

Guide til krypteret tilslutning

Indholdsfortegnelse

Overblik	2
Certifikater	2
Nuværende certifikater.....	2
Certifikater, der stoles på i produktion:	2
Krypteret forbindelse	3
Baggrundsforklaring	3
Hvordan trækker man et certifikat ud fra serveren	5
Yderligere sikkerhedstrin.....	6
Gensidig validering.....	7
Eksempel:.....	7
Eksempel på opsætning af lokalt certifikat gennem SoapUI	7
Test af service-kald.....	9
TLS V1.2.....	10
WS-Policy	11
Keystore og truststore	13
Opsætning af WSS	13
Screenshot på opsætningen af et SoapUI projekt:	17
Tildele rettigheder i DCS	18
Debugging.....	18
Ofte opståede fejl	18

Overblik

For at koble på PSRM (via NyMFs B2B komponent) så skal følgende trin gennemføres:

1. Opret krypteret forbindelse
2. Tildel system-til-system rettigheder i DCS

Tidskrævende!

NB: Dette er tidskrævende og vanskelig at fejlsøge.

Sørg derfor for at afsætte ekstra tid til den krypterede forbindelse - både mod test og produktion!

Certifikater

Der skal oprettes en krypteret forbindelse i forbindelse med onboarding til PSRM.

For at kunne gøre det, skal fordringshaveren benytte sig af følgende truststores, hvor test_truststore.jks (password= *password*) skal bruges til at starte med på testmiljøerne og prod_truststore.jks skal bruges, når man skal skifte til produktionen senere i forløbet:

Nuværende certifikater

Se og download tilgængelige certifikater på [Gældsstyrelsens hjemmeside](#).

Certifikater, der stoles på i produktion:

Hvis jeres VOCES-certifikat er en af nedenstående, så stoles der automatisk på jeres certifikat, og der skal derfor ikke tilføjes noget til truststoren.

- Alias name: trust2408 oces ca iv (trust2408 oces primary ca)
 - Alias name: trust2408 oces primary ca
 - Alias name: trust2408 oces ca iii (trust2408 oces primary ca)
 - Alias name: trust2408 oces ca ii
 - Alias name: trust2408 oces ca v (trust2408 oces primary ca)
-

Krypteret forbindelse

Baggrundsforklaring

Keystore:

- Indeholder private/public key par for certifikater.
- Benyttes til at signere og validere, dvs. at identificere en afsender.
- Information som er signeret (=krypteret) med den private nøgle kan kun valideres (depryteres) med den tilhørende offentlige (=public) nøgle.
- Man kan eksportere sin public key og dele den med alle, men private key må aldrig sendes ud til andre. Den skal holdes hemmelig.

Truststore:

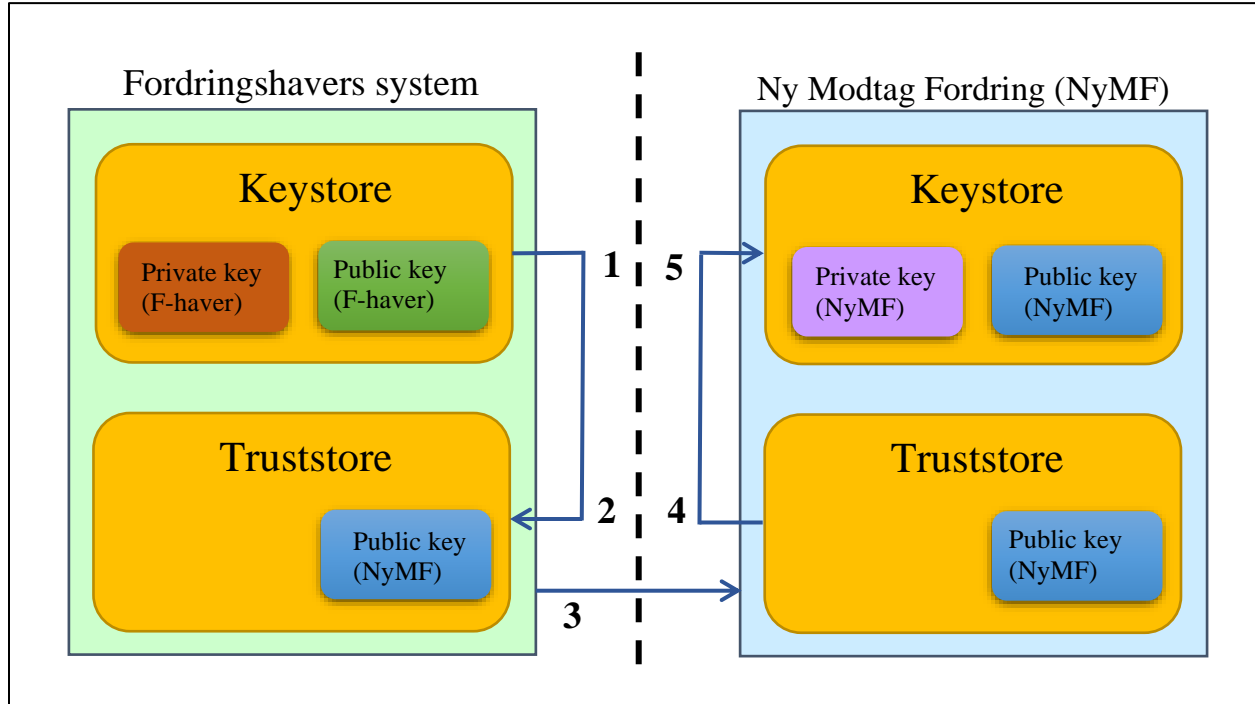
- Indeholder eksterne navne (public keys/certifikater) på dem man stoler på.
- Truststoren bruges til at kryptere med således at beskeden ikke kan læses af andre undervejs.
- Information som er krypteret med den offentlige (=public) nøgle kan kun depryteres med den tilhørende private nøgle.

Man har dermed brug for et VOCES-certifikat, som skal bestilles fra NETS.

I første omgang skal der bruge et test-certifikat, som kan hentes herfra <https://www.nets.eu/dk-da/kundeservice/nemid-tjenesteudbyder/NemID-tjenesteudbyderpakken/Pages/OCES-II-certifikat-eksempler.aspx> (det første gyldige Virksomhedscertifikat på siden downloades - bemærk password= *Test1234*) Dette kan bruges til test, indtil man får udstedt sit eget certifikat.

VOCES-certifikatet bruges sammen med test-Truststore.jks til hhv. at signere og kryptere med.

Følgende tegning illustrerer flowet:



1. Fordringshaver: Bruger egen private key fra Keystore til at signere med.
2. Fordringshaver: Bruger NyMFs public key fra Truststore til at kryptere med.
3. Fordringshaver: Sender fordring.
4. NyMF: Bruger egen private key til at dekryptere med.
5. NyMF: Bruger egen public key (men kun udsteder delen) til at validere med.

Hvordan trækker man et certifikat ud fra serveren

Man kan forbinde og trække certifikatet ved kommandoen her:

Kommando til at trække certifikat

```
openssl s_client -showcerts -connect nymf-b2b-soap-oio-dockXX-secure.inddr.dk:443 -servername nymf-b2b-soap-oio-dockXX-secure.inddr.dk </dev/null 2>/dev/null|openssl x509 -outform PEM >clientTrustCert.pem
```

Man kan også få vist certifikatet sammen med andre egenskaber med følgende kommando:

```
openssl s_client -connect <ip>:<port> -servername <ip>
```

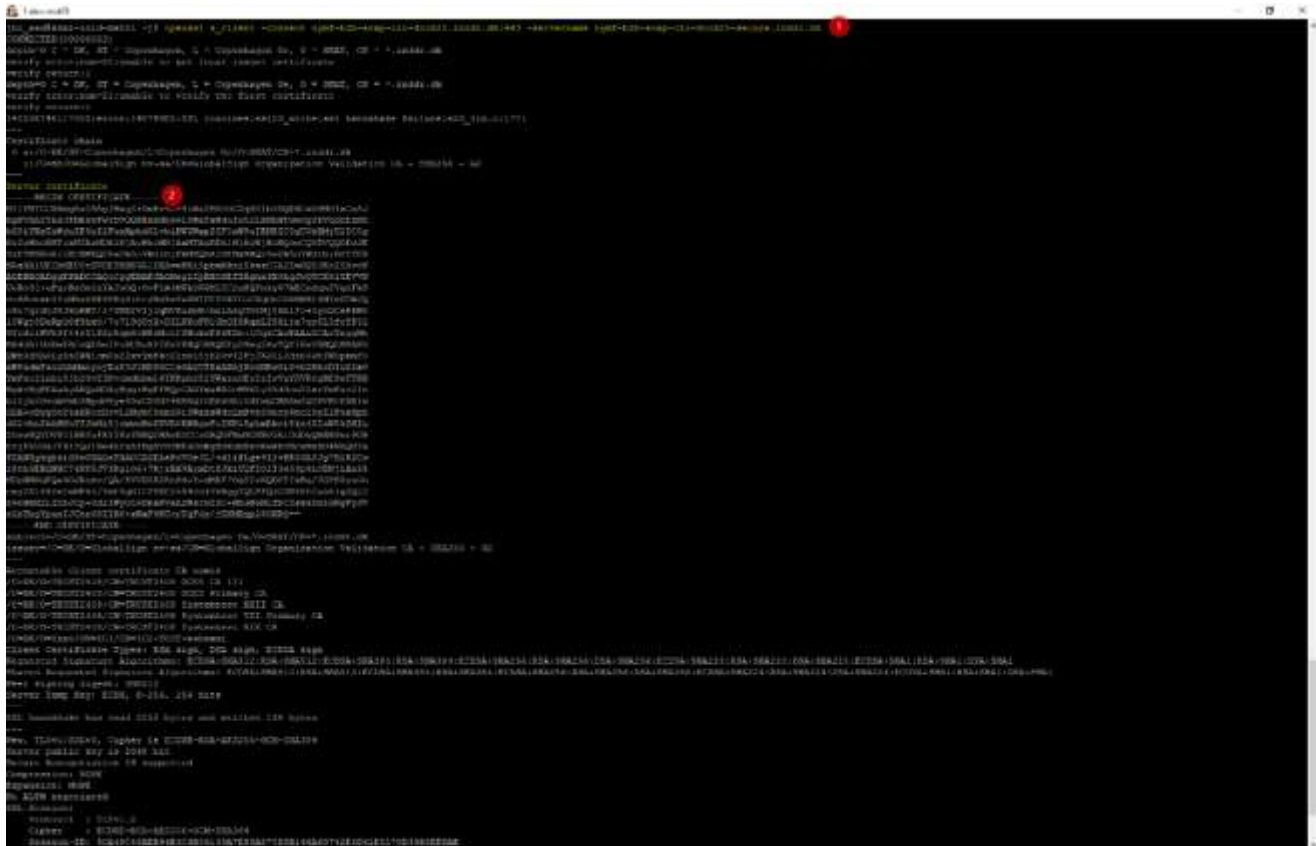
Erstat *<ip>* and *<port>* med url på den server du kommunikerer med. Kommandoen kan køres i Unix eller i Windows gennem fx Cygwin eller Git Bash. Ved forbindelse listes bl.a. serverens test-certifikat.

Eksempel:

Brug altid det docker miljø, I har fået tildelt.

I må ikke sende fordringer ind på andre miljøer end det miljø, der er tildelt til jer.

openssl s_client -connect nymf-b2b-soap-oio-dockXX.inddr.dk:443 -servername nymf-b2b-soap-oio-dockXX-secure.inddr.dk



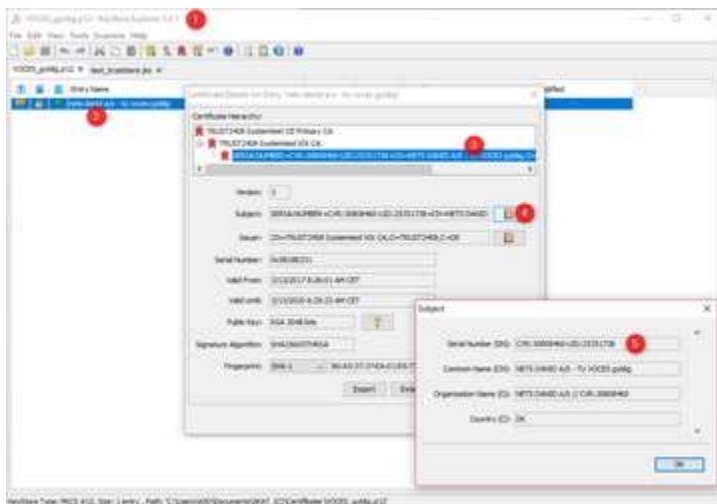
Herefter kan der forbindes til testmiljøet gennem servicen FordringIndberetService. Det kræver et par yderligere sikkerhedstrin, som vi gennemgår nedenfor.

Yderligere sikkerhedstrin

NyMF kræver, at klienter krypterer beskeder. Der skal udveksles certifikater, så NyMF kan tilføje dem til truststore. Klienter krypterer deres trafik med VOCES-certifikat, hvor CVR-nummer i certifikatet skal stemme overens med indberetters CVR-nummer.

Man kan desuden finde sit CVR-nummer på sit VOCES-certifikat ved at følge følgende skridt:

1. Hent og installer en keystore applikation til at håndtere certifikater. Vi bruger selv "keystore explorer" <http://keystore-explorer.org/downloads.html>
2. Åbn dit certifikat og dobbeltklik på det. Her kan du se dit Root-, Intermediate- og User-certifikat:
3. Tryk på User-certifikat (den nederste i hierarkiet).
4. Tryk på notesbogen
5. Aflæs dit CVR- og UID-nummer.



Gensidig validering

For at oprette system-til-system-adgang til PSRM, skal virksomheden have et validt OCES-II certifikat. For alle services tillades VOES (Virksomheds) certifikat. Det anbefales at bestille sit virksomhedscertifikat hos NETS. Indtil man får sit eget certifikat, kan man bruge NETS test-certifikat herfra: <https://www.nets.eu/dk-da/kundeservice/nemid-tjenesteudbyder/NemID-tjenesteudbyderpakken/Pages/OCES-II-certifikat-eksempler.aspx>

Vi benytter certifikaterne listet øverst på siden i prod og test-miljøer.

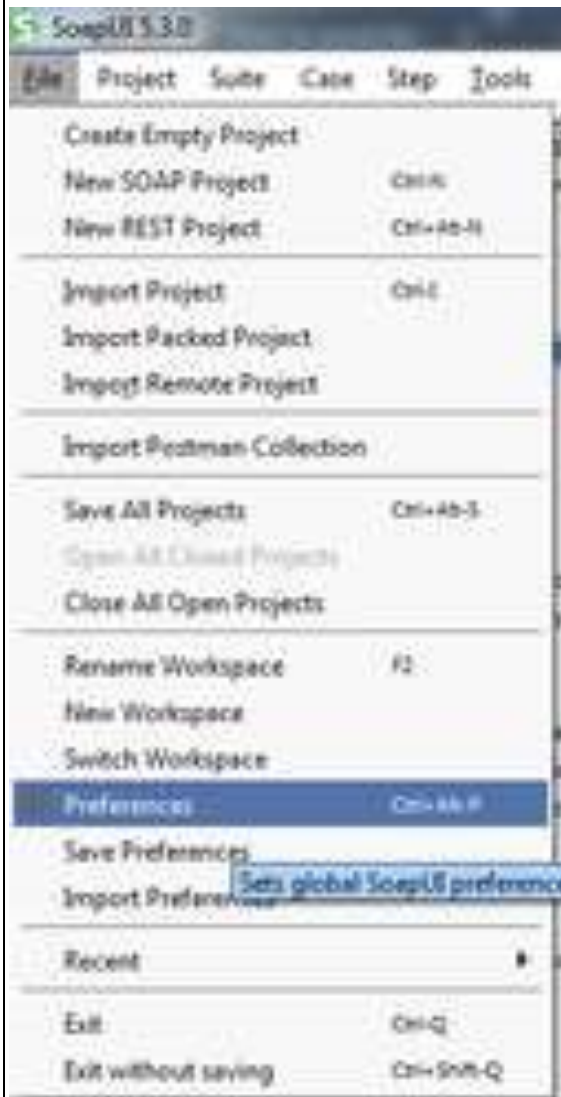
Eksempel:

Lokal certifikat gennem SOAPUI

Eksempel på opsætning af lokalt certifikat gennem SoapUI

Hent det gyldige test certifikat (det første virksomhedscertifikat under <https://www.nets.eu/dk-da/kundeservice/nemid-tjenesteudbyder/NemID-tjenesteudbyderpakken/Pages/OCES-II-certifikat-eksempler.aspx>).

Når certifikatet er hentet, kan det sættes op med eksempelvis SoapUI:



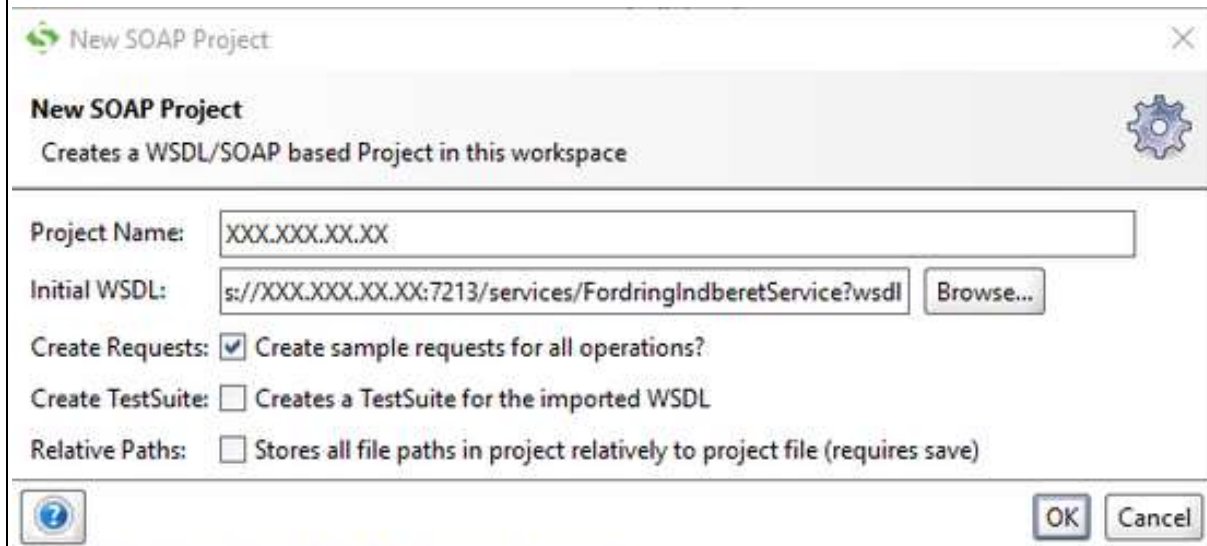
HTTP Settings	KeyStore:	C:\Users\brj\Desktop\certs\VOCES_gyldig.p12	Browse...
Proxy Settings	KeyStore Password:	••••••••	
SSL Settings	Enable Mock SSL:	<input type="checkbox"/> enable SSL for Mock Services	
WSDL Settings	Mock Port:		
UI Settings	Mock KeyStore:		Browse...
Editor Settings	Mock Password:		
Tools	Mock Key Password:		
WS-I Settings	Mock TrustStore:		Browse...
Global Properties	Mock TrustStore Password:		
Global Security Settings	Client Authentication:	<input checked="" type="checkbox"/> requires client authentication	
WS-A Settings			
Global Sensitive Information Tokens			
Version Update Settings			
AlertSite Connector Plugin			

Certifikatets password findes på siden, det er hentet fra (Test1234)

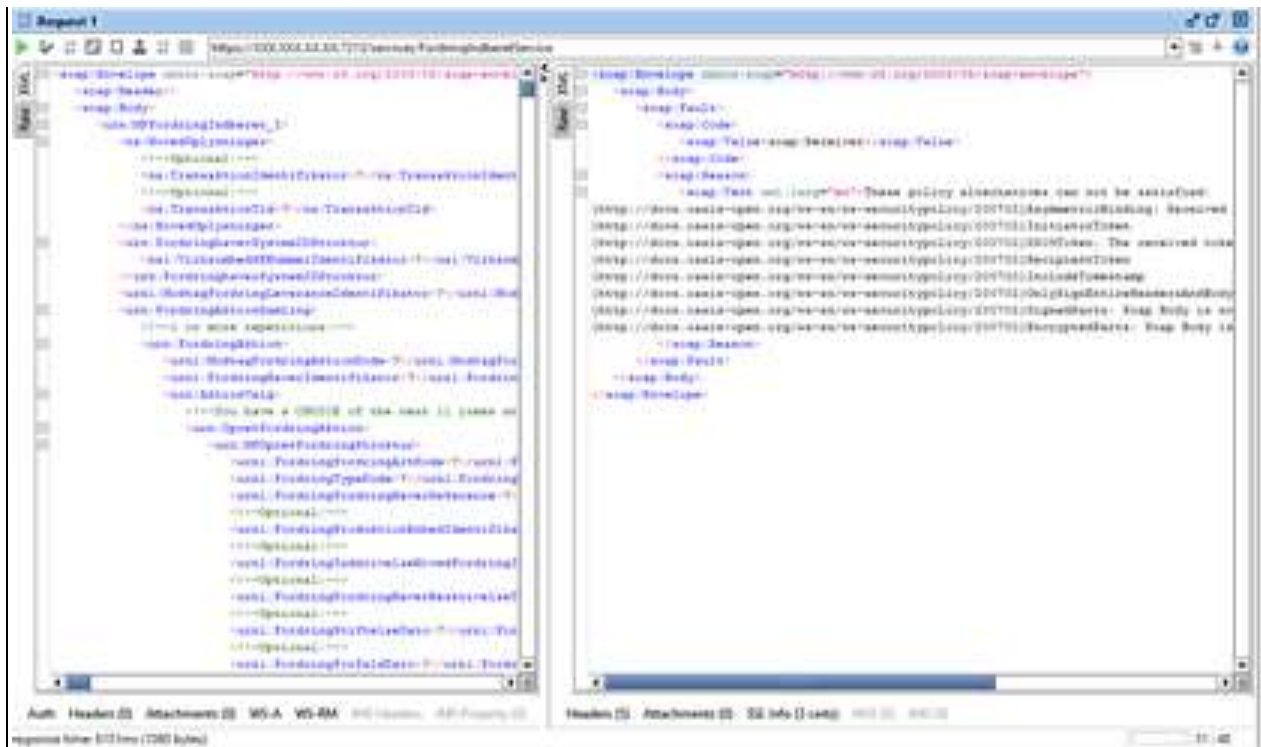
Test af service-kald

Opret forbindelse til servicen FordringIndberetService:

For docker-miljøer er det følgende service, man skal kalde: <https://nymf-b2b-soap-oio-dockXX-secure.inddr.dk/services/FordringIndberetService?wsdl>, hvor XX skal erstattes med det docker-nummer man har fået.



Send nu et request. Sørg for at pege på den udgående trafik mod HTTPS endpoint:



Ved forbindelse til miljøet fås et response som ovenstående. Egentlige indberetninger kræver kryptering af requestet. Derfor skal vi nu opsætte certifikater og nøgler for in-going og out-going beskeder. Disse gøres nedenunder:

Transport Layer Security (TLS)

TLS V1.2

For at sikre at data udveksles efter sikre, velafprøvede principper, og fuldt krypterede kanaler, benyttes HTTP over TLS. Det kun tilladt at forbinde til NyMF ved at benytte nyeste TLS version (TLS v1.2).

Desuden er der begrænsning på, hvilke ciphers, der er tilladt. NyMF er konfigureret til at afvise forbindelser, der anvender følgende ciphers:

- TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA

- TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA

SoapUI opsætning til TLS v1.2

Serveren understøtter kun TLS v1.2, som SoapUI skal konfigureres til:

1. Åbn folderen C:\Program Files\SmartBear\SoapUI-5.2.1\bin (den kan muligvis også gemme sig her, hvis man ikke kan finde den: C:\Program Files (x86)\SmartBear\SoapUI-5.2.1\bin)
2. Åbn filen SoapUI-5.2.1.vmoptions med fx Notepad
3. Tilføj linjen "-Dsoapui.https.protocols=TLSv1.2"
4. Genstart SoapUI

Web Service Security (WSS)

Web Services Security (WSS) krypterer indholdet og sørger for en sikker kommunikation med web services i PSRM.

WS-Policy

NyMF kræver, at indholdet af beskeden anvender følgende WS-Policy:

WS-Policy

```
<?xml version="1.0" encoding="UTF-8"?>
  <wsp:Policy
    xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702"
    xmlns:wsp="http://www.w3.org/ns/ws-policy"
    xmlns:wssu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wsssecurity-utility-1.0.xsd"
    wssu:Id="skat-b2b-x509-policy">
    <wsp:ExactlyOne>
      <wsp:All>
        <sp:AsymmetricBinding>
          <wsp:Policy>
            <sp:InitiatorToken>
              <wsp:Policy>
                <sp:X509Token sp:IncludeToken="http://docs.oasis-open.org/ws-
sx/ws-securitypolicy/200702/IncludeToken/AlwaysToRecipient">
                  <wsp:Policy>
                    <sp:WssX509V3Token10/>

```

```

        </wsp:Policy>
        </sp:X509Token>
    </wsp:Policy>
</sp:InitiatorToken>
<sp:RecipientToken>
    <wsp:Policy>
        <sp:X509Token sp:IncludeToken="http://docs.oasis-open.org/ws-
sx/ws-securitypolicy/200702/IncludeToken/Never">
            <wsp:Policy>
                <sp:RequireThumbprintReference />
                <sp:WssX509V3Token10/>
            </wsp:Policy>
        </sp:X509Token>
    </wsp:Policy>
</sp:RecipientToken>
<sp:AlgorithmSuite>
    <wsp:Policy>
        <sp:TripleDes/>
    </wsp:Policy>
</sp:AlgorithmSuite>
<sp:Layout>
    <wsp:Policy>
        <sp:Lax/>
    </wsp:Policy>
</sp:Layout>
<sp:IncludeTimestamp/>
<sp:OnlySignEntireHeadersAndBody/>
<sp:SignBeforeEncrypting/>
</wsp:Policy>
</sp:AsymmetricBinding>
    <sp:SignedParts>
        <sp:Body/>
    </sp:SignedParts>
    <sp:EncryptedParts>
        <sp:Body/>
    </sp:EncryptedParts>
</wsp>All>
</wsp:ExactlyOne>
</wsp:Policy>

```

I grove træk betyder det, at klienter forventes at:

- Indsætte tidsstempel.
- Signere besked ved brug af samme certifikat som SSL-forbindelsen (signatur algoritme: RSA_SHA1, Canonicalization: xml-exc-c14n, digest: sha1). Både body og tidsstempel signeres.
- Kryptere besked ved at anvende NyMF's offentlige certifikat (Symmetrisk algoritme: tripledes-cbc, nøglealgoritme: rsa-oaep). Kun body krypteres.

Keystore og truststore

SoapUI skal sættes op til kryptering og dekryptering af beskeder sendt over B2B med WSS. Det sker med client keystore og truststore.

Det tidligere hentede virksomhedscertifikat kan bruges som keystore, men der er også brug for et truststore til den offentlige del af servercertifikatet.

a) Som udgangspunkt kan truststore øverst på denne side benyttes: test_truststore.jks. (Et tilsvarende findes for PROD når man kommer så langt og skal koble på produktionen.)

Brug "password" som kodeord for at komme ind. Alternativt kan man selv oprette et truststore. Processen er beskrevet her:

Opsætning af TrustStore (valgfri)

Denne guide kan følges, hvis det ovenstående truststore ikke kan bruges. Guiden forudsættes, at du benytter et linux-miljø. Både Java og OpenSSL skal være installeret. Windows kan også benyttes, men da kan openssl-kommandoerne ikke nødvendigvis bruges.

- 1) Download det offentlige certifikat fra endpoint:

```
openssl s_client -showcerts -connect <ip>:<port> </dev/null 2>/dev/null|openssl x509 -outform PEM >clientTrustCert.pem
```

Erstat <ip> and <port> med den server, du kommunikerer med.

- 2) Opret JKS truststore, der indeholder certifikatet:

```
keytool -import -file clientTrustCert.pem -alias trustedServer -keystore clientTrustStore.jks -deststoretype JKS -storepass password.
```

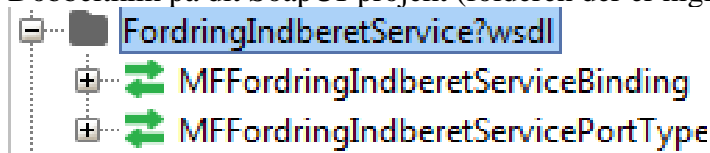
 Indtast yes, hvis der spørges til certifikatets autenticitet

Opsætning af WSS

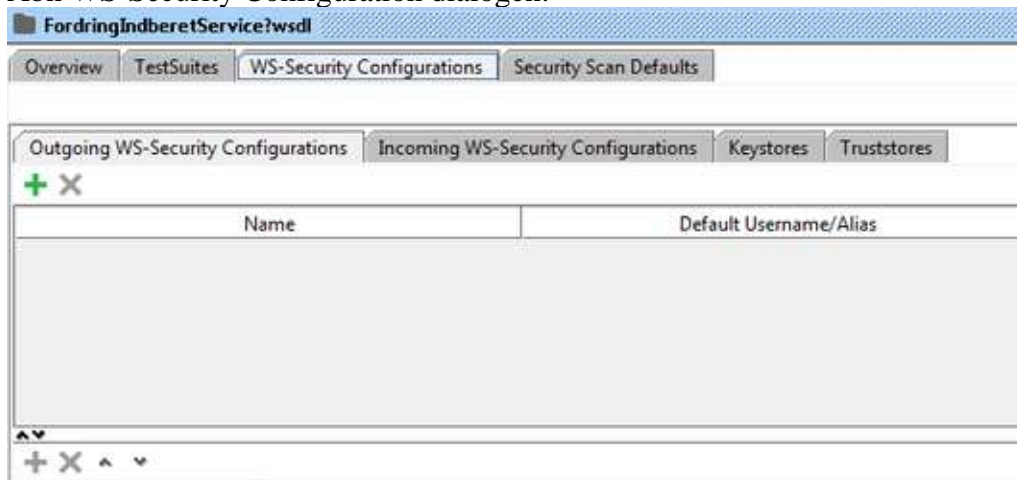
Efter ovenstående trin kan der i kommunikerer med serveren. Ved request får man en fejlmeddelelse i stil med *A security error was encountered when verifying the message*, hvis requestet ikke er krypteret. Det kan forhindres af WSS.

Opsætning af indgående og udgående kryptering og dekryptering i SoapUI

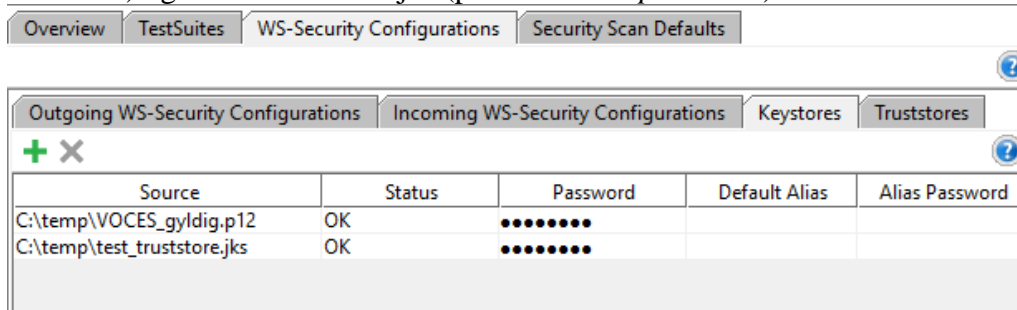
1. Dobbeltklik på dit SoapUI-projekt (folderen der er highlightet i blå):



Åbn WS-Security Configuration dialogen:

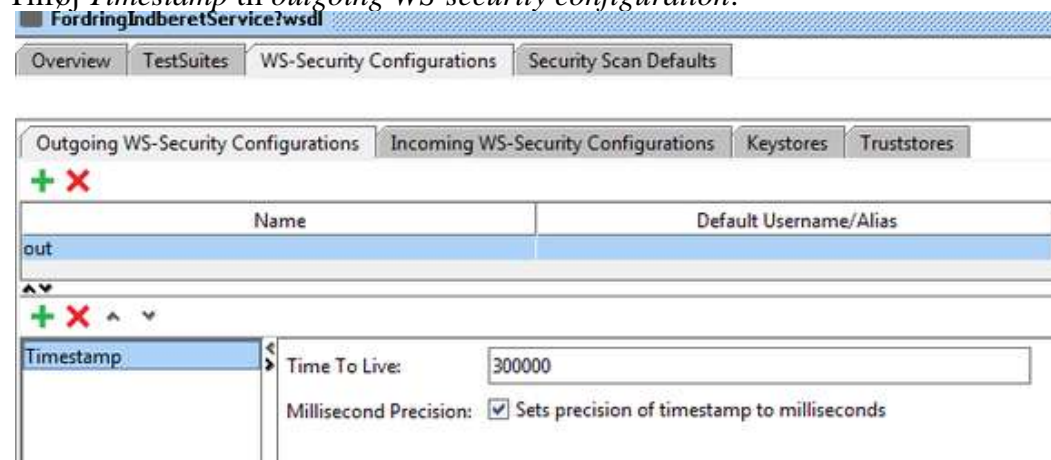


2. Gå til Keystores, og importerer både virksomhedscertifikatet (VOCES, password er Test1234) og clientTrustStore.jks (passwordet er *password*):

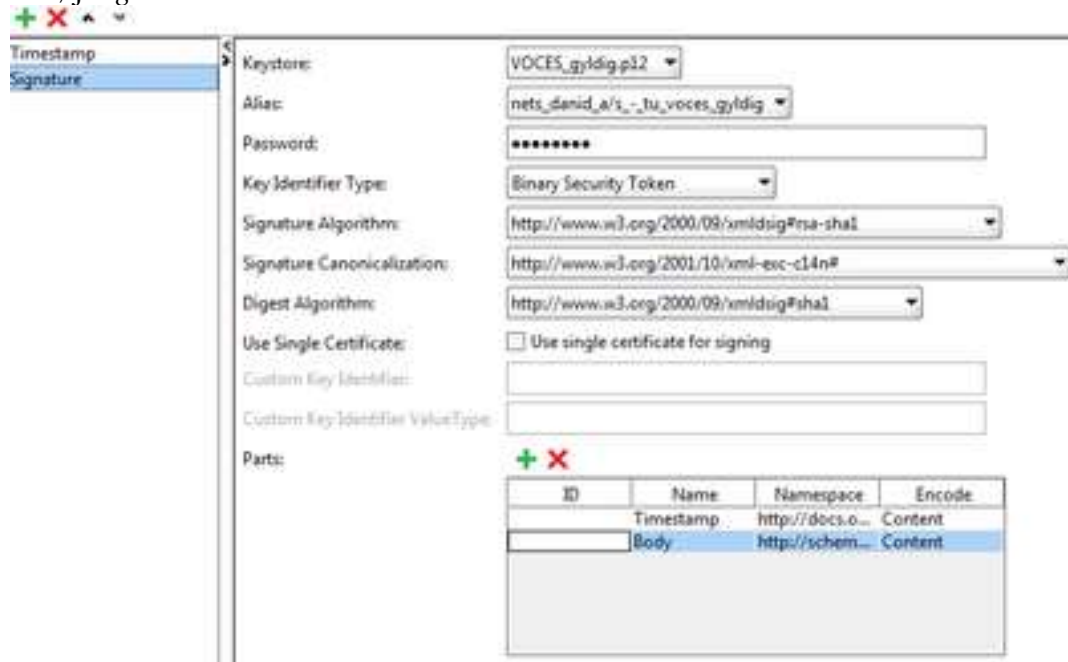


Alias og *alias password* skal ikke fyldes ud.

3. Tilføj *Timestamp* til *outgoing WS-security configuration*:



4. Tilføj *Signature*:



ID	Name	Namespace	Encode
	Timestamp	http://docs.o...	Content
	Body	http://schem...	Content

Sørg for at vælge *Binary Security Token* under *Key Identifier Type*.

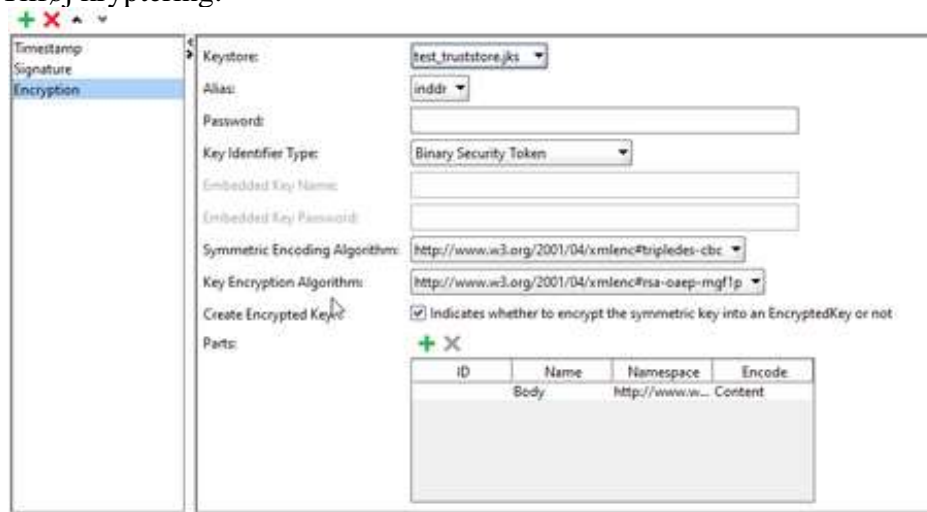
De valgte algoritmer skal være identiske med dem i screenshot ovenfor!

Signature password er VOCES-certifikatets password (Test1234)

Under *Parts* indtastes nedenstående information:

ID	Name	Namespace	Encode
	Timestamp	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd	Content
	Body	http://www.w3.org/2003/05/soap-envelope	Content

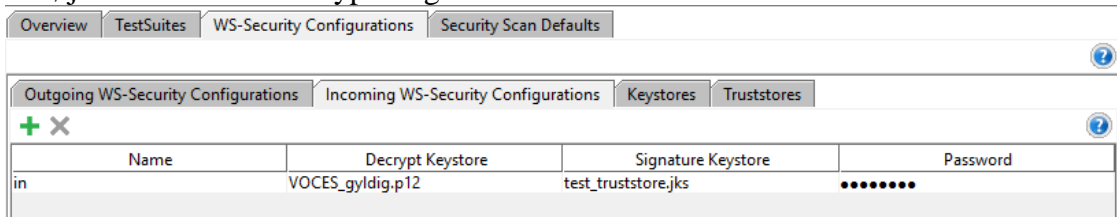
5. Tilføj kryptering:



ID	Name	Namespace	Encode
	Body	http://www.w...	Content

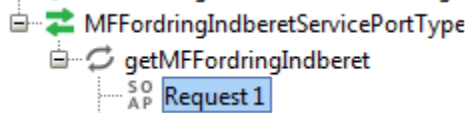
ID	Name	Namespace	Encode
	Body	http://www.w3.org/2003/05/soap-envelope	Content

- Sørg for at vælge *Binary Security Token* under *Key Identifier Type*. De valgte algoritmer skal være identiske med dem i screenshot ovenfor!
- Tilføj indkommende dekryptering:

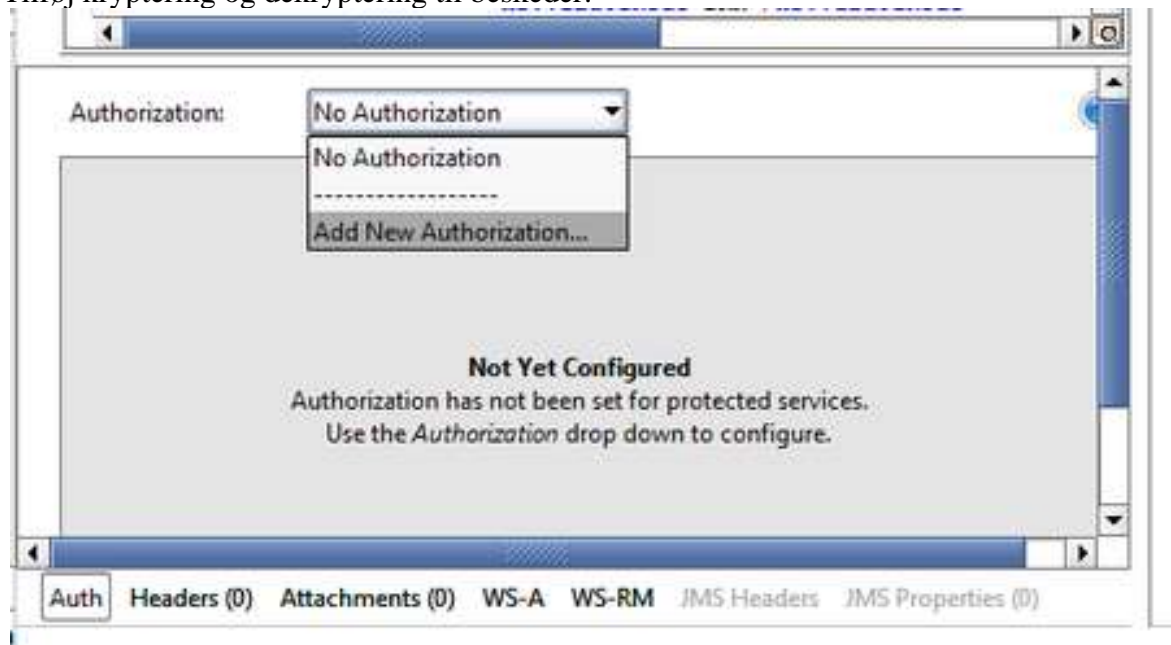


Det indkommende password er VOCES' password (Test1234).

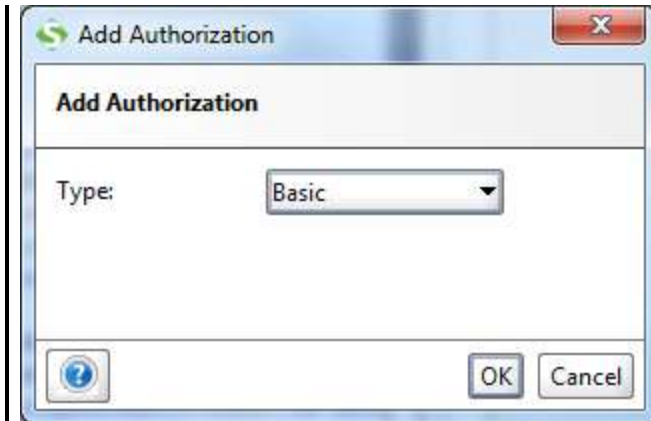
Åbn request-editoren, (dobbeltklik *Request1*, markeret blå nedenunder):



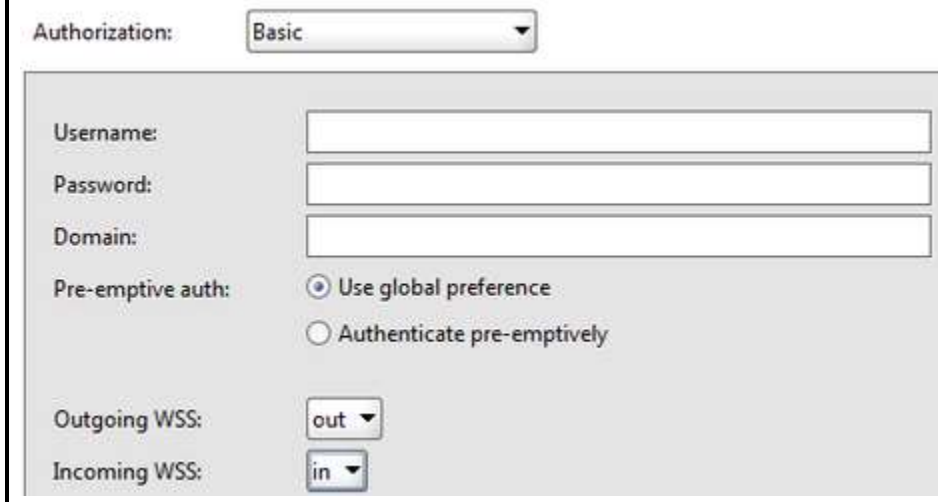
Tilføj kryptering og dekryptering til beskeder:



Vælg *Basic*:

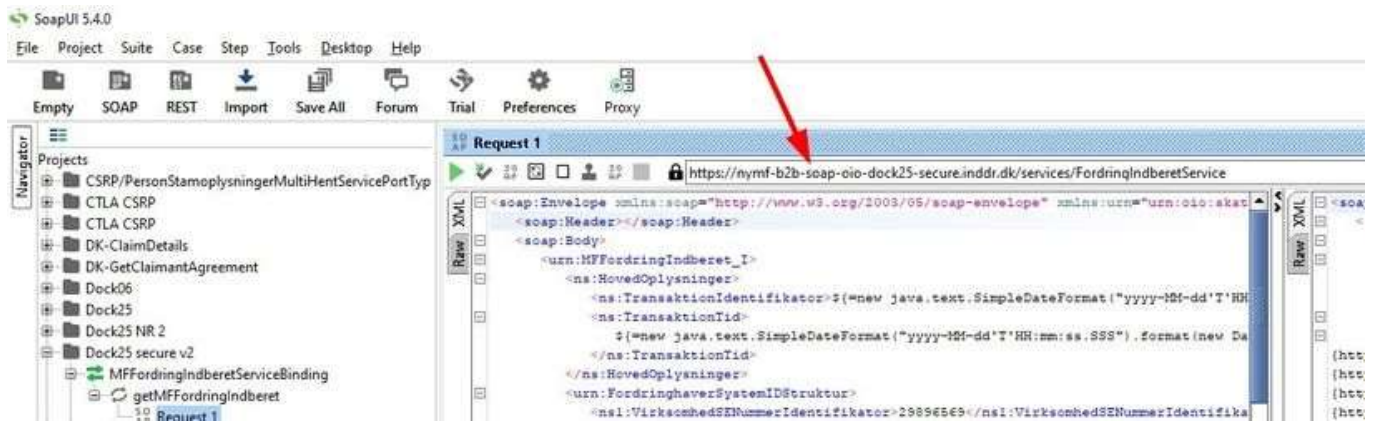


Vælg ind- og udgående kryptering:



Screenshot på opsætningen af et SoapUI projekt:

1. Download og importér en SoapUI-fil.
2. Husk at omdøbe IP-adressen:



▼ Genbrug af samme transaktionsID (også kaldet LeveranceID internt):

Error Code Persistence Exception

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <soap:Fault>
      <soap:Code>
        <soap:Value>soap:Receiver</soap:Value>
      </soap:Code>
      <soap:Reason>
        <soap:Text xml:lang="en">javax.persistence.PersistenceException:
org.hibernate.exception.DataException: could not extract
ResultSet</soap:Text>
      </soap:Reason>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Fejlen opstår, når man bruger det samme TransaktionsID <ns:TransaktionIdentifikator>. Løsningen er, at den skal være unik.

NB! LeveranceID skal være unikt! Der må heller ikke genbruges gamle LeveranceID'er fra DMI.

▼ 503 Service Unavailable: Mangler Host Name

SNI in SAP server

503 Service Unavailable

Denne fejl kan opstå, hvis SNI mangler i kaldet. **SNI** = Server Name Indicator

På nogle servere, eks. SAP servere, skal SNI bevidst slås til for at det ønskede hostnavn sendes med i requestet, ellers får man en 503 Service Unavailable.

Se mere her:

https://en.wikipedia.org/wiki/Server_Name_Indication

<https://security.stackexchange.com/questions/101965/ssl3-error-when-requesting-connection-using-tls-1-2/102018#102018>

<https://answers.sap.com/questions/473015/sap-ssl-handshake-failed.html>

Java

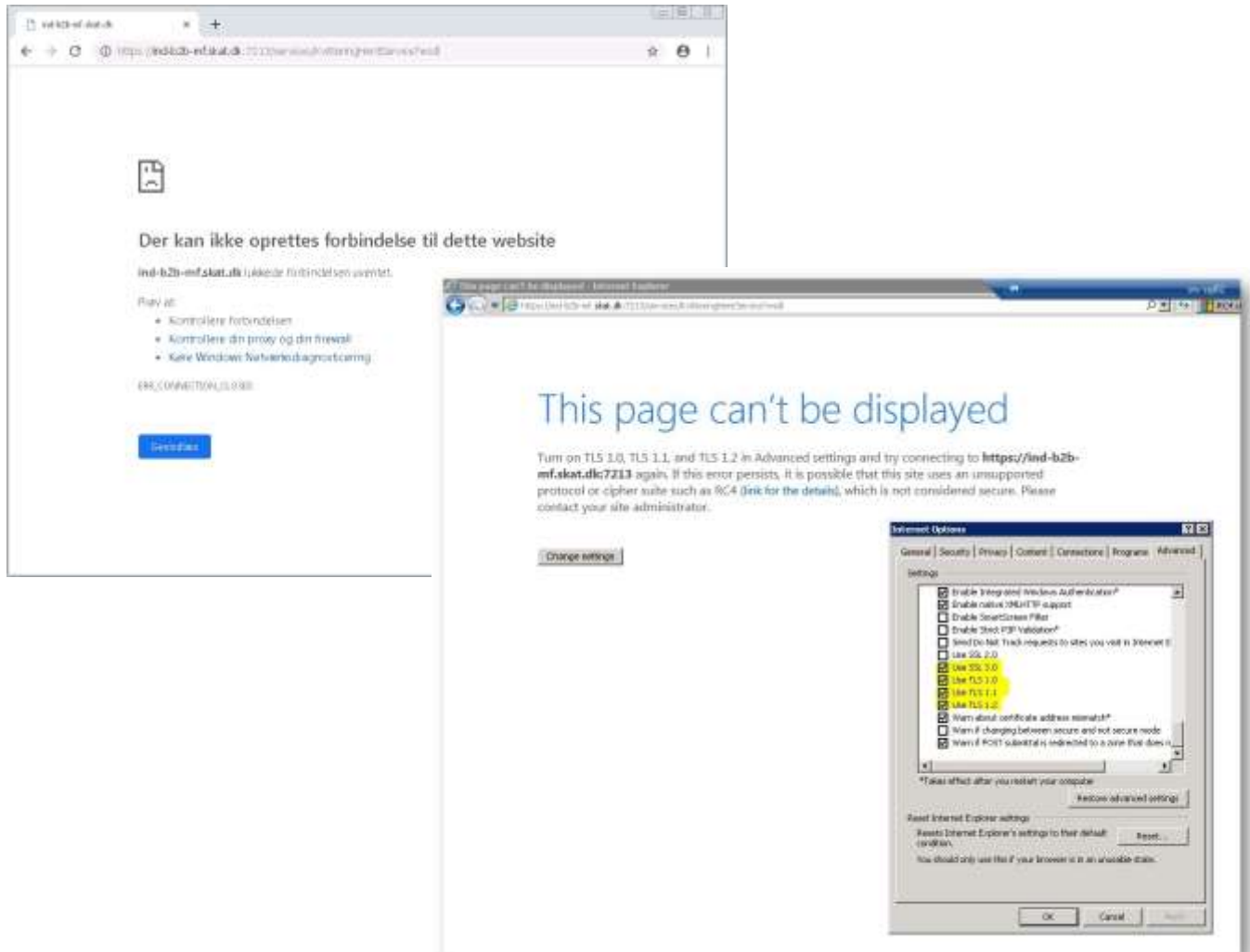
Java:

Man skal bruge Java full version 1.8.0.141 eller senere!

Tidligere versioner har en fejl som IKKE sender SNI med og dermed kommer man ikke længere en OpenShift miljøet; man rammer aldrig serveren og får fejl "503 Service Unavailable"

∨ Manglende produktionscertifikat hos fordringshaveren

Følgende fejl:



Skyldes manglende produktionscertifikat på fordringshavers server.

Se afsnit "Hvordan trækker man et certifikat ud fra serveren?" ovenover.

∨ Fejlbesked: A Security error was encountered when verifying the message

Hvis fordringshaveren møder følgende fejlbesked: An unsecured or incorrectly secured fault was received from the other party - A Security error was encountered when verifying the message, så kan det skyldes følgende:

- Undersøge med operations om b2b komponenten er ens på de to servere hvor disse er deployed
- Undersøge med operations om de to servere hvor b2b komponenten er deployed på er deployed ens og på samme tidspunkt-ish.
- Sikre at truststoren indeholder de certifikater der skal til for at der stoles på kommunikationen (ved virksomhedscertifikater fra TRUST 2048, så gælder dette intermediate og rod-certifikatet)

▼ FID is not supported

Når man kalder ind mod PROD kan man få fejlen: "FID is not supported" alternativt noget med fejl i GetEntityInformation.

Det er fordi man ikke har oprettet sin system-adgang i TastSelv-Erhverv.

Se guiden "Systembrugeroprettelse ved SKATs Tastselv" på [Gældsstyrelsens hjemmeside](#).

▼ NyMF is down. Service not reachable

Når man får denne fejl, er det fordi NyMF er nede.

PROD: Man vil typisk få den, hvis man har forsøgt at kalde NyMF i et servicevindue. Man skal benytte korrekt procedure for at undersøge, hvilke aktioner, der er gået igennem.

Dette er beskrevet her: [IndberetforDringService fejlhåndtering](#)

▼ No trusted certs found

Fejlen set fra FH:

Mar 22, 2019 2:55:09 PM dk.dsb.igp.secure.lib.ThirdPartySecurity postMessage

```
SEVERE: <?xml version="1.0" encoding="UTF-8" standalone="no"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"><soap:Body><soap:Fault><faultcode
xmlns:ns1="http://ws.apache.org/wss4j">ns1:SecurityError</faultcode><faultstring>A security
error was encountered when verifying the
message</faultstring></soap:Fault></soap:Body></soap:Envelope>
```

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"><soap:Body><soap:Fault><faultcode
xmlns:ns1="http://ws.apache.org/wss4j">ns1:SecurityError</faultcode><faultstring>A security
error was encountered when verifying the
message</faultstring></soap:Fault></soap:Body></soap:Envelope>
```

javax.xml.soap.SOAPException: A security error was encountered when verifying the message

Fejl fundet i loggen: Openshift -> dock38-> pods->nymf-b2b

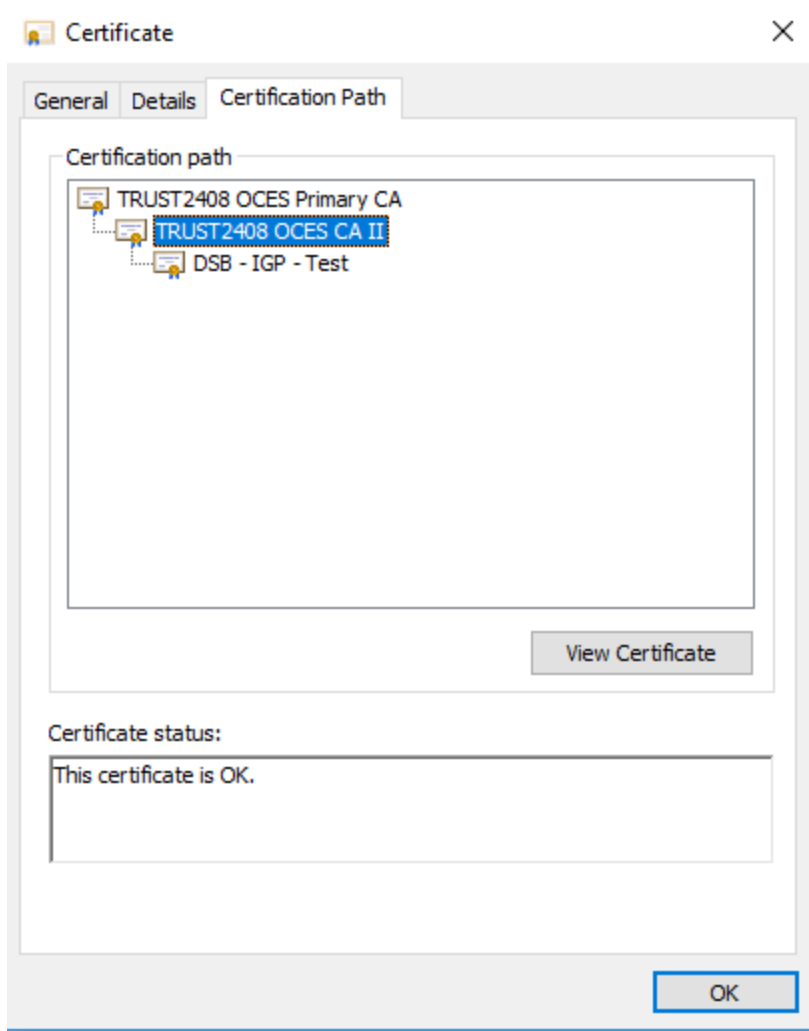


No trusted certs found

Det betyder, at der ikke stoles på certifikatet. Dette er sikkert, fordi intermediate/root certificater ikke er i truststore.

Løsning:

Få certifikat fra FH, tjek deres certifikat kæde:



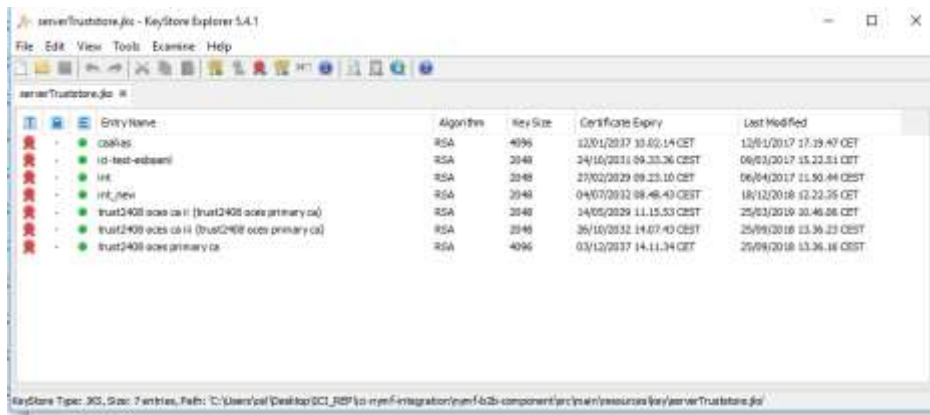
Hent truststore og kontroller det:

Fra github:

`ici-nymf-integration\nymf-b2b-component\src\main\resources\key\serverTruststore.jks`

Åben truststore med Keystore explorer (det er lettere end kommandoer)

Password er: storepass



Check at intermediate og root certificates er i store.

Hvis de ikke er, tilføj dem dertil.

Dette gøres ved at trykke View Certificate i ovenstående billede af FH-certifikat på intermediate og vælg Details → Copy To file → P7B format.

Inde i KeyStore explorer Tools → add trusted certs

Lav en ny branch, commit din nye truststore, opret pull request, få reviewet og merge ind.

Kør release-nymf Project Release-ici-nymf-integration

Releaseall Project ReleaseAll

Docker deploy nymf-b2b Docker - Deploy - nymf-b2b

All good.

✖ Certificate systemID does not match FordringsnaverSystemID in the message

Denne fejl kommer når systemid i beskeden ikke matcher det i certifikatet.

Her er eksempel med DSB:

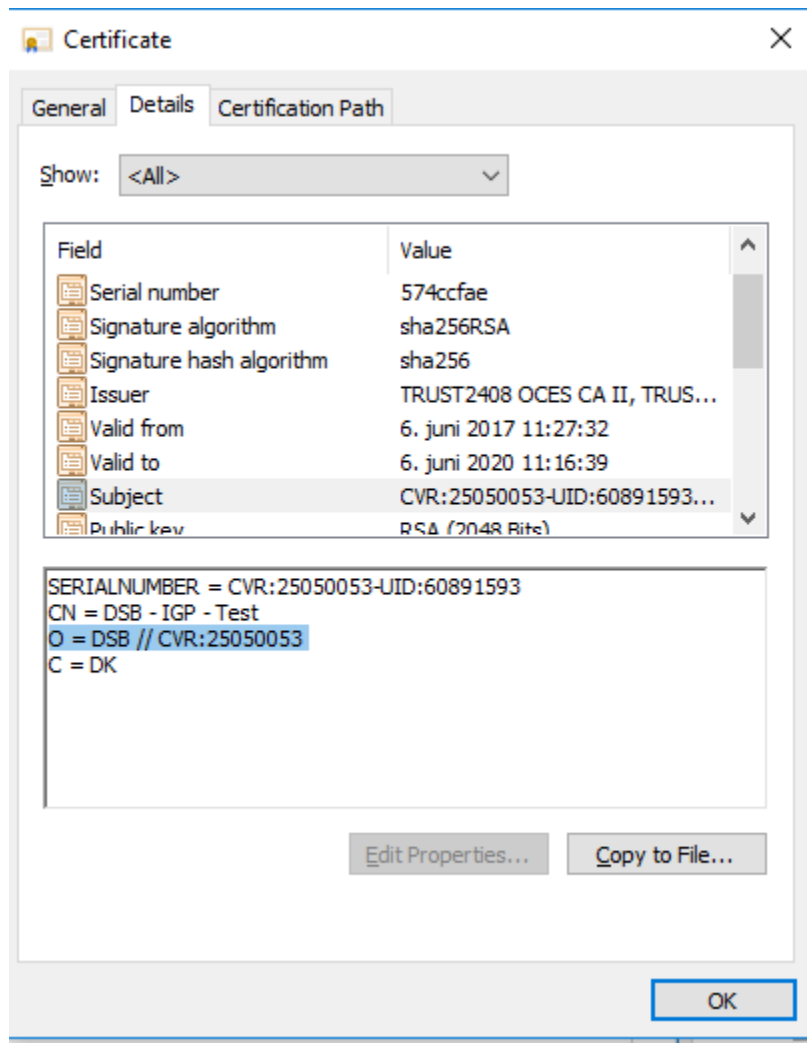
Det er pga. at denne værdi skal matche med subject værdien i certifikatet.

<urn:FordringsnaverSystemIDStruktur>

<ns1:VirksomhedSENummerIdentifikator>**25050053**</ns1:VirksomhedSENummerIdentifikator>

</urn:FordringhaverSystemIDStruktur>

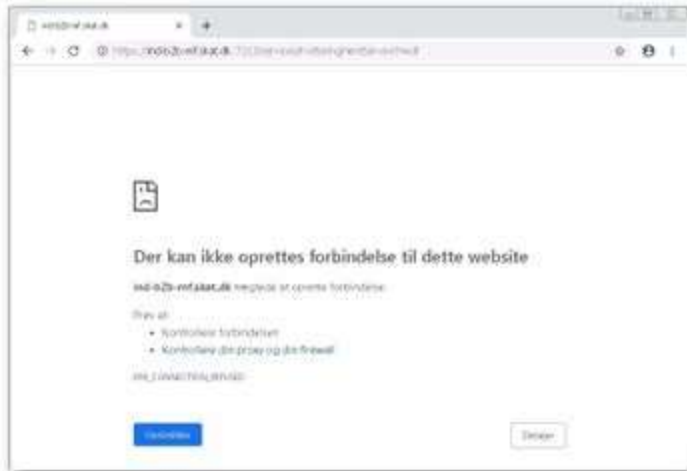
Certifikatet



Dette skal fordringshaver rette til. Er det for test, kan man rette SystemIndberetternummeret i fordringshaveraftalen på testmiljøet f.eks. til Nets certifikatet (som man typisk vil bruge til test)

▼ Nægtede at oprette forbindelse

Hvis der ikke er forbindelse med denne fejlbesked. (taget fra Metros onboarding, da de prøvede at forbinde fra deres PROD miljø til vores prod miljø.)



- Kontroller at de kalder på den rigtige URL (det er mod B2B). [PROD](#)
- Kontroller at deres IP er whitelistet (spørg Operations, eller fremfind case på toolkit <https://goto.netcompany.com/cases/GTO587/SKMICI/Lists/Tasks/AllItems.aspx>)
- Spørg Operations om de har modtaget pakker fra deres IP (de vil gerne have et tidspunkt fra hvornår adgang er forsøgt)
- Bed fordringshaveren om at kontrollere deres firewall (porten der bruges er ikke standard, derfor er den ofte lukket) og åbne op for porten hvis nødvendigt.
- Kontrollere at de kalder med korrekt certifikat (men der burde man få en anden fejl, da browseren har spurgt efter hvilket certifikat man vil bruge)

▼ SSL/TLS padding fejl

I forbindelse med KOBRA's restance-kørsel, blev der set forskellige SSL/TLS fejl i KOBRA's ende, eksempelvis:

Log-fejl

```
2019-08-13 21:19:21,696 WARN [[STUCK] ExecuteThread: '4' for queue:
'weblogic.kernel.Default (self-tuning)'] logging.LogUtils (LogUtils.java:443)
- Interceptor for {http://www.springframework.org/schema/beans}MFKvi
tteringHentServiceSynkronPortTypeImplService#{http://skat.dk/begrebsmodel/200
9/01/15/}getMFKvitteringHent has thrown exception, unwinding now
org.apache.cxf.interceptor.Fault: Connection has been shutdown:
javax.net.ssl.SSLException: Invalid TLS padding data
    at
org.apache.cxf.interceptor.LoggingInInterceptor.logging(LoggingInIntercep
tor.java:167)
    at
org.apache.cxf.interceptor.LoggingInInterceptor.handleMessage(LoggingInInterc
eptor.java:78)
    at
org.apache.cxf.phase.PhaseInterceptorChain.doIntercept(PhaseInterceptorChain.
java:262)
    at org.apache.cxf.endpoint.ClientImpl.onMessage(ClientImpl.java:800)
```

```
at
org.apache.cxf.transport.http.HTTPConduit$WrappedOutputStream.handleResponseInternal(HTTPConduit.java:1694)
at
org.apache.cxf.transport.http.HTTPConduit$WrappedOutputStream.handleResponse(HTTPConduit.java:1530)
at
org.apache.cxf.transport.http.HTTPConduit$WrappedOutputStream.close(HTTPConduit.java:1438)
at
org.apache.cxf.io.CacheAndWriteOutputStream.postClose(CacheAndWriteOutputStream.java:50)
at
org.apache.cxf.io.CachedOutputStream.close(CachedOutputStream.java:229)
at
org.apache.cxf.transport.AbstractConduit.close(AbstractConduit.java:56)
at
org.apache.cxf.transport.http.HTTPConduit.close(HTTPConduit.java:659)
```

Det viste sig, at de kunne fremtvinge fejlen, ved blive ved med at lave kvitteringhent kald på 1000 fordringer.

En ændring i Java.security i KOBRA's ende løste problemet.

Følgende post beskriver problemet:

<https://stackoverflow.com/questions/40964961/intermittent-javax-net-ssl-failure-bad-record-mac/41143041>

Løsningen er altså:

```
jdk.tls.disabledAlgorithms=SSLv3, DH, DHE, ECDH, ECDHE
```

En anden løsning er at opgradere fra Java 7 til Java 8.

▼ Client received SOAP Fault from server

This error can be due to following:

1. The time difference in the server at the client is big. They need to adjust the clock at their server
2. The encryption and decryption is not working at the client.

---[HTTP response - <https://ind-b2b-mf.skat.dk:7213/services/FordringIndberetService> - 500]---

null: HTTP/1.1 500 Server Error

Content-Length: 424

Content-Type: application/soap+xml; charset=UTF-8

Date: Tue, 25 Aug 2020 10:57:35 GMT

Server: Jetty(9.2.15.v20160210)

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-  
envelope"><soap:Body><soap:Fault><soap:Code><soap:Value>soap:Sender</soap:Value><soap:  
Subcode><soap:Value  
xmlns:ns1="http://ws.apache.org/wss4j">ns1:SecurityError</soap:Value></soap:Subcode></soa  
p:Code><soap:Reason><soap:Text xml:lang="en">A security error was encountered when  
verifying the  
message</soap:Text></soap:Reason></soap:Fault></soap:Body></soap:Envelope>-----  
-----
```

[WSGW] Something went wrong!

[com.sun.xml.internal.ws.fault.ServerSOAPFaultException](#): Client received SOAP Fault from server: A security error was encountered when verifying the message Please see the server log to find more detail regarding exact cause of the failure.

at

[com.sun.xml.internal.ws.fault.SOAP12Fault](#).getProtocolException(SOAP12Fault.java:214)

at

[com.sun.xml.internal.ws.fault.SOAPFaultBuilder](#).createException(SOAPFaultBuilder.java:116)

at

[com.sun.xml.internal.ws.client.sei.StubHandler](#).readResponse(StubHandler.java:238)

at

[com.sun.xml.internal.ws.db.DatabindingImpl](#).deserializeResponse(DatabindingImpl.java:189)

at

[com.sun.xml.internal.ws.db.DatabindingImpl](#).deserializeResponse(DatabindingImpl.java:276)

at

[com.sun.xml.internal.ws.client.sei.SyncMethodHandler](#).invoke(SyncMethodHandler.java:104)

```
at
com.sun.xml.internal.ws.client.sei.SyncMethodHandler.invoke(SyncMethodHandler.java:77)
    at com.sun.xml.internal.ws.client.sei.SEIStub.invoke(SEIStub.java:147)
    at com.sun.proxy.$Proxy39.getMFFordringIndberet(Unknown Source)
    at
com.supeo.wsgw.WSGWClientHandler.handleFordringIndberet(WSGWClientHandler.java:153)
    at com.supeo.wsgw.WSGWClientHandler.run(WSGWClientHandler.java:86)
    at java.lang.Thread.run(Thread.java:745)
```

Java tip: Slå full debugging til

Java Tip

Kør dit program med:

-Djavax.net.debug=all