

Danske Bank

Encryption, Signing and Compression in Financial Web Services

Details of how to call the Danske Bank financial web service

Table of Contents

Version history.....	3
Introduction.....	3
Regarding encryption	3
Our implementation of encryption	3
Regarding the Encryption and EncryptionMethod elements.....	5
Regarding the XML Encryption specification.....	6
Encryption Algorithms supported	7
Our implementation of compression	7
Regarding signatures	8
Multiple business signatures	8
Supported algorithms.....	8
Requirements on the signatures	8
Future changes to the specification	9
References	9
Appendix A: Example XML and SOAP	10
Generation of the ApplicationRequest.....	10
Signing the ApplicationRequest (business signature)	10
Encrypting the ApplicationRequest	11
Generation of the SOAP message	13
Signing the SOAP message (transport signature).....	15
Appendix B: Web service URL.....	18

Version history

Version	Date	Description of change	Author
2.2	5. January 2010	RFC 1952 added to clarify gzip. Section on future changes removed. Example XML added. Section on signatures added.	Mikkel T. Jensen
2.3	22. February 2010	RSA-OAEP replaced by RSA-v1.5. More requirements to signatures added.	Mikkel T. Jensen
2.4	16. April 2010	Section added on future changes to the financial WS specification. Introduction changed.	Mikkel T. Jensen
2.4.1	4. March 2010	Example XML corrected to use RSA 1.5.	Mikkel T. Jensen
2.4.2	18. April 2012	Corrected Introduction text	Christian Enevold
2.4.3	29. September 2012	Changes due to renaming of Sampo Pankki to Danske Bank.	Mikkel T. Jensen
2.4.4	4. December 2012	Changes to URL.	Lea Troels Møller Pedersen
2.4.5	20. August 2014	Corrected mistake with optional encryption	Mikkel T. Jensen
2.4.6	16. September 2014	Corrected diagram	Mikkel T. Jensen
2.4.7	12. May 2015	Error removed from transport signature example XML.	Mikkel T. Jensen

Introduction

The Danske Bank Web Services solution is build on the Web Services specifications from the Federation of Finnish Financial services[1].

This document clarifies the use of XML encryption and compression and how this is implemented in Danske Bank. This document was first written to cover details of the security elements in the Financial Web Service. However, the requirements on XML encryption and enveloped signatures also apply to the PKI Web Service.

Regarding encryption

Our implementation of encryption

XML encryption is applied at the ApplicationRequest/ApplicationResponse level. This means that the ApplicationRequest (or ApplicationResponse) element is replaced by an EncryptedData element from the XML encryption standard. The EncryptedData contains the actual encrypted data (in a CipherData subelement) as well as sub-elements necessary for decryption (EncryptionMethod and KeyInfo). The reason this approach can work is that the ApplicationRequest/ApplicationResponse/EncryptedData element is base64 encoded when the actual web service call is made. This means that there is no schema-check of the ApplicationRequest/Response schema on the web service level. The schema-check of the ApplicationRequest/Response should not be done until the actual XML has been reconstructed using

base64 decoding, followed by decryption. The order of processing when receiving a web service call should be as follows:

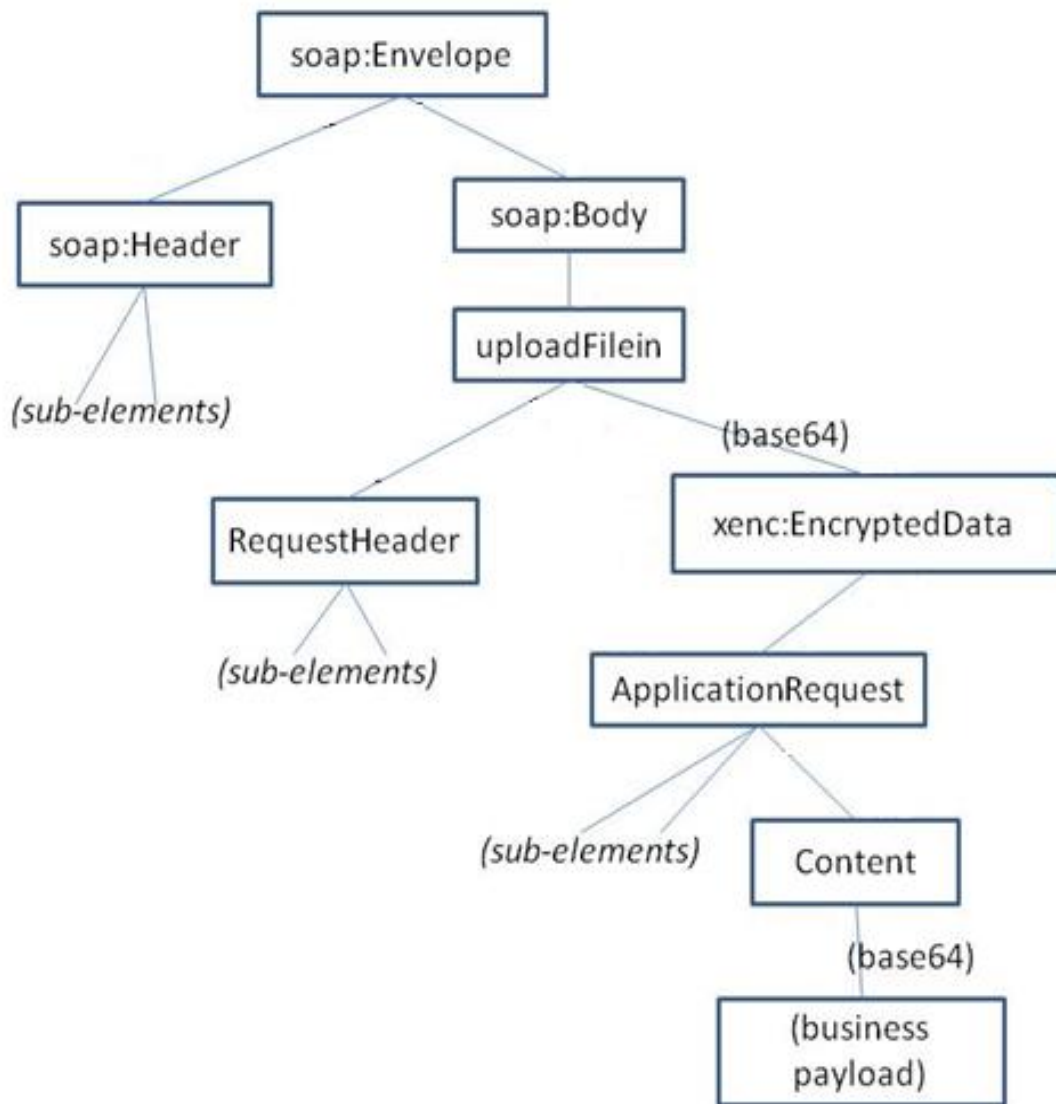
1. The web service call arrives at the server
2. As part of the web service call, the incoming message is checked against the WSDL. This does not include schema checking the ApplicationRequest, due to the base64 encoding.
3. The SOAP header and security information in it is processed, including the transport signature. If everything is not correct, an error message is returned and processing stops.
4. Processing based on RequestHeader can be done.
5. XML is constructed by base64 decoding the content of the ApplicationRequest in the incoming message (in the WSDL, it is defined to be of type xsd:base64Binary). The base64 decoding results in an EncryptedData element.
6. The EncryptedData element is decrypted using XML encryption functionality. This creates an ApplicationRequest element.
7. The ApplicationRequest element can be schema checked now.
8. Further processing of the ApplicationRequest.

The processing of a received ApplicationResponse on the client follows the same pattern. In the same way, the order of processing in the construction of XML prior to a web service call should be as follows:

1. The ApplicationRequest element including the Content element is created.
2. The ApplicationRequest is Digitally signed (business signatures).
3. The ApplicationRequest element is encrypted, producing an EncryptedData element.
4. The EncryptedData element is base64 encoded.
5. A RequestHeader and a soap Body are produced.
6. The base64 data produced in step 4 is inserted in the soap Body in the ApplicationRequest element.
7. The rest of the soap message is constructed (soap Header and soap Envelope)
8. The soap message is digitally signed (transport signature).
9. The webservice is called.

The construction of an ApplicationResponse on the server follows the same pattern. This description can be compared to the message lifecycle description on page 9 of the specification.

The layering of information in the web-service call is visualized in the following figure:



The figure shows the soap envelope containing the soap Header and soap Body. The soap Body contains an uploadFilein element (the name of this element depends on the operation called, it could also be getUserInfoin, downloadFileListin, downloadFilein, or deleteFilein). This element contains a RequestHeader, and an element containing base64Binary coded data. The base64Binary data can be decoded an EncryptedData element. The EncryptedData element will decrypt into an ApplicationRequest element.

The figure is closely related to the figures on pages 15 and 16 of the specification (version 1.05).

Regarding the Encryption and EncryptionMethod elements

The ApplicationRequest and ApplicationResponse elements contain subelements called *Encryption* (*Encrypted* in ApplicationResponse) and *EncryptionMethod*. These elements can give rise to confusion, since for instance the description of the Encryption elements states:

'If this element is present and the content is the string "true" (case-sensitive) it means that the Content is encrypted or the requested data should be encrypted by the bank'

The first part of the sentence is misleading, since if the Encryption element is visible (i.e. not itself in an encrypted form) then the ApplicationRequest and the Content are no longer encrypted (they may have been in a previous processing step). The only true use of the element is stated in the second part of the sentence: If the element contains "true", the response should be encrypted by the bank. This interpretation also implies that the Encrypted element in ApplicationResponse serves no purpose, since no reply is needed in response to the response.

Similarly, the EncryptionMethod element is only used to communicate the cipher to be used in the response. For this, one EncryptionMethod element is inadequate, since two encryption ciphers are needed: One for symmetric encryption to encrypt the actual data, and one for asymmetric encryption to encrypt the symmetric key. This problem should be solved in a future version of the specification. At present it should be up to the individual banks to decide how to handle this. Maybe the same encryption ciphers used in the request should be used in the response, or maybe the same ciphers should always be used.

The specification does not state a default behavior if the Encryption and EncryptionMethod elements are not present in ApplicationRequest.

Regarding the XML Encryption specification

An introduction to the XML Encryption specification is outside the scope of this text. It is recommended to read the specification, which can be found at <http://www.w3.org/TR/xmlenc-core/>. However, since the specification contains some degrees of freedom as to the XML elements and their content, a brief description of the expected content of an EncryptedData element (which is the root element of encrypted data) will be given here. The encryption used is in two layers:

1. The actual business data is encrypted using symmetric cryptography. The key used to perform this encryption is called the *ephemeral key*. A new ephemeral key is generated every time a message is encrypted. In the example below, the encrypted business data is contained in the EncryptedData/CipherData element. The symmetric cipher is declared in the EncryptedData/EncryptionMethod/@Algorithm attribute.
2. The ephemeral key is encrypted using asymmetric cryptography. The encrypted ephemeral key is contained in the EncryptedData/KeyInfo/EncryptedKey element. The asymmetric cipher is declared in the EncryptedData/KeyInfo/EncryptedKey/EncryptionMethod/@Algorithm attribute. The actual encrypted ephemeral key is contained in the EncryptedData/KeyInfo/EncryptedKey/CipherData element. The certificate used to perform the asymmetric encryption is contained in the EncryptedData/KeyInfo/EncryptedKey/KeyInfo/X509Data/X509Certificate element.

The expected structure of the EncryptedData element can be found in the abbreviated example below.

```
<xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
Type="http://www.w3.org/2001/04/xmlenc#Element">
  <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripleDES-cbc"/>
  <dsig:KeyInfo xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
```

```

<xenc:EncryptedKey>
  <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
  <dsig:KeyInfo>
    <dsig:X509Data>
      <dsig:X509Certificate>MIIDyT...UijzrQQ==</dsig:X509Certificate>
    </dsig:X509Data>
  </dsig:KeyInfo>
  <xenc:CipherData>
    <xenc:CipherValue>NoO3acd...ggQWA=</xenc:CipherValue>
  </xenc:CipherData>
</xenc:EncryptedKey>
</dsig:KeyInfo>
<xenc:CipherData>
  <xenc:CipherValue>279oO2AZ6...okVnVDvl+KfG</xenc:CipherValue>
</xenc:CipherData>
</xenc:EncryptedData>

```

Messages sent to the bank must obey the following rules:

- The EncryptedData/KeyInfo/EncryptedKey/KeyInfo/X509Data/X509Certificate element must be present, and it must contain the base64 encoded certificate used to perform the encryption. This is necessary since at one time the bank may have several valid encryption certificates.
- The actual business data must be encrypted using a symmetric cipher. It is not allowed to use asymmetric encryption to encrypt the business data directly. This is because encryption and decryption using asymmetric ciphers is much more CPU intensive than symmetric ciphers.

Clients should use standard software solutions (libraries) supporting the XML encryption specification rather than implementing it on their own.

Encryption Algorithms supported

The encryption algorithms supported by the bank are:

- Symmetric encryption: 3DES. <http://www.w3.org/2001/04/xmlenc#tripleDES-cbc>
- Asymmetric encryption: RSA-v1.5. http://www.w3.org/2001/04/xmlenc#rsa-1_5

Other ciphers may be added later.

Our implementation of compression

Regarding compression, it should be applied at the Content element level. The Content element contains base64Binary coded data. Depending on the content of the Compression element, the base64Binary data can be in different formats:

1. If the Compression element contains 'true', the data is compressed. For decompression and decoding, the base64Binary data should be converted into binary data, which should be input to the decompression algorithm. The binary output of the decompression algorithm will be the legacy data or SEPA format XML.

2. If the Compression element contains 'false', the data is not compressed. For decoding, the base64Binary data should be base64 decoded. The binary output of this will be the legacy data or SEPA format XML.

When generating compressed data inside the Content element, the following procedure should be used:

The binary legacy data or the SEPA XML is used as input to the compression algorithm. The resulting binary data is base64 encoded and placed inside the Content element. Notice, that the input to the compression algorithm is the **binary** legacy or SEPA XML data, not a base64 encoded version of the data.

Regarding the compression algorithm, gzip (RFC1952) must be used. The CompressionMethod element should contain the string "gzip".

Regarding signatures

Multiple business signatures

The Financial Messages specification allows for up to three enveloped business signatures in the ApplicationRequest element. However, if more than one business signature is to be used, care has to be taken.

If enveloped signatures based on the standard enveloped signature transform (<http://www.w3.org/2000/09/xmldsig#enveloped-signature>) are used, only the last signature will be verifiable. This is because the standard enveloped signature transform removes only the signature being verified, and leaves other signatures in the element still in place when the message digest is calculated. This effectively means that the message digest value of the signed element is changed every time a signature is added. Thus, the first signature will not be verifiable after the second signature has been added. If a third signature is added, the second signature will not be verifiable either.

The best solution to this problem probably is to not support multiple business signatures until a new version of the specification [1] and the corresponding schemas [2] has been made.

Supported algorithms

The following algorithms should be used in the digital signatures:

- For message digesting, SHA1 (<http://www.w3.org/2000/09/xmldsig#sha1>) should be used.
- For signing, RSA based on SHA1 (<http://www.w3.org/2000/09/xmldsig#rsa-sha1>) should be used.
- For canonicalization, exclusive canonicalization without comments (<http://www.w3.org/2001/10/xml-exc-c14n#>) should be used.

Requirements on the signatures

When reading the XMLDSIG standard, there is a lot of liberty in the way a signature can look. In order to be accepted by Danske Bank, every signature must contain the certificate used to create the signature.

For enveloped style signatures (business signatures), this means that a Signature/KeyInfo/X509Data/X509Certificate element must be found. The element must contain the signing certificate in base64 form.

For the WSSEC signatures (transport signatures), it means that a Security/BinarySecurityToken element must be found. The BinarySecurityToken must contain the signing certificate in base64 form, and it must be referenced from the Signature/KeyInfo/SecurityTokenReference element.

See the Appendix for examples of signatures satisfying these requirements.

Additionally, there is a time requirement on the transport signature. The transport signature contains a timestamp indicating at what time the signature was made. The signature will only be accepted if this timestamp is less than 60 minutes old. The interval in which the signature will be accepted is subject to change.

Future changes to the specification

Work has started to revise the financial web-service specification [1] and the corresponding schemas [2] to solve the problems described in this document. Care will be taken to make the changes backwards compatible whenever possible.

References

1. Security and Message Specification For Financial Messages Using Web Services. Nordea, OP-Pohjola Group, Sampo Bank. Version 1.05.
2. The WSDL and schema files for the financial services: BankCorporateFileService_20080616.wsdl, ApplicationRequest_20080918.xsd, ApplicationResponse_20080918.xsd.
3. XML Encryption Syntax and Processing. W3C. <http://www.w3.org/TR/xmlenc-core/>
4. RFC1952 - GZIP file format specification version 4.3. <http://www.gzip.org/zlib/rfc-gzip.html>

Appendix A: Example XML and SOAP

This appendix will go through the generation of an example signed and encrypted SOAP message. The XML shown here is delivered in a zip-file along with this document.

Generation of the ApplicationRequest

First a business content and ApplicationRequest element must be generated. This corresponds to step 1 on the list on page 4. An example ApplicationRequest is found below:

```
<bxid:ApplicationRequest xmlns:bxid="http://bxid.fi/xmldata/">
  <bxid:CustomerId>ABC123</bxid:CustomerId>
  <bxid:Command>UploadFile</bxid:Command>
  <bxid:Timestamp>2009-12-17T09:30:47Z</bxid:Timestamp>
  <bxid:Environment>TEST</bxid:Environment>
  <bxid:Encryption>true</bxid:Encryption>
  <bxid:Compression>true</bxid:Compression>
  <bxid:CompressionMethod>gzip</bxid:CompressionMethod>
  <bxid:SoftwareId>CustomerSoftwareId</bxid:SoftwareId>
  <bxid:FileType>pain.001.001.02</bxid:FileType>
  <bxid:Content>UjBsR09EbGhjZ0dTQUxNQUFBUUNBRU1tQ1p0dU1GUXhEUzhi</bxid:Content>
</bxid:ApplicationRequest>
```

Note that the business content and the attributes may not make sense in this example.

Signing the ApplicationRequest (business signature)

A business signature is added to the ApplicationRequest. The signature is of the enveloped type. For this the signing certificate of the customer is used. This corresponds to step 2 on the list on page 4:

```
<bxid:ApplicationRequest xmlns:dpstate="http://danskebank.dk/AGENA/SEPA/dpstate"
  xmlns:sig="http://danskebank.dk/AGENA/SEPA/SigningService" xmlns:bxid="http://bxid.fi/xmldata/">
  <bxid:CustomerId>ABC123</bxid:CustomerId>
  <bxid:Command>UploadFile</bxid:Command>
  <bxid:Timestamp>2009-12-17T09:30:47Z</bxid:Timestamp>
  <bxid:Environment>TEST</bxid:Environment>
  <bxid:Encryption>true</bxid:Encryption>
  <bxid:Compression>true</bxid:Compression>
  <bxid:CompressionMethod>gzip</bxid:CompressionMethod>
  <bxid:SoftwareId>CustomerSoftwareId</bxid:SoftwareId>
  <bxid:FileType>pain.001.001.02</bxid:FileType>
  <bxid:Content>UjBsR09EbGhjZ0dTQUxNQUFBUUNBRU1tQ1p0dU1GUXhEUzhi</bxid:Content>
  <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="">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
        <DigestValue>A3OH3jwrbbcYxX3Nv5+eFyEuTww=</DigestValue>
      </Reference>
    </SignedInfo>
```

```
<SignatureValue>DneTCqbKEbQUE/TpaMILpOKJmB0asSfWr5O53M1HVwwE4JQqTJ3XNeZtB21C/yoYKP
Rky3lz8du+ObuhvuD0TiCASJZZw7FkxcpCm5+mAJDjCeB1KD+zPGjtGCo/B5Fcu6FfMcBT1/Q3oWMUuUP
GxfOKxWg1Lf1hUPpmjDtNF6Y=</SignatureValue><KeyInfo><X509Data><X509Certificate>MIIDljCCAougA
wIbAgIeL4j5xzANBqkqhkIG9w0BAQUFADBpMQswCQYDVQQGEwJESzEQMA4GA1UEBxMHRGVubWFy
azEaMBGGA1UEChMRRRGFuc2tIIEJhbmsgR3JvdXAxEADAQgNVBAwTB0FwcGxTSW4xGjAYBgNVBAMUE
UZpbm5pc2hfU0VQQV90ZXN0MB4XDTA5MTAyNjEyMDMzMVVoXDTEyMDIwMzEyMDMzMVowaTElMAkG
A1UEBhMCREsxEDAQgNVBAwTB0Rlbn1hcmsxGjAYBgNVBAoTEURhbnNrZSBCYW5rIEdyb3VwMRAw
DgYDVQQLLEwdBcHBsU0luMR0wGAYDVQQDFBFGaW5uaXNoX1NFUEFfdGVzdDCBnzANBqkqhkIG9w0B
AQEFAAOBJQAwgYkCgYEAYKLcdWd6TvZ+AF3zAabivvf+s+QoTnxQ4+DOI8mDBMpo7Ka41i59u79qyHa24
3HuQgW4eCMist+CUfY/Ynu6DgsP/NFduuV2VZNMdQIKwJMaytQ1zhYufMeiw8YdgGk8qVjZJOMwbCjnDI
1+da0CvEeEs1ZUZtVK9nRDDsnG9cCAwEAAaOB1jCB0zAMBgNVHRMERTADAQH/MB0GA1UdDgQWBb
Ru/JyaNnOeCQC6jNdNZI/qfLNoYTCBlgYDVR0jBIGOMIGLgBRu/JyaNnOeCQC6jNdNZI/qfLNoYaFtpGswaT
ELMAkGA1UEBhMCREsxEDAQgNVBAwTB0Rlbn1hcmsxGjAYBgNVBAoTEURhbnNrZSBCYW5rIEdyb3Vw
wMRAwDgYDVQQLLEwdBcHBsU0luMR0wGAYDVQQDFBFGaW5uaXNoX1NFUEFfdGVzdIIEL4j5xzALBgN
VHQ8EBAMCArwwDQYJKoZIhvcNAQEFBQADgYEAq6rHlg41XYLMjOLZ33toWKL0s3uWJuchK0F6TKAXS
Np7Qo34hx04P0MidOXuj18mYZkU/usVHn/L6rkWsfJretpO0xWTzw50XSm83JUQIYeQyMiU/w7ByJSNITa8
G1i4nKWjZ/kcl2Ur0oDUTTF/X8mIC24EkDCI9RQjGEYdUvE=</X509Certificate><X509IssuerSerial><X509Is
suerName>CN=Finnish_SEPA_test, OU=AppSIln, O=Danske Bank Group, L=Denmark,
C=DK</X509IssuerName><X509SerialNumber>797505991</X509SerialNumber></X509IssuerSerial></X50
9Data></KeyInfo></Signature></bxd:ApplicationRequest>
```

Encrypting the ApplicationRequest

The signed ApplicationRequest element is encrypted using XML encryption. For this, the encryption certificate of the bank is used. This corresponds to step 3 on the list on page 4:

```
<xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
Type="http://www.w3.org/2001/04/xmlenc#Element">
  <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripleDES-cbc"/>
  <dsig:KeyInfo xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
    <xenc:EncryptedKey>
      <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
      <dsig:KeyInfo>
        <dsig:X509Data>
          <dsig:X509Certificate>MIIDyTCCArGgAwIbAgIKNDQ0NDQzMDAwMjANBqkqhkIG9w0BAQsFADCBxD
EQMA4GA1UEAxMHRERJHUk9PVDELMAkGA1UEBhMCREsxEzARBgNVBAcTCKNvcGVuaGFnZW4xEDA
OBgNVBAgTB0Rlbn1hcmsxGjAYBgNVBAoTEURhbnNrZSBCYW5rIEdyb3VwMRAwGAYDVQQLLEwEYm9u
za2UgQmFuayBHcm91cDEYMBYGA1UEBRMjExMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYy
gNVBCoTADkxMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYy
wMTIwMDAwWjCBxTERMA8GA1UEAxMIREJHUk9JZUFQxXzA1UEBxMHRERJHUk9JZUFQxXzA1UEBxMHRER
b3BlbmhZ2VvMRAwDgYDVQQLLEwEZW5tYXJrMR0wGAYDVQQLKEwEYm9uza2UgQmFuayBHcm91cDE
aMBGGA1UECXMRRRGFuc2tIIEJhbmsgR3JvdXAxEADAQgNVBAUTDzYxMTI2MjYyMjYyMjYyMjYyMjYy
A1UEBBMAMQkwBwYDVQQQeEwEYm9uza2UgQmFuayBHcm91cDEYMBYGA1UEBRMAMIGdMA0GCSqGSIb3DQEB
AQUAA4GLADCBhwKBgQCpae2tNbCmPRni74t4pXpzUwsUTRBeazPpchZHjn+HBpfQAH66kCzOcxLHORL
71CFYIj6B4WrqXz4SUIIdQpKu67vSh5ZsLyyXPZniCo+DtX2zOYGEigzq6hlyIR5XKnXkZfJSxo/TgGX3rkLx
6cbaP6mxN7LcCv9dUj+eRnngQIBA6NAMd4wEQYDVROBAAoECERCRONSWVBUMBkGA1UdlwQSMBC
ADsL0GxMP29sL29/X39cXEMA4GA1UdEwEB/wQEAWIEMDANBqkqhkIG9w0BAQsFAAOCAQEAQAD8+BpnM/
K10m5iKO4BTcvGi77oSp0sul5utP6ZFVEuqjIMx4xxh+1SVRSMXQttd49/08BIUOaYrIjvP+dJT5J6D+UeFRSj
E6mevjRE1qqIpcCsHoCWVE9WkiuF3QLaouD9PH5FPsk53QVdd4Yp+vIKDvYQbpc12A8wAEu8hknqgxN2/E
pd8u/3LQgsTmY6mn5ek2EFNI6o2dQa503DpzoXGURIFOOaAWB7ZUJhe7nxG1Xjn1OnttreoP6twCTrG8IoL
fGKDKLVQ3G+7vAqXgB+JwFLUoULRaw0elSPgE7nMkz0la0lu+aJlaHRJZq/F9eMkDXFqRSMGHhNUijzrQ
Q==</dsig:X509Certificate>
```

```

</dsig:X509Data>
</dsig:KeyInfo>
<xenc:CipherData>
  <xenc:CipherValue>NoO3acdWBuH0nduRMdLZtOhN645erE7JjABu4waNg7GYbEvxjr6m1mdhjp32BZ4
  el+cp5r5UIhEYxt4sHlj8tMxJqcgxrJEiD6lg/zzc1mDV61fQidJWgclkWOUA2deb1Wz4OeS1oMe1F6M1wmvS
  PWNpSZbEmEG/4zv9ggGQWA=</xenc:CipherValue>
  </xenc:CipherData>
  </xenc:EncryptedKey>
</dsig:KeyInfo>
<xenc:CipherData>
  <xenc:CipherValue>279oO2AZ63oVYSFjMT+mW/YSUBbWJmNiGeg1OlendoFEhphHYcpTbGXG7kO/
  SXtewRr1aZp19L4/1qqNJW0BQeXluwVd7Qz9odekyzIGN4gBPuvp3ZH46yPT7kPfc3umwUIVZ5R3GPhEYG
  m4AUhK8Ygju/FZPZy+iU84bUG5iSV8ORYtEck+UQpGbbfCyYPX5U96zkV0hj6HL+Cwalg2Q9Gp6chu0lpBL
  /WRF0n1Kn1J/Z7LkVbKB9cROm3B2v77vJkjrhc4FwwT0wlkZZbq4yX5w7qeMlguAUZ2OdPDwhEGSj+TM6ai
  dGYddNDhQ0SDfbLKETWg8Lh6vuRhFk925uVUoiLJgw1MBf22t3H1Tz5z30leOJHdsEJt6XfHP0P2WUpSw1
  EAAqFq4RBl4hOgrzoRqVfZM9u1MPML8bYhLDI4O8cKCDvo4iwd1Mv0niu0veD7Asyo43nA4hVh/BsUskEZe
  /SaStOo+W/XzeL7cdJmeW9AkTkpsiidF8IEERY2iANdJpDNGS7B7jQ3uDZiVOVYBoCG2YrecMSrlcpfxzNap
  DJKUshyxRigPSGtN9YkMZsgc/whhkIVN1VnV2A541butR9SG3m8ITbm/fgcvt36DPXcmIV4cyHWTiX2lv9LF
  +IAmG+WVBnu6d4LMCXWMa4gzTnrqoXvXpAKNI51UmH6jiYOn1Or8Ls5YOzDeZRcVdgRFT1K2sHrZ4W
  WYeSUQfsxfZ09qzngTQ4BvAomPrL6Xh4IYAPMQxJZio1yo8/C7IyZiww/XAeNuZ4/tG7tF3vKi/B6mjHm9iqN6
  OtAuQ8iHCZt8ayBMhbgHb6vORxv77A1+OKxCKZQfcxyHesCnij0ONEvnoJi3Z/2TjupjAMYNgJ+O20Wt4Q8J
  hWGfuCOI13H4oy2Z/Wy+lsKqRyJ4ihlXrz0T5LsFwncdE7ho3MhyT2C3NAkrosgpVRIyB99DLJpWhWR4s3tx
  pfc47DRihny3gq7iepcl60ANTL1tBePk0PRD/y/tf59aZjhWSMBWQpwwJaVAHJgpBGtCUYkGXLakUICZ+cY4
  osHeTylivNshX4F/O8AvDjXvlQUKYss+Bz7i+raQMFvVQNp0Lr3BLc3FPAQCgZDFB9xuaVOu245hxtWakaq5
  ZC/jg09KkA+vWj13nkK0dY0YE3IDlxTIEVvuN61uIRJEBwh0JWU96fPycVfmhdvhlQF6lcWzvbsCKhAMZu83v
  0Wb941LsssZziKHa7Qj/2U4maTHgl03iZhdETQtVyvr1FS+PK86V62jDhy53VSF1bqP9eH/16bqLXSSyvyCE
  ILEO4LhZCbS33fUpKM0sej5gp7AQhFTA4kVrv0K6D9/7R0gR72GZF9/+HXwKq4VHg+s8380xen3F5O9ZWe
  2nX9y3yleH2Us/PoKjyYjKwV3DWYUioJtwv93uobzxAMLFhqFPX8+bHFBnyN1yqEjJFB5BZkX8Xa1G8Sp6ig
  kGtXZItX0IaxBzrWvbpyuZW2LU/1igkQkA20cuLFPrEax1gZzVZYBP9zEKdqCGRUGc+HFuirgtG0ncR+4KKbf
  VtN7r3xkVNfaihXVVeF45T1/TBE+mdjRd5kMICLEY29uT3IFZbZsPUoSSEa6wNLyTtSt7DSJiX4abhxUDx
  QeHgF0iLbKJw99h/OjopfOCpyTIlxglayLIT6TrDsiwHnxuwbRkeRmh7gXS79qBZlyBpi1g6ZMTB4FDfP0+1QI
  FzT5wmoOlzEZ1SgOT8JgGd5c/xelZ3NqLvlO4ZxefO6KV+I8ladOipV5utV3mOhgAl9xDzOnqYb7UZY9OcZR
  p9hWX6KxzokymQGwUqjAio/k3wqpWGBuCNGLd+piWUBkot2RRIQeVFeYjaa0hTsNmAmpbvgoj2HhERFfg
  RQk6/IU4XbBSHmSX3Va+NK56uDEBqPhEg57G6Ok1ozqXQPRJayob87GZaEzEgUDW12ieXYIjXXZlZ92le
  w04/2bhXc8dtQEyxOcNkFugfWEd7EeNyFULSAIjdsQfXzaouOKsWL3XzQ71+slIJKFU0KfKitxIFaukmwUfLx
  W/ZwGglQcOchs7E5Tu5ZGIl+ih5wOm1py+M+cgBk3wyFGlcjD9IJZ5oM9IQAJn78NwTQ8LG8VrFzAe0ObM
  7/8sWZbr+c1IMmDvAl3qAARray6rNTMamqgu0cc1hcdyEyn3CmZjy1x9VWbbNXddq/IXu6c6Kgda8rMrVyV9
  B9KYWutw9ht9u8bK9ny31zDnllzEm3sVaY4XzYbXqvX7ZpVvcC607KiKE7m9HWcuPXsqA7eFPbPrZl6Jue9
  F8OkZIIH4vsOY+VfV0ju9JufJSoAA+eHctUZY4c4biYPJlxa82THQ29XkE47X8eWF24cY6upGJ8Lh8H27wavM
  cuKJNYxMnJKKpdpMlMmBMgTZuyvyAzfyo/CiaSgms8WBMr2KUohdQ6jXonYEEhKFeJU2HFamPgjKI+Mcp
  +6a3as3L8l10JihBdgrM6vKe7fHc6dPhzNVz3ep8AYII/6sr01v5uAK42Oy9Q9TNxZsL47zecMd4CwRJdHQMS
  4DKfiHZBOLE3Cep7Ca5bKhYIkTK+nkCw5twkMSJdqzEqaVu0WPFwLQv4ysibrOPtN5E+alzWYQPnGB2kq
  3vH0mfqnxcp1aLxJbTR0u5MPWehfCem9i/IO/few4XnbplMMcf8YjzH1brpp+P667c9O9qVDAsj9KFVIVZlvp2Z
  0DU3pS49oZkU+vVQWOahFqXYYNcbW5kjhXwAjHfjHTjQnRQPcGqkjJ4fEiLvYOQ5yE1hW/LwNdua8NcC
  nzuDg6P7SpOmIZ8p3GENITvPRUxZvTcDK5efdsoVb8FXvuyfVxwUXmnZ9FCmdjwr57tt3iWb8S064LHJYO
  6X4HvcfR4VtPBvjVO5qCE57hdy+XYOvyT9bhCWLvdll354ni6L6lgxx0Y/fiBNZImT7FeVl91Plw6lW0qvzHR2
  Pkz+abzVqU5pNITyJrt20+5hezvwnL//VTlyZmBentiG6FL4y11NL+PWcAvVw86kQBikTvlSuMvaZwQ1kRlRw
  1gu/uEGcU5COacYwvKTuL/GMh10g5e0GYcYEdHctjTE5CCWCOYt/tLgNkeV2Q7QdY5Xk9Zsc3zS1Krrr4
  Dbs9Hk4WCWHhoM4ZUn35F08bs2p2N0bdeag1BQ1Wn/tHdJ3lZ2bpAqJGkZYj4BcEW+3oI9VmaQOI+eIGN
  Odv4lydIOtoz8bl50LqxaCMGIAu5BCQUHRZK0m95SvcLZocG+UCPKnrfVXkCPz20nhwMXh6ZPK/vBCoE
  Yf2ICSMJfpvEQRfKeM/j4SYSjm/C0C3QneeCtOjxmCrwqinKqrzkmMpNTYXZjUua3WtGj4H4nvwVHKGSL/
  ZSs6OifBO3YBFeyChqC58ZtYGBafFH+aaTdNqQmpd8X0U37rgtqfeg06Nst+QT9Ezj35SL7yjs+4+rvX2Rpyk
  FD3s4S2l+3aVBxqY5vTTIKP7iDZH1ADKjn7rSxeiuYhxqJnNzQyMMsgGmYbW3rnjlsTqKU/XD6twOcEb17K
  SII9+2Jp3MGjclPdYdWkSiobdamdU9Cj6mlhG5d1DThNajwdTxjVzP7aPsmjJtNaMPFRr2l62wSiUfTLsILJm
  pF9W98umQyWpfWXJ9yOx1zCR674X4JEiT6waP1cGV7ZGklmYbr/Q3XsfjJAMKZvqOpnVLPzZwOGVq4oQ

```


RGtMVEzRys3dkFwGdCK0p3RkxVb1VMUmF3MGVJU1BnRTduTWt6MGxhMGx1K2FKbGFIUkpacS9GO
WVNa0RYRnFSU01HSEhOVWlqenJRUT09PC9kc2InOlglMDIDZXJ0aWZpY2F0ZT48L2RzaWc6WDUwOU
RhdGE+PC9kc2InOktleUluZm8+PHhIbmM6Q2lwaGVyRGF0YT48eGVuYzpdXBoZlJWYXZlZT5Ob08zY
WNkV0J1SDBuZHVSTWRMwNRPae42NDVlckU3SmpBQnU0d2FOZzdHWWJFdnhyajZtMW1kaHBqMzJC
WjRISStjcDVyNVVJaEVZeHQ0c0hJajh0TXhKcWNvZ3hySkVpRDZJzy96emMxbURWNjFmUWlkSldnY0lrV0
9VQTJkZWlXV3o0T2VTMW9NZTFGNk0xd212U1BXTnBTWmJFbUVHLzR6dJlnZ0dRV0E9PC94ZW5jOkNpc
GhIclZhbHVIPjwveGVuYzpdXBoZlJWYXZlZT5Ob08zYWNkV0J1SDBuZHVSTWRMwNRPae42NDVlckU3SmpBQnU0d2FOZzdHWWJFdnhyajZtMW1kaHBqMzJC
PHhIbmM6Q2lwaGVyRGF0YT48eGVuYzpdXBoZlJWYXZlZT5Ob08zYWNkV0J1SDBuZHVSTWRMwNRPae42NDVlckU3SmpBQnU0d2FOZzdHWWJFdnhyajZtMW1kaHBqMzJC
VQmJXSm1OaUdlZzFPbGVuZG9GRWhocEhZY3BUYkdYRzdrTy9TWHRld1JyMWFacDE5TDQvMXFXTkpX
MEJRZVhSdXdWZDdRejlvZGVreXpJR040Z0JQdXZwM1pINDZ5UFQ3a1BmYzN1bXdVSZaNVlZr1BoRVI
HbTRBVWhLOFInanUvRlpQWnkraVU4NGJVRzVpU1Y4T1JZdEVjaytVUXBHYmJmQ3lZUFg1VTk2emtWM
GhqNkhMK0N3ZWxnMIE5R3A2Y2h1MElwQkwvV1JGMG4xS24xSi9aNoxrVmJLQjijUk9tM0lydcj3dkpranJoc
TRGd3dUMHdJa1paYnE0eVg1dzdxZU1sZ3ZBVVoyT2RQRHdoRUdTaitUTTzhaWRHWWrkTkRoUTBTRG
ZiTtEtfVfDnOExoNnZ1UmhGazkyNXVWVW9pTEpndzFNQmYyMnQzSDFUejV6MzBsZU9KSGRzRU0Nlh
mSFAwUDJXVXBTdzFFQUfXrNE0UkJsNGhPZ3J6b1JxVmZaTT11MU1QTUw4YlloTERJNE84Y0tDRHZvN
G13ZDFNdjBuaXUwdmVEN0FzeW80M25BNghWaC9Cc1Vza0VaZS9TYVN0T28rV1h6ZUw3Y2RkBVVXO
UFRvGtwc2lpZEY4aUVFUIkyaUFOZEpwRE5HUzdCN2pRM3VEWmIWT1ZZQm9DRzJZcmVjTVNyBGNwZn
h6TmFwREpLVXNoeXhSaWdmUFNHdE45WwTnWnNnYy93aGhrSVZOMVZuVjJBNTQxYnV0UjITRzNtOEI
UYm0vZmdjdNqZnkrQWGNtbFY0Y3IIV1RpWDJsdjIMRitJQW1HK1dWQm51NmQ0TE1DWFdNYTRnelRuc
mdvWHZYcEFLTkk1MVVtSDZqaVIPbjFPcjhMczVZT3pEZVpSY1ZkZ1JGVDFLMnNlclo0V1dZZVNVUWZze
GZabzlxem5nVFE0QnZBb21Qckw2WGg0SVIBUE1ReEpaaw8xeW84L0M3bHlaaXd2L1hBZU51WjQvdEc3
dEYzdktpL0I2bWpIbTlpcU42T3RBdVE4aUhdWnQ4YXICTWhiZ0hiNnZPUnh2NzdBMStPS3hDa1pRZmN4e
Uhlc0NuamowT05Fdm5vSmkzWi8yVgP1cGpBTVIOZ0orTzlwV3Q0UTHKaFdHZNVDt2wxM0g0b3kyWi9Xe
Stsc0txUnIKNGlobFhyejBUNUxzRnduY2RFN2hvM01oeVQyQzNOQWtyb3NncFZSSXICOTIETEpwV2hXUjR
zM3R4cGZjNDdEUmlobnkzZ3E3aWVwY2w2MEFOVEwxdEJIUGswUFJEL3kvdGY1OWFaamhXU01CV1Fw
dndKYVZBSEpncEJHdENVVWtHWExBa1VJQ1orY1k0b3NIZVR5SWI2TnNoWDRGL084QXZEalhb2bFFVS1
lzcYtCeJdpK3JhUU1GVnZRTnAwTHlzQkxjM0ZQQVFDZ1pERkl5eHvHvK91MjQ1aHh0V2FrYXE1WkMvamc
wOUtrQStWd2oxM25rSzBkWTBZRTNJREI4VEIFvNz1TjYxdUISSkVCd2gwSldVOTZmUHljVmZtaGR2aGxR
RjZsY1d6dmJzQ0toQU1adTgzdjBXyjk0MUxzc3NaekILSGE3UWovMIU0bWFUSGdJMDNpWmhkRVRrdFZ
5dnJkMUZTK1BLODZWNjJqRGh5NTNWU0YxYnFQOWVILzE2YnFMWFNTeXZ5Q0VsTEVPNExoWkNiUz
MzZlVwS00wc2VqNWdwN0FRaEZUQUTrVnJ2MEs2RDkvN1lwZ1I3MkdaRjKvK0hYd0txNFZlZytZODM4MH
hbjNGNU85WldlMm5YOXkzeUllSDJvcy9Qb0tqeVlqS3dWM0RXWVvPEp0dnc5M3VvYnp4QU1MRmhxRI
BYOCTiSEZibnlOMXlxRWpKRkl1QlprWDhYYTFHOFNwNmIna0d0WFpsdFgwSWF4QnpyV3ZicHI1WlcyTFU
vMWlNa1FrQTIwY3VMRiByRWF4MWdaelZaeUJQOXpFS2RxQ0dSVUdjK0hGdWlyZ3RHMG5jUis0S0tiZIZ0
TjdyM3hrVk5mYWloefhWZUY0NVQxL1RCRSttZGpSZDvrTUIDTEVZMj1VDNJRpIwNnQVW9TU09FYTZ3
TkxZdFFU3Q3RFNkaVg0YWJoeFVEeFFISGdGMGIMYktKdzk5aC9Pam9wZk9DUHIUSUI4Z2xheUxsVDZ
UckRzaXdlbnh1d2Jsa2VsbWg3Z1hTnzlxQlpJeUJwaTFnNlpNVEI0RkRmUDArMVfJRnpUNXdtb09sekVaM
VNnT1Q4SmdHZDVjL3hITFozTnFMdmxPNFp4ZWZPNktWK2w4SWFkt2lwVjV1dFYzbU9oZ0FJOXhEek9u
cVliN1VaeTIPY1pScDloV1g2S3h6b2t5b21RR3dVcWpBaW8vazN3cXB3R0J1Q05nTGQrcGIXVUJrb3QyUIJs
UWVWxRmVZamFRhMGhUc05tQW1wYnZnb2oySGhFukZmZ1JRazYvSVU0WGCJU0htU1gzVotErTks1NnVE
RUJXUGhFZzU3RzPazFvenFYUVBSSMf5b214NodaUY0V6RWdVRFcxMmlIWFJlJalHYWkl0tJZCX3cWNC8y
YmhYYzhkdFFFfeXhPY05rRnVnZldFZDDfZU55RIVMU0YfsamRzUWZYemFvdU9Lc1dMM1h6UJtZcxK3NJSU
prRIUwS2ZLaXR4bEZhdWtt1VmTHhXL1p3R2dsUWNPY2hzN0U1VHU1WkdpSStpaDV3T20xcHkrTstjZ0Jr
M3d5RkdsY2pEOUIKwJvVTTIJUUFqSm43OE53VFE4TEc4VnJGekFIME9iTTcvOHNXWmJyK2MxbE1tRHZ
BSTNxQUFScmF5NnJOVE1hbXFndTbjYzFoY2R5RXluM0NtWmp5MXg5VldiYk5YZHFrl2xYdTzjNktnZGE4
ck1yVniWOUi5S3IXdXR3OWh0OXU4Yks5bnkzMXpEbksekVtM3NwYVvK0WHPZYIhxdlg3WnBWdmNDNjA3
S2ILRTdtOUhXY3VQWHNqTdlRIBiUHJabDZKdWU5RjhPa1pJSUg0dnNPWStGvm8wanU5SnVmSINVQU
ErZUhjdFVaNGM0YmIZUEpJeGE4MIRIUVQ5WGtFNDdYOGVXRj1Y1k2dXBHsjhMYThIMjd3YXZNY3VLS
k5ZeE1uSkTLcGRwTWxtQk1nVFp1eXZ5QXpmb3IPL0NpYVnNz20zd2FuWWw1ODZGTmU5elhuRHF0c1lr
NFJUTnJCTmQyOGZxd3luZUxvZUVpM09tL2RPQjNXTEdRSGtDeWJ1cnozSEluOFpQT3lyZERNdVJRenM
4V0JNcjJLVW9oZFE2alhvbllFRWhLRmVKVXgySEZhbVBnaktS01jcCs2YTNhczNMOEkxMEppaEJkZ3JNN
nZLZTdmSGM2ZFBoek5WejNlcDhBWWxsLzZzcjAxdjV1QUs0Mk95OVE5VE54WnNMNDd6ZWNNDZRDd1
JKZEhRTVM0REtmaUhaQk9MRTNDZXA3Q2E1YktoWUlrVesrbmtDdzV0d2tNU0pkcXpFcfWfWdTBXUEZ3
TFF2NHlzaWJyT1B0TjVfK2FselZUVBuR0Iya2Nxm3ZIMG1mcW54Y1AxYUx4SkJ0UjB1NU1QV2VoZkNlB
TlpL2xPL2ZldzRYbmJwbE1NY2Y4WWp6SDFicnBwK1A2NjdjOU85cVZEQXNqOUtGvkIWWmxwdjJaMERV
M3BTNDivWmtVK3ZWUvPYWhGcVhZwU5DYlc1a0poWHdBakhGakhUaIFuUfQY0dxa2pKNGZFaUx2W
U9RNXIFMWhXL0x3TmR1YThOY0NuenVEZzZQN1NwT21JWjhwM0dFTkiUdIBSVXhadIRjREs1ZWZkc29W
YjhGWFZ1eWZ0Vnh3Vvhtblo5RkNtZGp3cjU3dHQzaVdiOFMwNjRMSEpZTzZYNEh2Y2ZSNFZ0UEJWAlZP

```

NXFDRTU3aGR5K1hZT3Z5SVQ5YmhDV0x2ZGxJMzU0bmk2TDZsZ3h4MFkvZmICTlpsbVQ3RmVWbDkxU
EI3NkIXMHF2ekhSMIBreithYnp2cVU1cE5sVHIKUnQyMCs1aGV6dnduTC8vVIRMeVptQkVOdGIHNkZMNHI
pMU5MK1BXy0F2Vnc4NmtRQmxrVHZsU3VNdmFad1Exa1JJcncxZ3UvdUVHY1U1Q09hY1I3dktUdUwvR0
1oMTBnNWUwR1ljWUVkSEN0aIRFNUNDV0NPWXQvdFRMZ05rZVYyUTdRZfK1WGs5WnNjM3pTMUtyd3I
ORGJzOUhrNFdDV0hob000WIVuMzVGMdhiczJwMk4wYmRIYWcxQIEV24vdEhkSjNsMlpicEFxSkdrWllqN
EJjRVcrM29JOVZtYVFPSStISudOT2R2NEI5ZGxPdG96OGJMNTBMcXhhQ01HSUF1NUJDUVVIUlpLT205
NVN2Y0xab2NHK1VDUEtucmZWWGtDUHoyMG5od01YaDZaUESvdkJDb0VZZjJJQ1NNSmZwdkVRUkZLZ
U0vajRTWVNqbS9DMEMzUW5IZUN0T2p4bUNyd3FpbktxcnprbU1wTIRZWFpqVXVhM1d0R2pxNEg0bnZ3
VkhLR1NML1pTczZPaWZCTzNZQkZFWWNlciUM1OFp0WUdCYWZGSCthYVRkTIFxTXBkOFgwVTM3cmd
0cWZIZzA2TIN0K1FUOUV6ajM1U0w3eWpZkZQrcnZYMIJweWtGRDNzNFMYSszYVZCeHFZNXZUUVGxLU
DdpRFpIMUFES2puN3JTeGVpdUpZaHhxSm5OeIF5TU1zZ0dtWWJXM3JuamxzVHFLVS9YRDZ0d09jRWlx
N0tTSUk5KzJKcDNr2pjTFBkWWRXS3Npb2JkYW1kVTIDajZtSWhHNWQxRFRoTmFqd2RUWGPwElA3Y
VBzbWpKdE5hTVBGUonlybDYyd1NpVUZ0TFNsTEptcEY5Vzk4dW1ReVdwZldYSjI5T3gxeKNSNjc0WDRKR
WIUNndhUDFjR1Y3WkdrbG1ZYnlvUTNYU2ZqSkFNS1p2cU9wbIZMcHpad09HVnE0b1FXOURSRkVMNnd5
M0F4MnlleXd5a2RxVWVBMIRGWTlhbzQ3WDNXQU85SXIOYVpQQVMYdmFKTElvWEhsYXZPSGMzcdNP
V1QzSUdwcTZuZ3NPMjFgB2tWblZEdkkrS2ZHPC94ZW5jOkNpcGhlcIzhbHVIPjwveGVuYzpdaxBoZXJEYX
RhPjwveGVuYzpdFbmNyeXB0ZWREYXRhPg==</mod:ApplicationRequest>
</cor:deleteFilein>
</soapenv:Body>
</soapenv:Envelope>

```

Signing the SOAP message (transport signature)

The last step is to create the transport signature on the SOAP message. The signature is based on a customer signing certificate, and the style of the signature is detached. This corresponds to step 8 on the list on page 4:

```

<soapenv:Envelope xmlns:mod="http://model.bxd.fi" xmlns:cor="http://bxd.fi/CorporateFileService"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sign="http://danskebank.dk/AGENA/SEPA/SigningService">
  <soapenv:Header>
    <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsu:Timestamp xml:id="Timestamp-5293c3c7-b69f-48e2-866e-9fc9cd35a422"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wsu:Created>2010-05-07T11:26:49Z</wsu:Created>
        <wsu:Expires>2010-05-07T11:31:49Z</wsu:Expires>
      </wsu:Timestamp>
      <wsse:BinarySecurityToken xml:id="SecurityToken-192f1eb9-6c10-482c-8719-
76754a128e84" EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-
security-1.0#Base64Binary" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-
token-profile-
1.0#X509v3">MIIDnzCCAoegAwIBAgIHDSQ4jQY0sTANBgkqhkiG9w0BAQsFAADCBWjEQMA4GA1UEAxMH
REJHQ0FEQjELMAkGA1UEBhMCRlEsbzEzZDktUdUwvR01oMTBnNWUwR1ljWUVkSEN0aIRFNUNDV0NPWXQvdFRMZ05rZVYyUTdRZfK1WGs5WnNjM3pTMUtyd3I
ORGJzOUhrNFdDV0hob000WIVuMzVGMdhiczJwMk4wYmRIYWcxQIEV24vdEhkSjNsMlpicEFxSkdrWllqN
EJjRVcrM29JOVZtYVFPSStISudOT2R2NEI5ZGxPdG96OGJMNTBMcXhhQ01HSUF1NUJDUVVIUlpLT205
NVN2Y0xab2NHK1VDUEtucmZWWGtDUHoyMG5od01YaDZaUESvdkJDb0VZZjJJQ1NNSmZwdkVRUkZLZ
U0vajRTWVNqbS9DMEMzUW5IZUN0T2p4bUNyd3FpbktxcnprbU1wTIRZWFpqVXVhM1d0R2pxNEg0bnZ3
VkhLR1NML1pTczZPaWZCTzNZQkZFWWNlciUM1OFp0WUdCYWZGSCthYVRkTIFxTXBkOFgwVTM3cmd
0cWZIZzA2TIN0K1FUOUV6ajM1U0w3eWpZkZQrcnZYMIJweWtGRDNzNFMYSszYVZCeHFZNXZUUVGxLU
DdpRFpIMUFES2puN3JTeGVpdUpZaHhxSm5OeIF5TU1zZ0dtWWJXM3JuamxzVHFLVS9YRDZ0d09jRWlx
N0tTSUk5KzJKcDNr2pjTFBkWWRXS3Npb2JkYW1kVTIDajZtSWhHNWQxRFRoTmFqd2RUWGPwElA3Y
VBzbWpKdE5hTVBGUonlybDYyd1NpVUZ0TFNsTEptcEY5Vzk4dW1ReVdwZldYSjI5T3gxeKNSNjc0WDRKR
WIUNndhUDFjR1Y3WkdrbG1ZYnlvUTNYU2ZqSkFNS1p2cU9wbIZMcHpad09HVnE0b1FXOURSRkVMNnd5
M0F4MnlleXd5a2RxVWVBMIRGWTlhbzQ3WDNXQU85SXIOYVpQQVMYdmFKTElvWEhsYXZPSGMzcdNP
V1QzSUdwcTZuZ3NPMjFgB2tWblZEdkkrS2ZHPC94ZW5jOkNpcGhlcIzhbHVIPjwveGVuYzpdaxBoZXJEYX
RhPjwveGVuYzpdFbmNyeXB0ZWREYXRhPg==</wsse:BinarySecurityToken>

```


NXJJRW5YjNWd01Sb3dHQVIEVIFRTEV4RkVZVzV6YTJVZ1FtRnVheUJJIY205MWNERNVINQIIHQTFVRUJ
STVBOakV4TWpZeU1qZ3hNVE13TURBeE1Ra3dCd1IEVIFRRUV3QXhDVEFIQmdOVkJDb1RBREVKTUFj
R0ExVUVEQk1BTVFRd0J3WURWUVFSRXdBd0hoY05NRGt4TURNd01USXdNREF3V2hjTk1URXhNRE13
TVRJD01EQXdXakNCeFRFUK1BOEdBMVVFQXhNSVJFSkhRMUpaVUZReEN6QUpCZ05WQkFZVEFrUkx
NUk13RVFZRFZRUUhFd3BEYjNCbGJtaGhaMIZ1TVJbD0RnWURWUVFJRXdKRVpXNXRZWEpyTVJvd0d
BWURWUVFLRXhGRVIXNXphMIVnUW1GdWF5QkhjbTkkY0RFYU1CZ0dBMVVFQ3hNUIJHRnVjMnRsSU
VKaGJtc2dSMOp2ZFhBeEdEQVdCZ05WQkFVVER6WXhNVEkyTWpJNE5EUXpNREF3TWpFsk1BY0dB
VMVVFQkNQU1Ra3dCd1IEVIFRcUV3QXhDVEFIQmdOVkJDb1RBREVKTUFjR0ExVUUVUk1BTUIHZE1BME
dDU3FHU0liM0RRRUJBUVVBQTRHTEFEQ0Jod0tCZ1FDcGFIMnROYkNtUFJuaTc0dDRwWHB6VXdzVVR
SQmVhelBwY2haSGpuK0hCcGZRQUg2NmtDek9jeExIT3JMNzFDRIIJajZCNFdyCvhaeDRTVWxkUXBLdTY
3dINoNVpzTHI5WFBabmlDbytEdFgyek9ZR0VpZ3pxNmhJeWxSNVhLbIhrWmZKU3hVl1RnR1gzcmtMeEY2
Y2JhUDZteE43TGNDdijlVWorZVJubmdRSUJBNk5BTUQ0d0VRWURWUjBPQkFvRUNFUkNSME5TV1ZC
VU1Ca0dBMVVksXsdRU01CQ0FEc0xHeE1QMjIzTDI5L1gzOWNYRU1BNEdBMVVkrHdFQI93UUVBd0IFT
URBTkJna3Foa2IHOXcwQkFRc0ZBQU9DQVFFQUQ4K0Jwbk0vSzEwbTVpS080QIRjkdKpNzdVU3Awc3Vs
NXV0UDZaRkV2ZXVxamxNeDR4eGgrMVNWUINNWF0dGQ0OS8wOEJVVU9hWXJJJanZQK2RKVDVKNk
QrVWVGUINqRTZtZXZqUkUxcXFJcENzSG9DV1ZFOVdraXVGM1FMYW91RDIQSDVGHUHNrNTRVmrRkN
FlwK3ZJS0RWVVFicGMxMkE4d0FFdThoa25ncXhOMi9FcGQ4dS8zTFFnc1RtWTZtbjVlajZFRk5JNm8yZF
FhNTAZRHB6am9YR1VSSUZPb0FhV0I3WIVKaGU3bnhHMVhqbJFPbnR0cmVvUDZ0d0NUckc4SW9MzkdL
RGtMViEzRys3dkFwGdCK0p3RkxVb1VMUMF3MGVJU1BnRTduTWt6MGxhMGx1K2FKbGFIUkpacS9GO
WVNa0RYRnFSU01HSEhOVWlqenJRUT09PC9kc2InOlgl1MDIDZXJ0aWZpY2F0ZT48L2RzaWc6WDUwOU
RhdGE+PC9kc2InOktleUluZm8+PHhIbmM6Q2IwaGVyRGF0YT48eGVuYzZpDaXBoZXJWYwX1ZT5CYXhDS
XUyWEVtWHJDSEdmQU1NSC8vVGxma1F5UXIzNnlQUmR0UTc4UTQvMjhtQ2k3Sih1S2hxN09EWFh5O
Wo0bDRVU0d0Y09TcTVsSU1QUFE3SnVnVknRaVhaVkrCCEVLNIRqN3VNL0NQULijb0Fmb3ZyMkw4QTB
telk3MDR0YzhDMVZlaXRaUDcyZVZrOUUs4aFN3ZTRXK25QaS8wUIVsaUxoMHBhcVNQdjg9PC94ZW5jOk
NpcGhlcilZhbHVlPjwveGVuYzZpDaXBoZXJJEYXRhPjwveGVuYzZpFbmNyeXB0ZWRLZkx+PC9kc2InOktleUluZ
m8+PHhIbmM6Q2IwaGVyRGF0YT48eGVuYzZpDaXBoZXJWYwX1ZT5MY3pPcHlrbHQ2bXV3dGFjSmodyMk9
mdlINMcZkMXV5bUZAra28xT2x3NExveG9hWjhZdkMvZm0wa04xOEViQUxtZU8xS2p0cTZ2VWp3MEc4SFI
qaGILanROUmpJSTJWbkRoTndZMGQ0cWU0QINONjNlBFFBZzZabTNmd003MjdyVss1WC9SZy9wSE9qU
DZwamFPenR1STZEY000Y1YzZmtBTnBGaGRFekhQWIQ5ek9ET0F6Y0wyRDdFam5hL0pnS1NybEVOde
ZscWd3NG1BOVB0c3UrQndWQnprQzA2U3N3VXRSTG9ZYnB1TzZL2RqMjc2S1Era05Nd1EzZ2FKTUvN
ZG0xcDBzRstkU0s2V1U0WHhrNTMzZkc1a21wRU9zWEtjBk1BanJIT25KdFVIUXRiamFLM2Vud2xIRjZpNEI
wQ0lvZmFqY0QyUFZBVmdlRXpxQ29DRmN1cEczVWlvR0EzNXRBRRTZ0emhjRHNzVUNEaWhFV2NUVINZ
VjFMMWOWR2RDYXhHUTBRdHRhZWRyTVU3OUZ0UXg3M2NjcfU0WnU0KzliNEVqRTRkanZMUXBJckd
VRnh3Z0I4NDI5Y0JzdmkrSVYvVG9haEZ3QXNwTHk3dlZaTUh6TnRGcm5uVGxUTFZ1ZVRmZmROZXZOT
XdaUGtKcFJwSUE0ZllmCDmWY2pEY2JqRmKvdlRsRmJJR3lzMnRIZHh4M0JPWkVX0ThFeHBYTFZorGd2
SDhMnzF5MDJaMVIbTCtRTDZxTFZrcDhXWnlROS9nSEo1dTVHZ1BiZfDsNWwhKb2w3dTV0aHRGdUROU
k95aStQTDhOUXZLeXFaa1RoNVFWYnFBbjVBK2QxMEdqeW4vZ0JHbUpvRkxRbHFqM0pCamJaNktYYU9
QSy8xSEMrMTkrNUZmQ1NmNnB0eG04RjISMxVNMzhEdHuVSiBVN1RKM3h1a2tiU1pZUGxQbTFKZnlKR
GFISiYrcHivMGVvUHNNDMGhDcFRleXJzWHpCbIRyTg5tOHfnbkkRmktLK2hfY01kQzRxSVdZeGJRreVOa
1NCNXIaWdDITXurbThndDNzYnZoT2ozOXUyK09HK2FLKzka1VuT3ZMYTJqUgTLsWw5SWEzTXRZemd
qbFdZSk1GQzdacWdwUnRNL0dCckFrYjRjXU0IzbtdlVIRYS3NwYzBXT3BaWTZHY1Y2Wf5CZt0N3Z
XUUIwQitDcHhZTEJXNVJHazU3TTNzY2gxeGfIbWwxVW4yvcVNaQWxIYS85SjY1QIZkMG1SaDR1ZtNpaW
RWdEZIWUFickMzQWNyNEM3WmlraVI2MW5LWEIpekWzRWk1cGZPcHo3YWNpTVk2dDBxQ3hjMIIQajR4
WUtWdU9FUHGzUFNDaEVOTStjQlpYMWxWYk52cC9ncTJ4YmdDWXZJemtnYmcxQWxvZitCnNhPVIz6cz
F3SnVOMk1IdnJtVDVMBiNuMU1HaVbZMFVWTDI2VTJYT043QjBMN090dis4SVBYL0Myb1NTanZCS1h4Yk
g3RFFENTNmWTV0N2INcHZydEFhVzI5cC9ySW5BYm9zN3R5SGpBRXAxeEU5bVdlSk96Vjc0VnpydDVla
Gs1dU9xYzk0Nm4rWERuR21RMDV0SUIqSW50QnhlWFVDY0VoNTRhTXB0SjdZnjBCTmlHSU1EUWpVR
XVXU0pZZEpHeHFvYjA4Z3lrNVB5UjE0RUJFOWN6S2kvZnpsdlR1enp4RDBwdXFZQnZnNIRyeVB0WVJpN
2s4Nk42ZIR0OU4wYUhHeWhZL25OVWcxndhZ2xwRThhdkFtREZIQjIzazRJNnZUMXc4OEdxeUpCaHRvb
GZOVHdnWDJjVgPjbjBTUdkRytzWXdnSmF3VkrpU09GUiB1WnVKaXIEUzd6d2J4Wm13WIE3MjhHNHJK
OVFEaFc2U3R5MXcybzVxL2RZdjJiS0hHaDhZb0NMb0c1RFZoTUp2TnJjUk9aczB4RHNKeIVRaVhDMWZ4
Tm91WIIZRjU5OHc2N2dMWnZTNFh0Y1VVZUI2UkNoMTNONjZsaFzVUEdZZHFFSEJ2V3lpaVZYbHVjSmJl
YkJKOGtZanzhMM242WTVtcjAxRDJLNxFLKQW9RMDI3QVRFL282bXY3U3BCQ2EyoE9QMGIllajg3b283SnB
jVmdrSDViZG9XOHBsNEVYY1BnbVZkTzCwTWgzTjAzSHNCc1NDY0dRWGFnQThmK0hod3JucnZWTEENR
REVaNStBczBsbDRFVFRvTi9uSGhHMkRUyU14ZkZWwK40YU02eW95SmVub0NmNk4vSXZsSi9pWHo1N
UtXMKFoZ3JMNk9TTFXfZkZorVvSciVdJjBcC9nanQyYjvtVeh2KytnTTdXQWFwRTB6QkR0R3NQYmFrZ
WN0VjNEVkpjZzFFZmFvNEw4WGV3N3c3L0U3MXJGOE4xTzFxd3ZLVzlwZ1J6TDZhekpaajhqRFZabW5IW
TNnc1IQckQ1Y3BRReHZFMS82dDczdVQ0UHVKRDhBQVF1V05hMlpaZ0QxbTZyMFZINEJDbXZERWVKZG

```

NLcmRNYmdsdDR1RIVGWFI2OXgwwWY4STZSb3N5ekpYcUpUQ21Ba0V3UUFwUEJ3bE50LzRtaVBlaVN
ISldWMjg0cUI4L3gzMTJVvNr1YTdjNm1GZEVmUzISazlyM0hSVTR6cXM3Z2RDczlUM3A5cDjPnGRINDIrc
zJBalZyNkVDWUVsOVZETTI1REZzaHI4WnJEVUNCWIBGcjFLbniNDBwR0I0d2JoeTRvNE5mVnlzZlHTMIJ
IRXNrSXA2SiloYUFLNUFHR01xbINyUVQ4dWdPbDRrcldXbWJvL0ZiQldxMmtuQVM5LzdicE9ZSE9XVHEy
M2RLT3dWT000TVZKVGlxckRITXNmOUhOdIp6dkNwckxJRGRFTWRFYUZXSWSGxbGdQMk8yQkd6ZUR0c
FdhR0cxbHh3N0g5aGdaR2sxbENnSFhSWHd2R25LVUZLUWhCblZxaXk5VEp5M1dZbmYxOGJQRnlvblE3
QjZuditpRjJEMFprZGExMEhGd0kyMFpnWWIzcFpOakxkeIVHOXdkT28xaWo1WkVDUlpDZTh2TzB6R2xZel
JTTC9ITEIYUFZjV3FaS1InbHBMVzd5R0JkdDVRROt6TIBpM1d3MjNkZEVdcWFyKzZ4SIFYZG1RRk9EWMd
6Nk9ZMnc5bEpaOXhVdG9nMHFNVINCM1lzREZsOXDnNHovSGJLUTFnK0VKRW5YV0J0V0ZWbWInenB6
VjZuR0RSbUVINIQRzF2bmFabmRYbnJZci9yc1JDcU0wVzB2Q2hrRjdDbXpwWWtDUHFWRWNLdnEvc2Tc
V0IUT1IFdENUQmZzbXJaQUtodDdCckdSZVNaalp6U0ZoSXNKM212UnV6MFJaakhPSnJFTDJMYIgzT1R1
YmZML0JBSXo2Z2xLdWZTLzRmdnR0cnpPzVBV0FROHVIRi9KaUprYXNVcWFnalhtZzVnakFkOHFTQmt
mMFY5cj3cVfMm1poZEhXZ2xzNXIJZE1yTm5oN2ZRc2hzbGhaYzVYTTd4QVlkRHVUOVoxMkRvK3pJY3p
EeDFiOThEV0dTVkpnADVGVUx4emwxalhheEJkRFICSzJ3WXBIMnVNcTVCQXovbk5SN0Nka3RFdGVTTE
4xWGFOL3ZSYnp5TIU1ckthRDV3VmFrQ0RiYmJvSXBZOUFEOLvQXIZajAwOethZTJ6SXNCcW5KNGhSV
TIIS0zZwUJWY0g5MStSZkZtVS9Cb0hZZkw2dUdCTHhHU29FK3J5QIVOMVICWDIVQkg0R2FHc2IGUTIFc
Gk0cFJQUFdkNjJ6SGhxRzd1QUh1bGfKtGNVSjZnWU9Yb292W1Q5S2xPa2ZEZW1QdkpZeU9GWHp5K3V
PejMwSGgvYkVGNjNna25JMUZwY1kwZ0pyRnl3MUFTU2JkVWwrSDBZUHdYnN0K1pOMjd1S3IyV0dSS
GvqUIBrTUOTnArVmRuTy9YQWdlbjZFRnd1N0xUMnpVd2JONeI2amVLYTIRM0s1ZWtoVmRyQXUvTHhq
SWFHVJGmS0RqMGg2cDNISFNvWHFpdE5BaUNXcFRTQXR2Q3BiNnJ5UmXkWkxvOG1sMTgvQkpYaWJ
DZU5DY3hESfjNEpMemtleWJoR1lzMUFnQTZEZUhuLzArYzFMNS9sZ3RhRHNbjRpV21TcTZnZsBCUml
pNWZsWHFWcGxUTVNVU0RhNDNVmJEEdkZiand2OGJCWk1yWjBwUkJ0Z2QwT1ZPdZFObnV1dUS3B
sTzl0d1NaRG1SKzBZOxhJNGVBejk0ZWhuTHd2SHU4UHFJb3Y1NHc3cnNUM3JWwXNRU052WVBiWDI5Q
1liRFNkbTFUSUx2Yk9sS3RFa3dLNVZzYXdh3NPRml3clcwVEpqa282MFB1YINQRWc2alZZaElwNkNidXp
EN1I3cm9YWitVOC83MDZycF2QnY5ZVdsdjhZK2JUdUtrVW1GUWkwZTNoOHdUVHVsUzFIRXd5eGt1am
E1aUd2RTRNZTcraEdaSlh0elFRZ3ICb3hsZFhHbVJ5RmVld29mOHk2VIRWSXRPU0dBSldJvMx6WnRXM0
Uyajc2WCsvNUIVWG95Zmt4bXZvWFdmdTBQOEwrazMwYkRGNlcweC84MEIzQ3Y1dUN4SnA3NIJnQUhk
RWZtZ2RUL0RqTnptzR5MTIuNIJUIi8vYVJ4ZEVFdhPbYbDNybDE5YWJXdXfVzJnJTkwxV0pvT1ZqZEZNd
U83N3pJWHNUdVVMsm1ZenAxdiitNNFhCcko4aGxwek9kTzFtMVhVM0JtWEo4T1krOGtscFhyT0FyRS9mS
kFjWjdHTUVJdVbMsfBVM3FOMEdpSTFaUWdDZG1vSDdichSL0NxoWhFdW04ZXVjYVlscW95UUUhvQ2h
ValJ5aldhNmVaOG1VUzY2ZmRrbWdyWTh5UDhNWIEyME1UUGdFdkdaalkwVTJKdWx3bTZ4Q1crSGs3Vn
NsbExsNkJIQ2piSHpIRC9sOUlidmIxYy9McnNRMWgxUEdyRGhvZWIIISnFpbnB1R0MvM3d3akR6dnFBRncv
RGM5MS81ajdEVitYdEhtaGxvV3V3VWVs5UmM5Z05pOTAzSHRIVERud2hETWprVVFazmW0cXBmVy80OU
M2VTVCaWtSQytCWVIVjVHR296Z21MMjBTMHNXeGJVYUV1SmZTN0pZNGw3WHRFTjVacUxYcVYxNW
Z2QXNVQ2Z1VYrS0IFZ3VRd28wbnZEU25kVWAyRE1XQXVIZGpqaklhQ3BKSDJpZIBFdnNXMWlpbldsSIJ
zNmhGazVxSVUxcFE0dkJUT2d3S09BOWZLcVl1TVRCL2E0YWERQ0pub1ZMVcya3dzTkk2MU5GZUJvQz
MrUGF4ZTY3b0IRUXBVSjJBRG96ejRmcEZ6cE8yUnU3ZkxvNGpSVIE9PTwveGVuYzpdaxBoZXJWYwX1Z
T48L3hlbmM6Q2lwaGVyRGF0YT48L3hlbmM6RW5jcnlwdGVkRGF0YT4=
</mod:ApplicationRequest>
</cor:uploadFilein>
</soapenv:Body>
</soapenv:Envelope>

```

The XML examples are distributed along with this document.

Appendix B: Web service URL

The URL of the web service is:

<https://businessws.danskebank.com/financialservice/edifileservice.asmx>

It is not possible to fetch the web service wsdl by appending 'wsdl' to the URL. Instead, the wsdl must be retrieved from the bank homepage.