <txh:TxHeader soap:mustUnderstand="1"
            xmlns:txh="http://schemas.ssek.org/txheader/2003-04-03/"
            xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility"
            wsu:Id="txHeader">
          <txh:SenderId txh:type="CN">Företag B</txh:SenderId>
          <txh:ReceiverId txh:type="CN">Företag A</txh:ReceiverId>
          <txh:TxId>C61B0B07-EF5F-46a1-92B4-6E5FA574E46E</txh:TxId>
          <txh:Timestamp>2003-03-27T12:50:01</txh:Timestamp>
       </txh:TxHeader>
      <wsse:Security soap:mustUnderstand="1"
            xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/07/secext">
          <wsse:BinarySecurityToken ValueType="wsse:X509v3"
              EncodingType="wsse:Base64Binary"
              xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility"
              wsu:Id="SecurityToken-aca9eb37-8ba1-4ca4-8a98-
              50df7b66b87f">MIICcjCCAdugAwIBAgIBAzANBgkqhkiG9w0B...
              </wsse:BinarySecurityToken>
        <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
          <SignedInfo>
            <CanonicalizationMethod
                Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            <SignatureMethod
                Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
            <Reference URI="#txHeader">
              <Transforms>
                <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
                    c14n#" />
              </Transforms>
              <DigestMethod
                  Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
                  DigestValue>9lvyLVeHMTWsmyP0tEvjaAAuN0w=</DigestValue>
            </Reference>
            <Reference URI="#soapBody">
              <Transforms>
                <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
                    c14n#" />
              </Transforms>
              <DigestMethod
                  Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
                  DigestValue>CTxmSaf2GFELSDEs7qkZE8PLWn0=</DigestValue>
            </Reference>
          </SignedInfo>
          SignatureValue>bCTgHvWanAVpqTD9MCp1zK6AlD7lDi6r18STl+4Mco...
              </SignatureValue>
          <KeyInfo>
            <wsse:SecurityTokenReference>
              <wsse:Reference URI="#SecurityToken-aca9eb37-8ba1-4ca4-
                  8a98-50df7b66b87f" />
            </wsse:SecurityTokenReference>
          </KeyInfo>
        </Signature>
      </wsse:Security>
    </soap:Header>
    <soap:Body xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility"
        wsu:Id="soapBody">
      <r:Receipt xmlns:r="http://schemas.ssek.org/receipt/2003-04-03/">
          <r:ResponseCode>OK</r:ResponseCode>
          <r:ResponseMessage>Meddelandet är mottaget och kommer att
              bearbetas</r:ResponseMessage>
          <r:RequestSignatureValue>X1os6m3YReQJlk1JBzwvLe5hScdz09uBx4JplKZ...
              </r:RequestSignatureValue>
      </r:Receipt>
    </soap:Body>
  </soap:Envelope>
```

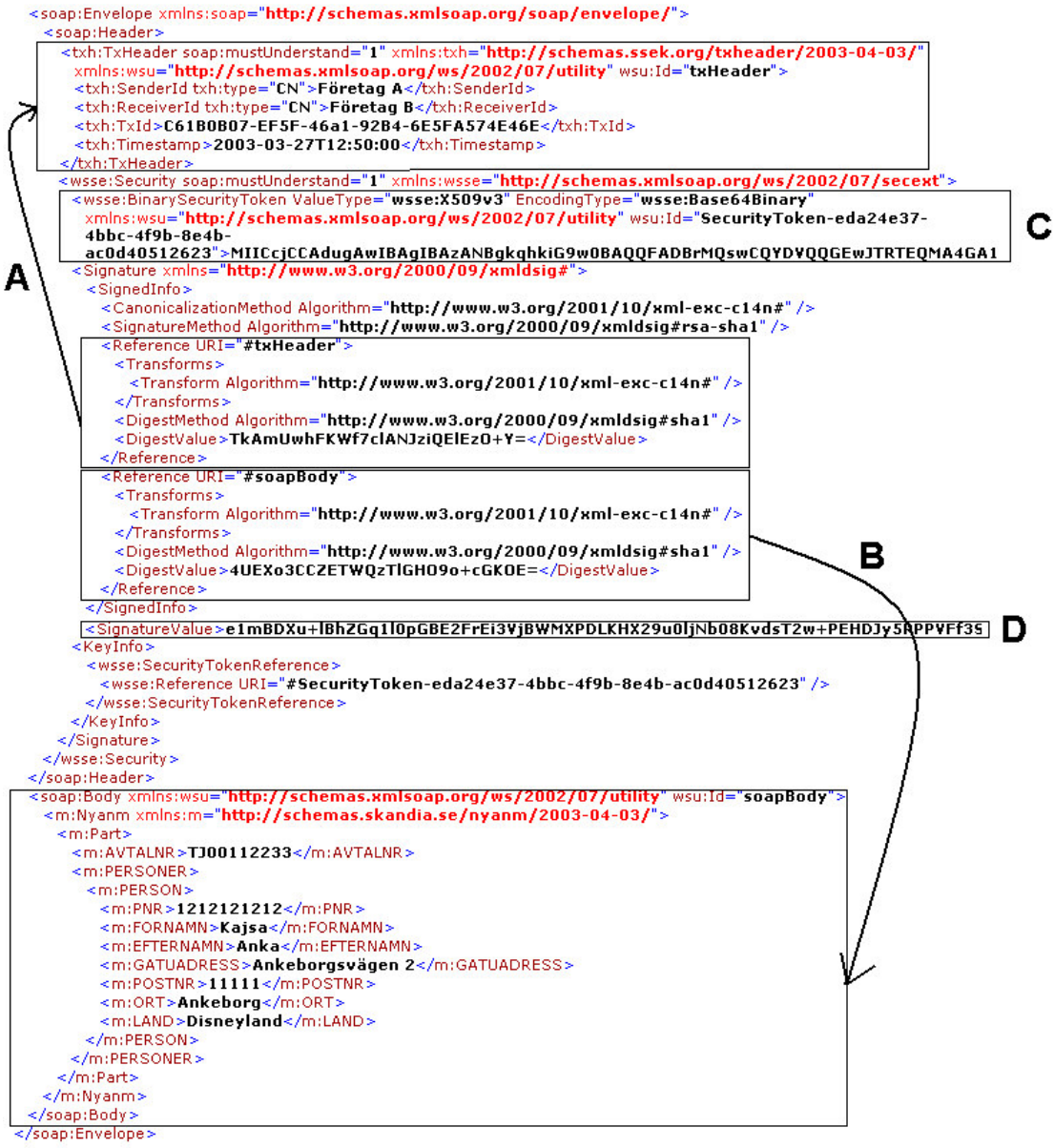
3.6 Soap fault

Vid fel som uppstår och som kan meddelas synkront för ett frågemeddelande SKALL mottagaren returnera ett soap fault-meddelande med en faultcode enligt *Bilaga D Felkoder*. Soap fault-meddelandet KAN vara signerat.

Nedan följer ett exempel på ett osignerat soap fault (för en begäran):

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <soap:Fault>
          <soap:faultcode>Client.InvalidXml</soap:faultcode>
          <soap:faultstring>Non valid XML</soap:faultstring>
          <soap:detail>Element 'PNR' more than 12 digits</soap:detail>
       </soap:Fault>
    </soap:Body>
 </soap:Envelope>
```

3.7 Exempel på signerat frågemeddelande




```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Header>
  <txh:TxHeader soap:mustUnderstand="1" xmlns:txh="http://schemas.ssek.org/txheader/2003-04-03/"
   xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility" wsu:Id="txHeader">
   <txh:SenderId txh:type="CN">Företag A</txh:SenderId>
   <txh:ReceiverId txh:type="CN">Företag B</txh:ReceiverId>
   <txh:TxId>C61B0B07-EF5F-46a1-92B4-6E5FA574E46E</txh:TxId>
   <txh:Timestamp>2003-03-27T12:50:00</txh:Timestamp>
  </txh:TxHeader>
  <wsse:Security soap:mustUnderstand="1" xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/07/secext">
   <wsse:BinarySecurityToken ValueType="wsse:X509v3" EncodingType="wsse:Base64Binary"
    xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility" wsu:Id="SecurityToken-eda24e37-
    4bbc-4f9b-8e4b-
    ac0d40512623">MIICcjCCAdugAwIBAgIBAzANBgkqhkiG9w0BAQQFADBrMQswCQYDVQQGEwJTRTEQMA4GA1
   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
     <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
     <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
     <Reference URI="#txHeader">
      <Transforms>
       <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <DigestValue>TkAmUwhFKWf7clANJziQElEzO+Y=</DigestValue>
     </Reference>
     <Reference URI="#soapBody">
      <Transforms>
       <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <DigestValue>4UEXo3CC2ETWQzTlGHO9o+cGKOE=</DigestValue>
     </Reference>
    </SignedInfo>
    <SignatureValue>e1mBDXu+lBhZGq1l0pGBE2FrEi3YjBWMXPDLKHX29u0ljNb08KvdsT2w+PEHDJy5RPPYFf35
    <KeyInfo>
     <wsse:SecurityTokenReference>
      <wsse:Reference URI="#SecurityToken-eda24e37-4bbc-4f9b-8e4b-ac0d40512623" />
     </wsse:SecurityTokenReference>
    </KeyInfo>
   </Signature>
  </wsse:Security>
 </soap:Header>
 <soap:Body xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility" wsu:Id="soapBody">
  <m:Nyanm xmlns:m="http://schemas.skandia.se/nyanm/2003-04-03/">
   <m:Part>
    <m:AVTALNR>TJ00112233</m:AVTALNR>
    <m:PERSONER>
     <m:PERSON>
      <m:PNR>1212121212</m:PNR>
      <m:FORNAMN>Kajsa</m:FORNAMN>
      <m:EFTERNAMN>Anka</m:EFTERNAMN>
      <m:GATUADRESS>Ankeborgsvägen 2</m:GATUADRESS>
      <m:POSTNR>11111</m:POSTNR>
      <m:ORT>Ankeborg</m:ORT>
      <m:LAND>Disneyland</m:LAND>
     </m:PERSON>
    </m:PERSONER>
   </m:Part>
  </m:Nyanm>
 </soap:Body>
</soap:Envelope>
```


Pilen A pekar från hashvärdet av headern till den del som hashvärdet representerar.

Pilen B pekar från hashvärdet av dokumentets body till den del som hashvärdet representerar.

Ruta C innehåller det certifikat som använts för skapandet av signaturen.

Ruta D innehåller signaturen av innehåller SignedInfo.

Själva signaturen ligger alltså i SignatureValue vilken signerar informationen i SignedInfo, inklusive de hashvärden som representerar information i dokumentets header och body.

3.8 Exempel på osignerat frågemeddelande

Nedan följer ett exempel på ett osignerat frågemeddelande (för en begäran):

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <txh:TxHeader soap:mustUnderstand="1"
        xmlns:txh="http://schemas.ssek.org/txheader/2003-04-03/">
      <txh:SenderId txh:type="CN">Företag A</txh:SenderId>
      <txh:ReceiverId txh:type="CN">Företag B</txh:ReceiverId>
      <txh:TxId>C61B0B07-EF5F-46a1-92B4-6E5FA574E46E</txh:TxId>
      <txh:Timestamp>2003-03-27T12:50:00</txh:Timestamp>
    </txh:TxHeader>
  </soap:Header>
  <soap:Body>
    <m:Nyanm xmlns:m="http://schemas.foretagb.se/nyanm/2003-04-03/">
      <m:Part>
        <m:AVTALNR>TJ00112233</m:AVTALNR>
        <m:PERSONER>
          <m:PERSON>
            <m:PNR>1212121212</m:PNR>
            <m:FORNAMN>Kajsa</m:FORNAMN>
            <m:EFTERNAMN>Anka</m:EFTERNAMN>
            <m:GATUADRESS>Ankeborgsvägen 2</m:GATUADRESS>
            <m:POSTNR>11111</m:POSTNR>
            <m:ORT>Ankeborg</m:ORT>
            <m:LAND>Disneyland</m:LAND>
          </m:PERSON>
        </m:PERSONER>
      </m:Part>
    </m:Nyanm>
  </soap:Body>
</soap:Envelope>
```

4 Säkerhet

Då informationen som kommuniceras är känslig och värdefull finns det behov av att skydda den från att obehöriga kan läsa eller manipulera informationen. Det kan också finnas behov av att kunna härleda information till avsändaren.

För att uppfylla ovan nämnda behov beskriver SSEK hur följande säkerhetsaspekter hanteras vid elektronisk kommunikation.

- **Autentisering**
  Identifiering av avsändande organisation (klient) och mottagande organisation (server).
- **Konfidentialitet**
  Informationen som skickas är inte läsbar av utomstående parter.
- **Riktighet och Integritet**
  Information som skickas är oföränderlig efter det att den lämnat avsändaren samt följer överenskommet format.
- **Oavvislighet**
  Den part som skapat och skickat information kan inte förneka att just den informationen skapats.

PKI (Public Key Infrastructure) utnyttjas vid hantering av ovan nämnda säkerhetsaspekter.

4.1 Säkerhetsnivåer

För att uppfylla alla, eller endast vissa, av ovanstående beskrivna säkerhetsaspekter kan signaturer och klientcertifikat kombineras. De tillåtna kombinationerna är definierade i fyra olika säkerhetsnivåer där nivå 4 har starkast säkerhet och nivå 1 svagast säkerhet.

Gemensamt för alla SSEK-säkerhetsnivåer är att en SSL-tunnel sätts upp, genom vilken konfidentialitet vid överföringen erhålls, och att servern autentiserar sig med ett servercertifikat.

Ytterligare information om certifikat och signaturer finns i avsnitt *4.3 Signering* och *4.4 CA och certifikat*.

SSEK-nivå	**SSL – med servercertifikat**	**Signatur**	**Klientcertifikat**
4	Används	Används	Används
3	Används	Används	
2	Används		Används
1	Används		

Information om varje SSEK-säkerhetsnivåer

1. Ger konfidentialitet vid överföring och autentisering av servern.
   Ger **inte** autentisering av klienten eller oavvislighet på grund av att signaturer och klientcertifikat inte används.
   Lämplig nivå för publika tjänster.
2. Ger konfidentialitet vid överföringen samt autentisering av både server och klient.
   Ger **inte** oavvislighet på grund av att signaturer inte används.
   Lämplig nivå för tjänster som riktar sig till vissa behöriga organisationer men där skickad information inte, i efterhand, behöver användas i bevissyfte.
3. Ger konfidentialitet vid överföring, autentisering av servern och oavvislighet.
   Ger **inte** autentisering av klienten på grund av att klientcertifikat inte används.
   Lämplig nivå för tjänster där avsändaren inte behöver autentiseras men där mottagaren har behov av att kunna bevisa vilken organisation som skapat den information som tas emot.
4. Ger konfidentialitet vid överföringen, autentisering av både server och klient samt oavvislighet. Både signaturer och klientcertifikat används.
   Lämplig nivå för känsliga tjänster som riktar sig till vissa behöriga organisationer och där mottagaren har behov av att kunna bevisa vilken organisation som skapat den information som tas emot.

4.2 Säker kommunikation

Båda parter kan initiera kommunikationen och gör det genom att koppla upp en så kallad säker tunnel mellan sig själv och motparten. Den säkra tunneln SKALL skapas med SSL varvid protokollet HTTPS (HTTP över SSL) används. Vid detta tillfälle autentiseras båda parterna, alltså både klient och server-autentisering, med hjälp av sina certifikat.

Tunneln SKALL vara krypterad med en säker nyckel.
För tillfället gäller minst en 128 bitars nyckel.

4.3 Signering

En digital signatur ger möjlighet att tekniskt bevisa vilken organisation som skapat viss information. Då meddelanden signeras SKALL de direktiv som beskrivs i detta dokument avseende signaturer följas.

För att skapa signaturer SKALL certifikat användas.

4.4 CA och Certifikat

De certifikat som krävs för att skapa signaturen och autentisera respektive part SKALL vara utgivet av en, av båda parter, godkänd CA (Certificate Authority) samt vara godkänt avseende typ och rutiner för utgivande av certifikat. Certifikaten SKALL följa standarden X.509 v3 och ha en tillräckligt säker publik nyckel. För tillfället gäller minst en 1024 bitars nyckel.

Användning av digitala signaturer bygger på det förtroende de kommunicerande parterna har till den CA som utfärdat certifikaten.

4.5 Revokering av certifikat

Om ett certifikat inte längre kan anses tillförlitligt SKALL de parter som utnyttjar certifikatet informeras om detta.

Den part som utnyttjar ett certifikat, för verifiering av en organisations identitet och signaturer, BÖR hämta den CRL (Certificate Revocation List) certifikatets CA ger ut samt kontrollera att certifikatet inte har revokerats varje gång certifikatet utnyttjas. Revokering av certifikatet hos certifikatets CA SKALL utföras av certifikatets ägare. Vid revokering läggs certifikatet in i den CRL som certifikatets CA regelbundet ger ut, exempelvis varannan timme.

Om inte rutiner och teknik för automatisk revokeringskontroll är på plats för att hämta och läsa en CRL SKALL ”manuell revokering” utnyttjas. Denna hantering utförs genom att den part som vill göra ett certifikat otillåtet så snart som möjligt, förslagsvis per telefon, kontaktar en ansvarig person hos den andra parten som då tar bort certifikatet så att det inte kan användas. Certifikatet SKALL i detta fall även revokeras hos certifikatets CA. Därmed fungerar det även för dem som faktiskt utnyttjar den CRL som ges ut.

Om ”manuell revokering” utnyttjas mellan två kommunicerande parter SKALL rutiner för vem som gör vad hos båda parter specificeras. Båda parter skall vara överens om de specificerade rutinerna.

Om ingen rutin för revokering fastslagits ökar risken för att en bedräglig part, som stulit certifikatets privata nyckel, kan utnyttja nyckeln under en lång period innan information går ut om att certifikatet är stulet och inte kan betraktas som en säker identifiering av motparten.

4.6 Godkända CA och certifikatstyper

Idag är Telia eCommerce (ger ut certifikat med Verisign som CA) och Posten eSäkerhet godkända som certifikatsutgivare av parter som utnyttjar denna specifikation för sin elektroniska kommunikation.

Även typen av certifikat som ges ut av godkända CA och används hos parterna SKALL vara godkänd för användning av båda parter. Kravet på certifikatstypen är i så fall att en tillräckligt säker validering av organisationens identitet utförs vid utgivning av certifikatet.

Utnyttjas en osäker CA respektive certifikat utan tillräckligt god validering av organisationens identitet ökar risken för att en bedräglig organisation lyckas med att utge sig för att vara motparten vid kommunikation. Detta kan få stora konsekvenser.

4.7 Hantering av privata nycklar

Autentisering och signering utförs med en privat nyckel som är kopplad till ett certifikat. Den gjorda autentiseringen respektive signaturen används sedan för att bevisa att viss information skapats och skickats av den avsändande parten. Därför SKALL den privata nyckeln skyddas på ett tillräckligt säkert sätt så att obehöriga inte stjäl nyckeln.

Skyddas inte de privata nycklarna på ett tillräckligt bra sätt ökar risken för stöld av nycklarna. En bedräglig part som stjäl nyckeln kan utge sig för att vara den organisation nyckeln tillhör och agera i dess ställe. Konsekvenserna kan alltså bli stora.

5 Bilaga A Standarder

[IETF]
IETF (Internet Engineering Task Force), http://www.ietf.org/

[OASIS]
OASIS (Organization for the Advancement of Structured Information Standards), http://www.oasis-open.org/

[PKI certifikat]
Public Key Infrastructure certifikat, http://www.ietf.org/html.charters/pkix-charter.html

[Posten]
Posten http://digitalid.postnet.se

[SHA-1]
http://www.itl.nist.gov/fipspubs/fip180-1.htm

[SOAP]
W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.

[SSL]
Secure Sockets Layer http://home.netscape.com/security/techbriefs/ssl.html

[WS-Security]
W eb Services Security (WS-Security), Version 1.0 05 April 2002,
http://www-106.ibm.com/developerworks/webservices/library/ws-secure/

[W3C]
W3C (World Wide Web Consortium), http://www.w3.org/

[X.509 v3]
X.509 v3, http://www.ietf.org/rfc/rfc2459.txt

[XML]
Extensible Markup Language (XML) 1.0 (Second Edition),
W3C Recommendation 6 October 2000, http://www.w3.org/TR/2000/REC-xml-20001006

[XML-C14N]
Exclusive XML Canonicalization,
Version 1.0, W3C Recommendation 18 July 2002, http://www.w3.org/TR/2002/REC-xml-exc-c14n-20020718/

[XML-Schema1]
W3C Recommendation, "XML Schema Part 1: Structures," 02 May 2001.

[XML-Schema2]
W3C Recommendation, "XML Schema Part 2: Datatypes," 02 May 2001.

[XML Signature]
http://www.w3.org/Signature/

6 Bilaga B Web Services Security

Följande delar av specifikationen för WS-Security, används inom SSEK.

Stycke	Förtydligande
4.2 Encoding Binary Security Tokens	Endast X.509 certifikat används. X.509 certifikat specificeras genom att använda BinarySecurityToken: <wsse:BinarySecurityToken xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext" Id="myToken" ValueType="wsse:X509v3" EncodingType="wsse:Base64Binary"> MIIEZzCCA9CgAwIBAgIQEmtJZc0... </wsse:BinarySecurityToken>
4.3 SecurityTokenReference Element	
4.5 ds:Signature	

Följande delar av specifikationen för Web Services Security Addendum, används inom SSEK.

Stycke	Förtydligande
3. ID References	
3.2 Id Schema	<x:myElement wsu:Id="ID1" xmlns:x="..." xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility"/>
4. Placement of X.509 Certificates	X.509 certifikat specificeras genom att använda BinarySecurityToken

7 Bilaga C SSEK-specifika scheman

Schema för TxHeader:

```
<xsd:schema targetNamespace="http://schemas.ssek.org/txheader/2003-04-03/"
      xmlns:tns="http://schemas.ssek.org/txheader/2003-04-03/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
      attributeFormDefault="qualified">
   <xsd:element name="TxHeader">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="SenderId">
               <xsd:complexType>
                  <xsd:simpleContent>
                     <xsd:restriction base="xsd:anyType">
                        <xsd:simpleType>
                           <xsd:restriction base="xsd:string">
                              <xsd:maxLength value="256" />
                           </xsd:restriction>
                        </xsd:simpleType>
                        <xsd:attribute name="type" use="optional" default="CN">
                           <xsd:simpleType>
                              <xsd:restriction base="xsd:string">
                                 <xsd:enumeration value="APP" />
                                 <xsd:enumeration value="CN" />
                                 <xsd:enumeration value="DN" />
                                 <xsd:enumeration value="ORGNR" />
                                 <xsd:maxLength value="16" />
                              </xsd:restriction>
                           </xsd:simpleType>
                        </xsd:attribute>
                     </xsd:restriction>
                  </xsd:simpleContent>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="ReceiverId">
               <xsd:complexType>
                  <xsd:simpleContent>
                     <xsd:restriction base="xsd:anyType">
                        <xsd:simpleType>
                           <xsd:restriction base="xsd:string">
                              <xsd:maxLength value="256" />
                           </xsd:restriction>
                        </xsd:simpleType>
                        <xsd:attribute name="type" use="optional" default="CN">
                           <xsd:simpleType>
                              <xsd:restriction base="xsd:string">
                                 <xsd:enumeration value="APP" />
                                 <xsd:enumeration value="CN" />
                                 <xsd:enumeration value="DN" />
                                 <xsd:enumeration value="ORGNR" />
                                 <xsd:maxLength value="16" />
                              </xsd:restriction>
                           </xsd:simpleType>
                        </xsd:attribute>
                     </xsd:restriction>
                  </xsd:simpleContent>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="TxId" minOccurs="0">
               <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                     <xsd:length value="36" />
                  </xsd:restriction>
               </xsd:simpleType>
            </xsd:element>
```

```
            <xsd:element name="Timestamp" type="xsd:dateTime" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
  </xsd:schema>
```

Schema för kvitto:

```
<xsd:schema targetNamespace="http://schemas.ssek.org/receipt/2003-04-03/"
      xmlns:tns="http://schemas.ssek.org/receipt/2003-04-03/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xsd:element name="Receipt">
      <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="ResponseCode" type="xsd:string" />
            <xsd:element name="ResponseMessage" type="xsd:string"
                minOccurs="0" />
            <xsd:element name="RequestSignatureValue" type="xsd:string"
                minOccurs="0" />
            <xsd:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
                processContents="lax" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
  </xsd:schema>
```

8 Bilaga D Felkoder

Här definieras de felkoder, faultcode, som kan sättas i soap fault-meddelanden. Vilka värden som ska sättas i Faultstring och Detail definieras inte här, dessa kan sättas till valfria värden som överenskommes mellan de kommunicerande parterna. Schema för soap fault finns under: **http://schemas.xmlsoap.org/soap/envelope/**

8.1 Fel avseende dokumentet

Faultcode	**Beskrivning**
VersionMismatch	En okänd version av soap angavs
MustUnderstand.TxHeader	Felaktigt format på TxHeader
MustUnderstand.Security	Felaktigt format på securityheader
Client.InvalidXml	Felaktigt format på meddelande
Client. WebService.Unknown	Kombinationen av TxHeader, MessageName och nsuri för meddelandet är felaktig.
Client.TxHeader.SenderId.Unknown	
Client.TxHeader.ReceiverId.Unknown	
Client.TxHeader.Timestamp.Invalid	

8.2 Fel avseende transaktionsid

Faultcode	**Beskrivning**
Client.TxHeader.TxId.Missing	TxId har ej angivits
Client.TxHeader.TxId.Unknown	
Client.TxHeader.TxId.Invalid	Fel format på TxId
Client.TxHeader.TxId.Duplicate	TxId är inte unikt
Client.TxHeader.TxId.NotAllowed	TxId ska ej anges för denna tjänst

8.3 Fel på serversidan

Faultcode	**Beskrivning**
Server.MessageNotProcessed	Internt fel
Server.WebService.Unavailable	Tjänsten är stängd
Server.WebService.Unsupported	Tjänsten stöds ej av mottagaren

8.4 Logiska fel i affärsdata

Faultcode	**Beskrivning**
Client.Mottargarnamn.valfri felkod	Fel i affärsdata avseende affärslogik, t.ex. för hög lön, kan meddelas på detta sätt. T.ex. ”Client.Skandia. InvalidSalary”

8.5 Fel avseende certifikat eller signatur

Faultcode	Beskrivning
Client.UnregisteredClientCertificate	DN i http-header i client-certificate överensstämmer ej med lagrat DN.
Client.InvalidSecurityToken	Certifikatet är ogiltigt med avseende på giltighetsperiod.
Client.InvalidSecurityToken	Certifikatet är revokerat
Client.FailedAuthentication	Angiven security token kunde inte autenticeras eller auktoriseras.
Client.UnsupportedSecurityToken	En okänd security token angavs, ej X509
Client.UnregisteredSecurityToken	DN I signaturcert överensstämmer ej med lagrat DN.
Client.UnsupportedAlgorithm	En okänd algoritm för signatur eller kryptering angavs
Client.InvalidSecurity	Ett fel upptäcktes vid bearbetning av <Security> headern
Client.InvalidSecurityToken	
Client.FailedCheck	
Client.SecurityTokenUnavailable	

8.6 Övriga fel som inte meddelas via SOAP-fault

Faultcode	Beskrivning
http	Server unavailable
TimeOut	Klienten får inget svar