SSEK version 2.0

Säkra webbtjänster för affärskritisk kommunikation

2006-05-10

Mats Andersson, Skandia Liv
Peter Danielsson, Skandia Liv
Gustaf Nyman, Skandia Liv

Sammanfattning

SSEK 2.0 specificerar hur säkra tjänster för informationsutbyte mellan organisationer utformas. I och med praktiskt utnyttjande har SSEK visat sig uppfylla de affärs- och säkerhetsmässiga samt juridiska behov som finns för elektronisk affärskommunikation. Genom att följa SSEK ökar en organisations möjlighet att skapa säker och kompatibel affärskommunikation med andra organisationer.

SSEK bygger på ett antal vedertagna standarder för tjänster över Internet.

Jämfört med SSEK 1.1 innehåller SSEK 2.0 flera förbättringar i funktionalitet samt uppdatering till gällande version avseende de standarder SSEK bygger på.

Innehållsförteckning

1 Inledning

Detta dokument definierar hur säkra tjänster produceras och konsumeras över Internet enligt SSEK 2.0. Specifikationen är resultatet av behov som identifierats i försäkringsbranschen, men kan med fördel användas i alla sammanhang där följande faktorer är av vikt:

- Tekniskt säkert informationsutbyte mellan organisationer.
- Interoperabilitet mellan implementationer och plattformar
- Säkert kunna omsätta avtalade tjänster i praktiskt tjänsteutnyttjande.

SSEK-specifikationen är uppbyggd kring riktlinjer runt vedertagna specifikationer från W3C, IETF, OASIS, WS-I med flera i kombination med ett antal specifika tillägg som är unika för SSEK. SSEK definierar bland annat:

- Adressering på organisationsnivå
- Hantering av meddelandeflöden och omsändning
- Standardstrukturer för fel och kvitton
- Riktlinjer för säkerhet, meddelanden, metadata, bilagor och interoperabilitet.

Detta dokuments syfte är att så exakt som möjligt beskriva vad SSEK är respektive inte är. Dokumentet är i första hand avsett för dem som bygger systemprogramvara för SSEK.

Författarna tar inget ansvar för hur SSEK används i praktiken. Författarna garanterar ingenting i bevisvärde vid en eventuell tvist avseende förhållanden som dokumenterats enligt SSEK.

1.1 SSEK arbetsgrupp

SSEK 2.0 har tagits fram av SSEK Arbetsgrupp som består av:

- Mats Andersson, Skandia Liv
- Lars Boshuis, SEB
- Peter Danielsson, Skandia Liv
- Johan Lidö, SEB
- Gustaf Nyman, Skandia Liv

1.2 Definitioner

Följande begrepp används i dokumentet med betydelse enligt nedan.

Begrepp	Betydelse
Meddelande	SOAP-meddelande [SOAP11][BP11] som följer SSEK.
Avsändare	Part som skickar meddelande enligt SSEK
Mottagare	Part som tar emot meddelande enligt SSEK
Tjänst	Webbtjänst som stöder SSEK.
Producent	Part som tillhandahåller värdeskapande tjänst enligt SSEK.
Konsument	Part som använder tjänst enligt SSEK
Soapheader	Elementet soap:Header enligt [SOAP11] och [BP11]
Headerelement	Element placerat under Soapheader
Soapbody	Elementet soap:Body enligt [SOAP11] och [BP11]
Meddelandeinnnehåll	Meddelandeinnehåll, eller payload, är den del av ett meddelande som placerats under Soapbody och utgör det väsentliga innehållet i ett meddelande.
Soapfault	Meddelande med meddelandeinnehåll bestående av soap:Fault enligt [SOAP11]
Felkod	faultcode-elementet i ett soapfault

1.3 Konventioner och 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 Namespaces och prefix

I dokumentet används prefix enligt tabellens definition.

Prefix	Namespace
soap	http://schemas.xmlsoap.org/soap/envelope/
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
wsse11	http://docs.oasis-open.org/wss/oasis-wsswssecurity-secext-1.1.xsd
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
ssek	http://schemas.ssek.org/ssek/2006-05-10/
ssekp	http://schemas.ssek.org/ ssek/2006-05-10/policy

2 Grunderna i SSEK

För transport av meddelanden inom ramen för SSEK används SOAP 1.1.

B001 Meddelanden SKALL vara kompatibla med SOAP 1.1 Envelopes [SOAP11]

Specifikationen WS-I Basic Profile Version 1.1 [BP11] är resultatet av praktiska erfarenheter kring problem med samverkan mellan olika implementationer av webbtjänster. Samverkansproblem beror i huvudsak på oklarheter i grundläggande specifikationerna som [SOAP11] och [WSDL]. [BP11] ger konkreta riktlinjer för hur webbtjänster skall konstrueras för god samverkan med andra system.

B002 WS-I Basic Profile Version 1.1 [BP11] inklusive korrigeringar [BP11ERR] SKALL följas förutom i explicit beskrivna fall i denna specifikation.

SSEK skiljer sig från [BP11] i några avseenden.

B003 UDDI SKALL INTE användas för beskrivning av tjänster.

B004 Endast document-literal bindning SKALL användas.

B005 Endast ett element SKALL förekomma under soapbody.

WS-Addressing [WSA][WSASB] definierar användandet så kallade endpoints för identifiering av avsändare och mottagare. Denna information är inte nödvändig för adressering inom ramen för SSEK, där adressering istället sker på en mer abstrakt nivå baserad på avsändar- och mottagaridentiteter. Dock kan adressering enligt SSEK och adressering enligt WS-Addressing samverka utan problem.

B006 WS-Addressing [WSA][WSASB] KAN användas tillsammans med SSEK.

3 Meddelandestrukturer

3.1 SSEK

SSEK-elementet är ett headerelement som används för att förmedla information om avsändare och mottagare av ett meddelande samt vilket meddelandeflöde meddelandet ingår i.

TX001 SSEK-elementet är ett headerelement som SKALL inkluderas meddelanden, med undantag för soapfault.

TX002 SSEK-elementet SKALL ha attributet soap:mustUnderstand med värde 1.

Förenklad beskrivning av SSEK-elementet. För exakt beskrivning se bifogat schema.

```
<ssek:SSEK ssek:AsynchMethod="AsynchPush|AsynchPull" soap:mustUnderstand="1" {any}?>
  <ssek:SenderId ssek:Type="CN|DN|ORGNR|APP">...</ssek:SenderId>
  <ssek:ReceiverId ssek:Type="CN|DN|ORGNR|APP">...</ssek:ReceiverId>
  <ssek:TxId>...</ssek:TxId>?
</ssek:SSEK>
```

/ssek:SSEK

Detta element är ett headerelement och SKALL placeras under soapheader.

/ssek:SSEK/@ssek:AsynchMethod

AsynchMetod är ett valfritt attribut som indikerar vilken asynkron kommunikation en konsument vill använda. Attributet får användas när aktuell tjänst är av asynkron karaktär och skall innehålla ett av de värden som definieras i tjänstens policy. Kan ha värdet AsynchPush eller AsynchPull.

/ssek:SSEK/ssek:SenderId

SenderId anger avsändarens identitet. Hur innehållet i SenderId skall tolkas styrs av ssek:Type-attributet.

/ssek:SSEK/ssek:SenderId/@ssek:Type='CN'

Attributet Type är ett valfritt attribut som definierar typ av innehåll i SenderId-elementet. Type kan ha någon av värdena CN, DN, ORGNR och APP. Anges inget värde så antas den ha grundvärdet CN. En tjänsts policy kan styra vilket värde som skall användas.

/ssek:SSEK/ssek:ReceiverId

ReceiverId anger mottagarens identitet. Hur innehållet i ReceiverId skall tolkas styrs av ssek:Type-attributet.

/ssek:SSEK/ssek:ReceiverId /@ssek:Type='CN'

Attributet Type är ett valfritt attribut som definierar typ av innehåll i ReceiverId-elementet. Type kan ha någon av värdena CN, DN, ORGNR och APP. Anges inget värde så antas den ha grundvärdet CN. En tjänsts policy kan styra vilket värde som skall användas.

/ssek:SSEK/ssek:TxId

TxId är ett valfritt element som identifierar det meddelandeflöde meddelandet ingår i. Detta sker med en unik identifierare, UUID. Om TxId skall användas styrs av en tjänsts policy.

/ssek:SSEK/@{any}

SSEK-elementet tillåter andra attribut.

3.1.1 SenderId och ReceiverId

Informationsutbyte med SSEK sker i första hand mellan organisationer, vilka även utgör de huvudsakliga adresserbara enheterna i SSEK. Organisationer identifieras och adresseras enligt följande tabell.

Typ	Betyder	Beskrivning
CN	Common Name	Referens till ett certifikat som identifierar organisationen.
DN	Distinguished Name	Referens till ett certifikat som identifierar organisationen.
ORGNR	Organisationsnummer	En organisations organisationsnummer.
APP	Applikation	Används typiskt internt i en organisation för att identifiera specifika applikationer eller system. Bör ej användas mellan organisationer.

Avsändare och mottagare kan identifieras på flera sätt, men det som rekommenderas i det fall man vill uppnå största säkerhet är att identifiera avsändare och mottagare som Distinguished Names (DN), där DN avser det X509-certifikat som används för digital signering av meddelanden. Därmed relateras den identitet som angivits som avsändare (SenderId) med det certifikat som används för att digitalt signera meddelandet.

TX003 Används digital signering så SKALL SenderId och ReceiverId anges som Common Name (CN) eller Distinguished Name (DN) .

TX004 Används digital signering så SKALL SenderId och ReceiverId ha samma Common Name (CN) eller Distinguished Name (DN) som de certifikat som används för digital signering av meddelande respektive svarsmeddelande.

TX005 Om CN används utan signering SKALL CN motsvara det CN som skulle ha specificerats i ett tänkt certifikat för organisationen.

3.1.2 TxId

TxId identifierar ett meddelandeflöde (se kapitel 4) mellan två eller flera parter. Ett meddelandeflöde är en serie av meddelanden som är relaterade. Syntaktiskt är ett TxId en unik identifierare [UUID]. Användning av TxId för en tjänst styrs av dess policy.

TX006 TxId SKALL användas för att identifiera meddelanden när detta behov finns.

TxId skall alltid skapas enligt vedertagna algoritmer för generering av UUID, vilket garanterar att nya TxId inte sammanfaller med befintliga.

TX007 TxId SKALL genereras enligt standardiserad algoritm [UUID].

Under vissa förutsättningar kan TxId återanvändas.

TX008 Om ett antal meddelanden hör ihop i ett meddelandeflöde så SKALL ett TxId användas för att identifiera samtliga meddelanden.

Om ett meddelande inte hör till det meddelandeflöde det identifierar så skall felmeddelande returneras från mottagaren.

TX009 Om ett meddelande inte hör till det pågående meddelandeflöde som dess TxId anger eller inte startar ett nytt meddelandeflöde så SKALL mottagaren returnera soapfault med felkod ssek:IncorrectContext.

Om ett meddelande inte bearbetats av mottagaren så kan avsändaren under vissa förutsättningar skicka samma meddelande på nytt, med samma TxId. Denna omsändning är tillåten förutsatt att ett soapfault mottagits av avsändaren och felkod inte är ssek:Timeout eller att omsändning vid timeout är tillåten enligt policy.

TX010 Omsändning av meddelanden KAN ske när soapfault mottagits och felkod inte är ssek:Timeout.

TX011 Omsändning av meddelanden vid uteblivet svar eller felkod ssek:Timeout KAN ske om policy för tjänst anger ssekp:ResendAllowedOnTimeout.

TX012 När soapfault returneras SKALL producenten backa ur alla transaktioner.

SSEK 2.0 tillåter alltså omsändning under vissa förutsättningar. En konsekvens för producenter är att när soapfault returneras (som inte har felkod ssek:Timeout) så får inga transaktioner ha genomförts utan att samtliga pågående transaktioner SKALL backas ur. För tjänster som inte är uppdaterande kan omsändning vid timeout även tillåtas med ssekp:ResendAllowedOnTimeout.

SSEK lägger inte något ansvar på mottagare för att kontrollera att inte Txid återanvänds efter det att ett meddelandeflöde avslutats.

3.2 Standardiserat felmeddelande

Samtliga fel i ett synkront meddelandeflöde, tekniska och applikationsspecifika fel, returneras som soapfault enligt [BP11] och [SOAP].

F001 Samtliga fel som identifieras hos mottagaren SKALL returneras till avsändaren som soapfault.

F002 Om soapfault returneras till avsändaren så SKALL INTE bestående förändringar hos mottagare genomföras, inga transaktioner får genomföras. Kan inte mottagaren garantera detta så skall ssek:Timeout returneras avsändaren.

För applikationsfel kan domänspecifika felkoder skapas. Dessa skall definieras i XML Schema och vara av datatypen QName.

F003 Applikationsspecifika felkoder SKALL vara av typen QName.

I vissa fall kan mottagare behöva kommuniceras ytterligare felinformation. För detta ändamål finns strukturen FaultData.

F004 FaultData KAN infogas i soapfault under detail-elementet för att detaljerat beskriva felsituationer.

Förenklad beskrivning av FaultData. För exakt beskrivning se bifogat schema.

```
<ssek:FaultData>
  <ssek:FaultingMessage>...</ssek:FaultingMessage>?
  <ssek:TxId>...</ssek:TxId>?
  <ssek:FaultItems>?
    <ssek:FaultItem>
      <ssek:Code>...</ssek:Code>
      <ssek:Description>...</ssek:Description>?
      <ssek:Location>...</ssek:Location>?
      <ssek:System>...</ssek:System>?
    </ssek:FaultItem>*
  </ssek:FaultItems>
  {any}*
</ssek:FaultData>
```

/ssek:FaultData

Detta element kan placeras i soapfault under detail-elementet och användas för att beskriva fel mer uttömmande än vad som är möjligt i faultcode- och faultstring-elementen i soapfault.

/ssek:FaultData/ssek:FaultingMessage
Om frågemeddelande innehåller fel kan detta element användas för att returnera det felaktiga dokumentet till avsändaren. Meddelandet skall infogas som textdata.

/ssek:FaultData/ssek:TxId
Det valfria elementet TxId kan användas för att returnera eventuellt TxId från frågemeddelande.

/ssek:FaultData/ssek:FaultingItems/ssek:FaultingItem
Elementet FaultingItem kan förekomma flera gånger för att beskriva fel i inkommande meddelande.

/ssek:FaultData/ssek:FaultingItems/ssek:FaultingItem/ssek:Code
Elementet skall innehålla en felkod för det specifika felet.

/ssek:FaultData/ssek:FaultingItems/ssek:FaultingItem/ssek:Description
Elementet används för att beskriva det specifika felet.

/ssek:FaultData/ssek:FaultingItems/ssek:FaultingItem/ssek:Location
Valfritt element som kan användas för att identifiera felande element i FaultingMessage med xpath eller på annat sätt.

/ssek:FaultData/ssek:FaultingItems/ssek:FaultingItem/ssek:System
Valfritt element för att specificera felande system.

/ssek:FaultData/{any}
Finns mer information som inte täcks in av övriga element så kan egendefinierade element infogas.

3.2.1 Felkoder

Tabellen nedan definierar felmeddelanden specifika för SSEK.

Felkod	**Beskrivning**
ssek:AsynchMethodUnsupported	Vald asynkron metod stöds inte av tjänst
ssek:ContentInvalid	Meddelandets innehåll är felaktigt.
ssek:IncorrectContext	Ett meddelande som inte hör till det pågående meddelandeflödet eller inte startar ett nytt meddelandeflöde har mottagits. Se TX009.
ssek:MessageNotProcessed	Meddelandet har inte bearbetats på grund av fel hos mottagaren.
ssek:NoResultAvailable	Inget resultat finns tillgängligt för hämtning.
ssek:ReceiverIdUnknown	Mottagarens identitet är okänd.
ssek:SenderIdUnknown	Avsändarens identitet är okänd.
ssek:Timeout	Mottagaren har hamnat i ett instabilt läge. Omsändning är inte tillåten om ej policy tillåter det. Kontakta mottagaren.
ssek:TxIdInvalid	Ej syntaktiskt korrekt TxId.
ssek:TxIdMissing	TxId saknas.
ssek:TxIdNotAllowed	TxId ej tillåtet för denna tjänst.
ssek:TxIdUnknown	Okänt TxId.
ssek:WebServiceUnavailable	Tjänsten är inte tillgänglig.
ssek:WebServiceUnsupported	Tjänsten stöds inte av mottagaren.

3.2.2 Felhantering vid asynkrona meddelandeflöden med leverans

Vid asynkrona meddelandeflöden kan inte soapfault användas för att beskriva fel under den bearbetande fasen då producenten anropar konsumentens tjänst. I dessa fall bör det applikationsspecifika meddelande som skickas från producenten till konsumenten innehålla information om eventuella fel.

3.3 Standardiserat kvitto

SSEK definierar ett standardmeddelande för att bekräfta mottagandet av ett meddelande. Det fyller sin funktion i första hand vid asynkrona meddelandeflöden.

K001 För att bekräfta mottagande av meddelanden SKALL standardiserat kvitto användas.

Förenklad beskrivning av standardkvitto. För exakt beskrivning se bifogat schema.

```
<ssek:Receipt {any}?>
  <ssek:ResponseCode>OK</ssek:ResponseCode>
  <ssek:ResponseMessage>...</ssek:ResponseMessage>?
  <ssek:RequestSignatureValue>...</ssek: RequestSignatureValue>?
  {any}*
</ssek:Receipt>
```

/ssek:Receipt/ssek:ResponseCode

Det standardiserade kvittot skall användas för att kvittera mottagande av frågemeddelande. Om fel uppstår skall istället soapfault användas. Av den anledningen kan elementet ResponseCode endast innehålla OK.

/ssek:Receipt/ssek:ResponseMessage

Det valfria elementet ResponseMessage kan används för mer detaljerad mottagandeinformation.

/ssek:Receipt/ssek:RequestSignatureValue

Om frågemeddelande är signerat så skall RequestSignatureValue innehålla signaturen för frågemeddelanden, med undantag för när [WSS11] används fullt ut då signaturen från frågemeddelanden returneras i Security-headern. Även svarsmeddelandet, dvs Receipt, skall vara signerat. På så vis skapas en teknisk beviskedja från det signerade frågemeddelandet till det signerade kvittot.

K002 Om frågemeddelandet är signerat så SKALL standardkvittot innehålla dess signatur förutsatt att policy-elementet ssekp:RequestSignatureHandling har värdet Receipt eller ReceiptAndSignatureConfirmation för tjänsten eller ingen policy finns angiven.

/ssek:Receipt/{any}

Finns behov som inte kan tillfredsställas av övriga element i det standardiserade kvittot så finns möjligheten att addera egendefinerad information.

3.4 Hantering av bilagor

Bilagor utgörs av icke XML-baserad information som efter BASE64-kodning infogas i XML-dokument. En nackdel med BASE64-kodning är att förfarandet använder extra dator- och nätverksresurser. Lösningen på problemet heter XML-binary Optimized Packaging (XOP). Med XOP serialiserar XML-dokumentet inklusive BASE64-kodade bilagor till ett MIME-dokument, där bilagor lagras i sitt originalformat i MIME-delar (MIME-parts).

Teoretiskt innebär det en serialisering av dokumentets XML-infoset. I praktiken betyder det att bilagor kan hanteras separat utan BASE64-kodning och infogas direkt i XML-dokument. Vi får alltså ett XML-dokument som består av flera MIME-delar. För detaljer se [XOP], [MTOM] och [SOAP11MTOM]. Notera att för meddelanden som signeras kan i praktiken inte BASE64-kodningen undvikas.

A001 Bilagor SKALL kunna hanteras både BASE64-kodade och optimerade enligt [XOP], [MTOM] och [SOAP11MTOM].

4 Meddelandeflöden

Med meddelandeflöde menas överföringen av ett eller flera meddelanden mellan konsument och producent i syfte att genomföra informationsutbyte.

SSEK definierar fyra grundläggande typer av meddelandeflöden:

- Enkelriktat meddelandeflöde
- Synkront meddelandeflöde
- Asynkront meddelandeflöde med leverans (push)
- Asynkront meddelandeflöde med hämtning (pull)

Dessa kan i sin tur kombineras till mer komplicerade meddelandeflöden.

4.1 Enkelriktat meddelandeflöde

Ett enkelriktat meddelandeflöde består av ett meddelande som skickas från konsument till producent. Producenten returnerar inte något meddelande, utan avslutar flödet genom att i normalfallet returnera HTTP returkod 200 [SOAP].

M001 Producenter KAN stödja enkelriktade meddelandeflöden. Konsumenter SKALL stödja enkelriktade meddelandeflöden.

4.2 Synkront meddelandeflöde

Ett synkront meddelandeflöde består av ett meddelande som skickas från konsument till producent, varvid producenten synkront returnerar ett svarsmeddelande. Med synkront menas att svaret returneras i HTTP-svaret.

M002 Producenter och konsumenter SKALL stödja synkrona meddelandeflöden.

4.3 Asynkront meddelandeflöde med leverans (push)

Asynkront meddelandeflöde med leverans (AML) byggs upp av två synkrona meddelandeflöden och två tjänster. Det inleds med ett synkront meddelandeflöde från konsumenten till producenten. Producenten levererar ett kvitto på mottaget meddelande till konsumenten och påbörjar bearbetning av meddelandet. Efter avslutad bearbetning blir rollerna ombytta, det vill säga producenten initierar ett synkront meddelandeflöde till konsumenten och levererar resultatet av bearbetningen. Både producenten och konsumenten måste i detta fall tillhandahålla tjänster.

M003 Om en tjänst stöder AML så SKALL detta framgå av dess policy.

M004 Om en konsument avser starta ett AML så SKALL detta anges med attributet AsynchMethod på ssek:SSEK satt till AsynchPush.

4.4 Asynkront meddelandeflöde med hämtning (pull)

Asynkront meddelandeflöde med hämtning (AMH) består även det av två synkrona meddelandeflöden. Meddelandeflödet inleds med ett synkront meddelandeflöde från konsumenten till producenten. Producenten levererar ett kvitto på mottaget meddelande till konsumenten och påbörjar bearbetning av meddelandet.

Efter en tidsperiod inleder konsumenten ett synkront meddelandeflöde där resultatet från bearbetning efterfrågas. Finns bearbetningsresultat tillgängligt så returneras det till konsumenten. I annat fall returneras felmeddelande med felkod ssek:NoResultAvailable.

Konsumenten efterfrågar resultatet med meddelandet PullMessage.

```
<ssek:PullMessage {any}?>
  {any}*
</ssek:PullMessage>
```

/ssek:PullMessage

Elementen PullMessage skall infogas som meddelandeinnehåll och markerar en förfrågan om bearbetningsresultatet i ett AMH.

/ssek:PullMessage/{any}
För applikationsspecifika ändamål kan egendefinierade element infogas.

/ssek:PullMessage/@{any}
För applikationsspecifika ändamål kan egendefinierade attribut infogas.

PullMessage innehåller i sig ingen information om vilken bearbetning som efterfrågas utan det identifieras implicit med meddelandeflödets TxId.

Fördelen med AMH förfarande jämfört med AML är att konsumenten förenklas avsevärt då den inte behöver tillhandahålla någon tjänst utan kan uteslutande agera avsändare av meddelanden.

M005 Om en tjänst stöder AMH SKALL detta framgå av dess policy.

M006 Om en konsument startar ett AMH SKALL detta anges med attributet asynchMethod på ssek:SSEK satt till AsynchPull.

M007 Konsumenten SKALL fråga efter bearbetningsresultat med meddelandet ssek:PullMessage.

M008 Felkoden ssek:NoResultAvailable SKALL returneras då inget resultat finns tillgängligt i ett AMH.

Det skall noteras att producenten med AMH inte erhåller något kvitto på att konsumenten mottagit bearbetningssvar.

4.5 Felsituationer kring meddelandeflöden

Felsituationer som är relaterade till meddelandeflöden.

M009 Om konsumenten valt en typ av meddelandeflöde som inte stöds av tjänsten SKALL producenten returnera soapfault med felkod ssek:AsynchMethodUnsupported.

5 Säkerhet

Avseende säkerhet vid kommunikation över Internet har följande affärsbehov identifierats:

- Att information skyddas mot att bli läst eller manipulerad av obehöriga.
- Att information säkert kan härledas till avsändaren.

För att möta affärsbehoven avseende säkerhet uppfyller SSEK de säkerhetsaspekter som följande tabell beskriver. För detta ändamål använder SSEK säkerhetsmodellen PKI (Public Key Infrastructure) [PKI].

Aspekt	**Förklaring**	**Stöd i SSEK**
Sekretess	Meddelanden kan under transport inte läsas av andra än mottagaren.	SSL
Autentisering av mottagare	Mottagarens identitet är styrkt för avsändaren.	SSL med servercertifikat
Autentisering av avsändaren	Avsändarens identitet är styrkt för mottagaren.	SSL med (klient) organisationscertifikat
Integritet, riktighet	Meddelanden har inte förändrats under transport från avsändare till mottagare.	Signering med stämpelcertifikat
Oavvislighet	Avsändaren kan inte förneka att innehållet i ett meddelande skapats av dem.	Signering med stämpelcertifikat

SSEK delar upp säkerhetsaspekterna i transportsäkerhet (TransportLevelSecurity) och meddelandesäkerhet (MessageLevelSecurity).

5.1 Transportsäkerhet

SSEK specificerar två nivåer av transportsäkerhet:

- Sekretess och autentisering av mottagaren (SSL)
- Sekretess och autentisering av avsändare och mottagare (SSLWithClientCertificate)

SSEK kräver att minst transportsäkerhetsnivån SSL används. Transportsäkerhetsnivån SSL innebär att informationen skyddas mot att obehöriga kan läsa den samt att mottagaren är identifierad. Transportsäkerhetsnivån SSLWithClientCertificate innebär att även avsändaren identifieras. Det kan exempelvis användas vid publicering av tjänst som behöver skyddas mot åtkomst av obehöriga.

S001 SSL med serverautentisering (SSL) SKALL användas.

S002 SSL med klientautentisering (SSLWithClientCertificate) KAN användas.

S003 En tjänsts transportsäkerhetsnivån SKALL anges i dess policy.

5.2 Meddelandesäkerhet

SSEK definierar två nivåer av meddelandesäkerhet:

- Ingen meddelandesäkerhet (None)
- Oavvislighet och integritet (Signature)

Huruvida meddelandesäkerhet ska användas eller inte är valbart. Finns behov av att kunna kontrollera och bevisa att viss information skapats av en viss avsändare används alltså meddelandesäkerhetsnivå Signature.

S004 Signering (Signature) av meddelanden KAN användas.

S005 En tjänsts meddelandesäkerhetsnivån SKALL anges i dess policy.

S006 Om signering används SKALL samtliga meddelande i det synkrona eller asynkrona meddelandeflödet signeras.

5.3 X509-certifikat

Ett X509-certifikat [X509V3] är ett dokument innehållande en relation mellan en publik nyckel och ett antal attribut som typiskt beskriver en organisation. Genom att certifikat skapas och signeras av en betrodd organisation (Certificate Authority) som garanterar att certifikatet korrekt beskriver organisationen så kan certifikatet användas för att identifiera organisationen. När en organisation använder sin privata nyckel, som hör ihop med den publika i certifikatet, för att exempelvis signera ett meddelande så kan denna signatur sedan användas tillsammans med certifikatet för att visa att organisationen signerat meddelandet.

Certifikat används i SSEK för att identifiera organisationer och webbserver. Certifikat skall ges ut av en CA (Certificate Authority) som garanterar dess riktighet. Ett certifikat kan revokeras, det vill säga återkallas, om det av någon orsak inte längre unikt identifierar en organisation eller webbserver.

Följande riktlinjer gäller för certifikat som används inom SSEK.

S006 De certifikat som används vid kommunikation mellan två parter SKALL vara utgivna av en CA (Certificate Authority) samt vara av en typ som båda parter godkänt.

S007 Den part som använder ett certifikat BÖR utföra revokeringskontroll genom att hämta den CRL (Certificate Revocation List) som certifikatets CA ger ut, alternativt utföra revokeringskontroll med OCSP (Online Certificate Status Protocol) [OCSP], enligt riktlinjer från CA. Om revokeringskontroll utförs med hjälp av CRL alternativt OCSP SKALL kontroll utföras på att certifikatet inte har revokerats varje gång certifikatet används.

S008 Om revokeringskontroll enligt S007 inte utförs SKALL manuell revokering utföras. Vid manuell revokering kontaktar den part som vill göra ett certifikat otillåtet så snart som möjligt 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.

5.4 Signering av meddelanden

Signering av meddelande vilar på underliggande specifikationer från IETF, W3C, OASIS samt WS-I. IETF och W3C lägger grunden för certifikathantering, kanonisering och signering av XML, OASIS definierar tillämpningen av detta på SOAP -meddelanden och WS-I i sin tur begränsar möjligheterna för att öka interoperabiliteten mellan samverkande system.

OASIS har utvecklat [WSS10] och en uppdatering till denna [WSS11]. [WSS11] inför ett antal nya funktioner, men är bakåtkompatibel med [WSS10]. För SSEK är speciellt signaturkvitteringen i [WSS11] en viktig funktion som i SSEK 1.1 implementerades med standardkvittot. Av praktiska skäl kräver SSEK 2.0 stöd för [WSS10] men låter för framtida bruk stöd för signaturkvitteringen i [WSS11] styras via policy.

SIG01 Signering av SSEK-meddelanden SKALL ske enligt [WSS10], [X509CTP], [C14], [XMLDSIG] samt [BSP10] med undantag definierade i denna specifikation.

SIG02 wsu:Timestamp SKALL bifogas enligt [WSS10] och [BSP10]. Både wsu:Created och wsu:Expired SKALL anges.

SIG03 ssek:SSEK, soapbody, wsu:Timestamp och, i förekommande fall för svarsmeddelanden, wsse11:SignatureConfirmation SKALL signeras.

SIG04 Det X509-certifikat som använts för att signera meddelandet SKALL bifogas i X509v3-format BASE64-kodat i ett wsse:BinarySecurityToken-element.

SIG05 Det X509-certifikat som signerat meddelandet SKALL refereras direkt via ett wsse:SecurityTokenReference-element.

SSEK gör vissa begränsningar kring användningen av specifikationerna.

SIG06 De delar av ett meddelande som signeras SKALL identifieras med wsu:Id-attribut och refereras med en förkortad XPointer (shorthand XPointer).

SIG07 De delar av ett meddelande som transformerats/signeras SKALL transformeras enligt "http://www.w3.org/2001/10/xml-exc-c14n#" [C14N].

SIG08 Signaturmetod "http://www.w3.org/2000/09/xmldsig#rsa-sha1" SKALL användas. [XMLDSIG].

SIG09 Kryptering av meddelanden SKALL INTE användas.

6 Metadata

Metadata är information om meddelande. Två typer av metadata används:

- Information om vad som skall kommuniceras [XMLSCHEMA], [WSDL].
- Information om hur det skall kommuniceras [WS-POLICY].

XML [XML] och XML Schema [XMLSCHEMA1] [XMLSCHEMA2] utgör grunden för hur meddelanden beskrivs i SSEK.

MD01 XML Schema SKALL användas för att beskriva meddelandeinnehåll.

WSDL 1.1 beskriver vilka meddelanden som en tjänst hanterar.

MD02 Tjänster SKALL beskrivas i WSDL 1.1 [WSDL].

MD03 Policies SKALL infogas i WSDL-dokument.

MD04 Conformance Claim för Basic Profile 1.1 och Basic Security Profile 1.0 SKALL infogas i WSDL-dokument [CCAM].

Observera att SSEK är baserat på Basic Profile 1.1, vilket betyder att WSDL skall användas enligt dess riktlinjer.

6.1 SSEK Policy Language

WS-Policy [WSPOLICY] [WSPOLICYATT] definierar ett ramverk för hur beteenden hos tjänster kan beskrivas. Med beteenden menas exempelvis om meddelanden skall vara signerade eller om SSL skall användas. Med policies kan följande aspekter av SSEK definieras för en tjänst:

- Om TxId skall användas eller ej.
- Hur avsändare och mottagare skall identifieras i meddelanden.
- Om omsändning tillåts generellt eller ej.
- Vilka typer asynkrona meddelandeflöden som stöds.
- Vilken transportsäkerhet som skall användas.

- Vilken meddelandesäkerhet som skall användas.
- Hur frågemeddelandets signatur skall returneras till avsändaren.

Förenklad beskrivning av SSEK Policy Language (http://schemas.ssek.org/ssek/2006-05-10/policy). För mer detaljer se bifogat schema.

```
<ssekp:ServiceAssertions>
  <ssekp:IdType>CN|DN|ORGNR|APP<ssekp:IdType>?
  <ssekp:UseTxId/>?
  <ssekp:TransportLevelSecurity>SSL|SSLWithClientCertificate</ssekp:TransportLevelSecurity>
  <ssekp:MessageLevelSecurity>None|Signature</ssekp:MessageLevelSecurity>
  <ssekp:RequestSignatureHandling>
        Receipt|SignatureConfirmation|ReceiptAndSignatureConfirmation
  </ssekp:RequestSignatureHandling>?
<ssekp:ServiceAssertions>
<ssekp:OperationAssertions>
  <ssekp:ResendAllowedOnTimeout/>?
  <ssekp:SupportsAsynchPull/>?
  <ssekp:SupportsAsynchPush/>?
</ssekp:OperationAssertions>
```

/ssekp:ServiceAssertion/ssekp:IdType
: Elementet IdType anger med vilken typ SenderId och ReceiverId skall anges för tjänsten. Se SSEK-schemat för innehållsbeskrivning.

/ssekp:ServiceAssertion/ssekp:UseTxId
: Elementet UseTxId anger att TxId skall användas vid anrop av tjänsten.

/ssekp:ServiceAssertion/ssekp:TransportLevelSecurity
: Elementet TransportLevelSecurity anger vilken säkerhet som skall användas på transportnivå för tjänsten. Kan antingen vara SSL eller SSLWithClientCertificate.

/ssekp:ServiceAssertion/ssekp:MessageLevelSecurity
: Elementet MessageLevelSecurity anger vilken säkerhets som skall användas på meddelandenivå för tjänsten. Kan antingen vara None eller Signature.

/ssekp:ServiceAssertion/ssekp:RequestSignatureHandling
: Elementet RequestSignatureHandling anger hur signaturen i frågemeddelanden skall returneras till avsändaren. Kan antingen vara Receipt, SignatureConfirmation eller ReceiptAndSignatureConfirmation. Receipt betyder att standardkvitto returneras med frågemeddelandens signatur. SignatureConfirmation betyder att förfarande definierat i [WSS11] används för att returnera frågemeddelandens signaturer och ReceiptAndSignatureConfirmation betyder att båda sätten används.

/ssekp:OperationAssertion/ssekp:ResendAllowedOnTimeout
: Elementet ResendAllowedOnTimeout anger att omsändning av meddelande tillåts utan restriktioner för operationen.

/ssekp:OperationAssertion/ssekp:SupportsAsynchPull
: Elementet SupportsAsynchPull anger att operationen kan inleda ett asynkront meddelandeflöde med hämtning.

/ssekp:OperationAssertion/ssekp:SupportsAsynchPush
: Elementet SupportsAsynchPush anger att operationen kan inleda ett asynkront meddelandeflöde med leverans.

Enligt den terminologi som presenteras i [WSPOLICYATT] så skall SSEK policy placeras ut i WSDL-filen för en tjänst enligt följande:

MD05	ssekp:ServiceAssertion SKALL refereras från wsdl:binding.
MD06	ssekp:OperationAssertion SKALL refereras från wsdl:binding/wsdl:operation.

7 Affärsavtal med SSEK 2.0 som komponent

Specifikationen SSEK 1.1 togs fram för att möjliggöra affärskommunikation av affärskritiska och känsliga uppgifter. Kraven på specifikationen ställdes inför en praktisk, affärsmässig utmaning där kommunikation av affärsuppgifter som fram tills dess utförts på undertecknade pappersdokument nu behövde ersättas med elektronisk kommunikation. Kraven ställdes från juridiskt, säkerhetsmässigt och affärsmässigt håll. Den information som skickades behövde skyddas mot insyn av obehöriga samt, på ett säkert sätt, kunna kopplas till den avsändande parten. Att SSEK 1.1 nu används i praktisk kommunikation av känslig, affärskritisk information är ett bevis för att specifikationen lever upp till de krav som ställts.

SSEK 2.0 är en förbättrad specifikation som genom utökad funktionalitet och modernisering avseende de standards som SSEK 2.0 baseras på ger ännu bättre möjligheter till effektiv elektronisk affärskommunikation.

Att använda SSEK 2.0 möjliggör säker användning av tjänster. Det krävs dock att ett antal punkter, utöver det faktum att
SSEK 2.0 ska användas, överenskommes i exempelvis affärsavtal. Överenskommelsen av dessa punkter, tillsammans med att SSEK 2.0 ska användas, bildar bas för säker elektronisk affärskommunikation.

Det som behöver överenskommas kan delas upp i två områden, sådant som SSEK 2.0 specificerar men där olika val ges samt sådant som ligger utanför specifikationen. För att säkert kunna omsätta avtalade tjänster, publicerade enligt SSEK 2.0, i praktiskt tjänsteutnyttjande så BÖR därmed följande punkter överenskommas.

Punkter som behöver definieras och där val ges av SSEK 2.0:

- Kommunicerande parters SenderId.
- Kommunicerande parters ReceiverId.
- URL för producent.
- I förekommande fall URL för konsument.
- Vilka nivåer avseende transport- och meddelandesäkerhet som ska användas.
- Vilken modell för revokering som ska användas.

Punkter som behöver definieras och som inte specificeras av SSEK 2.0:

- Vilka CA (Certificate Authority) som godkänns för certifikat avsedda för signering respektive autentisering.
- Vilken typ av certifikat som ska tillåtas för signering respektive autentisering från överenskommen CA.
- Vilken betydelse beståndsdelarna i de tjänster som tillhandahålls har.

8 Skillnader från SSEK 1.1

Följande förändringar har i korthet skett mellan SSEK 1.1 och SSEK 2.0:

- För signering av meddelanden används OASIS WS-Security: Soap message Security 1.0/1.1
- Hantering av TxId har förenklats.
- Timestamp i TxHeader utgår i SSEK-headern och ersätts av timestamp i WS-Security.
- Omsändning av meddelande tillåts under vissa förutsättningar.
- Standardiserad felhantering.
- Alla meddelandestrukturen inom SSEK använder samma namespace URI, http://schemas.ssek.org/ssek/2006-03-01/
- Stöd för asynkront meddelandeflöde med hämtning.
- Stöd för Policies.
- Anpassning mot WS-I Basic Profile 1.1 och Basic Security Profile 1.0.

9 Referenser

[BP11]	WS-I Basic Profile Version 1.1, 2004-08-24
[BP11ERR]	WS-I Basic Profile Version 1.1 Errata, Board Approval Draft, Rev: 1.8, 2005/10/25
[BSP10]	WS-I Basic Security Profile Version 1.0, Working Group Draft, 2006-01-20
[C14N]	Exclusive XML Canonicalization, Version 1.0, W3C Recommendation 18 July 2002 (http://www.w3.org/TR/2002/REC-xml-exc-c14n-20020718/)
[CCAM]	WS-I Conformance Claim Attachment Mechanism Version 1.0, Final Material 2004-11-15
[MIME]	MIME Media Types, Internet Assigned Numbers Authority
[MTOM]	SOAP Message Transmission Optimization Mechanism, W3C Recommendation 25 January 2005
[OCSP]	Online Certificate Status Protocol, http://www.ietf.org/rfc/rfc2560.txt
[PKI]	Public-Key Infrastructure (X.509) (pkix), http://www.ietf.org/html.charters/pkix-charter.html
[SOAP]	Simple Object Access Protocol (SOAP) 1.1, W3C Note, 08 May 2000
[SOAP11MTOM]	SOAP 1.1 Binding for MTOM 1.0, March 2, 2006
[UUID]	RFC4122: A Universally Unique IDentifier (UUID) URN Namespace, http://www.ietf.org/rfc/rfc4122.txt
[WSA]	Web Services Addressing 1.0 - Core, W3C Candidate Recommendation 17 August 2005, http://www.w3.org/TR/2005/CR-ws-addr-core-20050817/
[WSASB]	Web Services Addressing 1.0 - SOAP Binding, W3C Candidate Recommendation 17 August 2005, http://www.w3.org/TR/2005/CR-ws-addr-soap-20050817/
[WSDL]	Web Services Description Language (WSDL) 1.1, W3C Note, 15 March 2001
[WSPOLICY]	Web Services Policy Framework (WS-Policy), September 2004
[WSPOLICYATT]	Web Services Policy Attachment (WS-PolicyAttachment), September 2004
[WSS10]	Web Services Security: SOAP Message Security 1.0, (WS-Security 2004), OASIS Standard 200401, March 2004
[WSS11]	Web Services Security: SOAP Message Security 1.1, (WS-Security 2004), OASIS Standard Specification, 1 February 2006
[X509CTP]	Web Services Security X.509 Certificate Token Profile 1.1, OASIS Standard Specification, 1 February 2006
[X509V3]	Internet X.509 Public Key Infrastructure, Certificate and CRL Profile, http://www.ietf.org/rfc/rfc2459.txt
[XML]	Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation 04 February 2004. http://www.w3.org/TR/2004/REC-xml-20040204/
[XMLBD]	Assigning Media Types to Binary Data in XML, W3C Working Draft 2 November 2004
[XMLDSIG]	XML-Signature Syntax and Processing, W3C Recommendation, 12 February 2002. http://www.w3.org/TR/xmldsig-core/.
[XMLSCHEMA1]	XML Schema Part 1: Structures Second Edition, W3C Recommendation 28 October 2004. http://www.w3.org/TR/xmlschema-1/
[XMLSCHEMA2]	XML Schema Part 2: Datatypes Second Edition, W3C Recommendation 28 October 2004. http://www.w3.org/TR/xmlschema-2/
[XOP]	XML-binary Optimized Packaging, W3C Recommendation 25 January 2005

Bilaga 1: XML schema för SSEK

```
<xsd:schema targetNamespace="http://schemas.ssek.org/ssek/2006-05-10/"
       elementFormDefault="qualified" attributeFormDefault="qualified"
       xmlns:tns="http://schemas.ssek.org/ssek/2006-05-10/"
       xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:simpleType name="IdType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="APP"/>
      <xsd:enumeration value="CN"/>
      <xsd:enumeration value="DN"/>
      <xsd:enumeration value="ORGNR"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="FaultCode">
    <xsd:restriction base="xsd:QName">
      <xsd:enumeration value="AsynchMethodUnsupported"/>
      <xsd:enumeration value="ContentInvalid"/>
      <xsd:enumeration value="IncorrectContext"/>
      <xsd:enumeration value="MessageNotProcessed"/>
      <xsd:enumeration value="NoResultAvailable"/>
      <xsd:enumeration value="ReceiverIdUnknown"/>
      <xsd:enumeration value="SenderIdUnknown"/>
      <xsd:enumeration value="Timeout"/>
      <xsd:enumeration value="TxIdInvalid"/>
      <xsd:enumeration value="TxIdMissing"/>
      <xsd:enumeration value="TxIdNotAllowed"/>
      <xsd:enumeration value="TxIdUnknown"/>
      <xsd:enumeration value="WebServiceUnavailable"/>
      <xsd:enumeration value="WebServiceUnsupported"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:element name="SSEK">
    <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="512"/>
                  </xsd:restriction>
                </xsd:simpleType>
                <xsd:attribute name="Type" default="CN" type="tns:IdType"/>
              </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="512"/>
                  </xsd:restriction>
                </xsd:simpleType>
                <xsd:attribute name="Type" default="CN" type="tns:IdType"/>
              </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:sequence>
      <xsd:attribute name="AsynchMethod" use="optional">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="AsynchPull"/>
            <xsd:enumeration value="AsynchPush"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="FaultData">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="FaultingMessage" type="xsd:string" minOccurs="0"/>
        <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 minOccurs="0" name="FaultItems">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element maxOccurs="unbounded" name="FaultItem">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="Code" type="xsd:string"/>
                    <xsd:element name="Description" type="xsd:string" minOccurs="0"/>
                    <xsd:element name="Location" type="xsd:string" minOccurs="0"/>
                    <xsd:element name="System" type="xsd:string" minOccurs="0"/>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:any processContents="lax" minOccurs="0" maxOccurs="unbounded" namespace="##other"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Receipt">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="ResponseCode">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="OK"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="ResponseMessage" type="xsd:string" minOccurs="0"/>
        <xsd:element name="RequestSignatureValue" type="xsd:string" minOccurs="0"/>
        <xsd:any processContents="lax" minOccurs="0" maxOccurs="unbounded" namespace="##other"/>
      </xsd:sequence>
      <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="PullMessage">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Bilaga 2: XML schema för SSEK policy language

```
<xsd:schema targetNamespace="http://schemas.ssek.org/ssek/2006-05-10/policy"
       elementFormDefault="qualified" attributeFormDefault="qualified"
       xmlns:tns="http://schemas.ssek.org/ssek/2006-05-10/policy"
       xmlns:xsd="http://www.w3.org/2001/XMLSchema"
       xmlns:ssek="http://schemas.ssek.org/ssek/2006-05-10/">
  <xsd:element name="ServiceAssertion">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="IdType" type="ssek:IdType" minOccurs="0"/>
        <xsd:element minOccurs="0" name="UseTxId"/>
        <xsd:element name="TransportLevelSecurity">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="SSL"/>
              <xsd:enumeration value="SSLWithClientCertificate"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="MessageLevelSecurity">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="None"/>
              <xsd:enumeration value="Signature"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="RequestSignatureHandling" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="Receipt"/>
              <xsd:enumeration value="SignatureConfirmation"/>
              <xsd:enumeration value="ReceiptAndSignatureConfirmation"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="OperationAssertion">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element minOccurs="0" name="ResendAllowedOnTimeout"/>
        <xsd:element minOccurs="0" name="SupportsAsynchPull"/>
        <xsd:element minOccurs="0" name="SupportsAsynchPush"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```