Tullverket

PKI-based security concept

Technical FAQ

Version 1.0.0
2012-03-14

Please note that if were are differences between this FAQ and the technical specifications, the technical specifications shall apply

Questions about the checksum/hash value

What digest algorithm should be used?
SHA-256.

Which part of the edifact interchange should be used to calculate the hash?
UNH-UNT (inclusive).

Which encoding should be used for the checksum?
The checksum (hash value) should be hex encoded (not base64 encoded). There should be no whitespaces in the hex encoded value.

How do we calculate the checksum (hash value)?
It depends on which api you are using but may look something like this:

```
byte [] unhToUnt = ….
md = MessageDigest.getInstance(”SHA-256”);
md.update(unhToUnt, 0, unhToUnt.length);
byte [] checksum = md.digest();
```

Are there any command line tools for calculating the checksum?
OpenSSL could be used.

```
openssl dgst -sha256
```

Why do I get error Z80 – invalid checksum
Some common reasons are:

- You are not signing UNH-UNT
- You are not using the correct algorithm (SHA-256)
- The byte representation of the checksum is not hex encoded

Questions about the digital signature/encrypted checksum

What encryption algorithm should be used?
RSA. Often hashing and signing is done in one single step using an algorithm named SHA256WithRSA

Which data should be signed?
The byte representation of the checksum (not the hex encoded representation) should be signed.

How do we calculate the encrypted checksum (digital signature)?
Depending on your environment the signing is done in one or two steps.
In some environments (e.g. Java) it is the done in one step, in these cases you are signing the the content of UNH-UNT, e.g.

```
Signature sig = Signature.getInstance("SHA256WithRSA");
sig.initSign(signKey);
```

```
sig.update(unhToUntString, 0, unhToUntString.length);
byte [] bytes = sig.sign();
```

In other environments its done in two steps (e.g. Microsofts Crypto API). In these cases the hash is created using UNH-UNT and the signature is created using the byte representation of the hash, e.g.

```
bResult = CryptHashData(hHash, unhUntString, cbRead, 0 );
bResult = CryptSignHash(hHash, AT_SIGNATURE, NULL, 0,
pbSignature, &dwSigLen );
```

Which encoding should be used for the digital signature
The encrypted cryptographic checksum (digital signature) should be hex encoded, not base64 encoded. There should be no whitespaces in the hex encoded value.

We are not ready to send complete EDIFACT-files yet, but would like to have our signature implementation verified. If we send an encrypted checksum, could you verify it?
No, we need the entire EDIFACT interchange.

Which padding algorithm should be used
Padding according to RSASSA-PKCS1-v1_5. This is probably the default in your api.

Are there any command line tools for signing/verifying signatures?
OpenSSL could be used.

openssl rsautl –sign
openssl rsautl –verify

Why do I get Z81 Invalid digital signature (cryptographic checksum).
Make sure you are:

- signing with the correct private key
- signing the byte representation of the checksum (not the hex encoded checksum)
- hex encoding the cryptographic checksum
- using the correct algorithm (sha256withrsa)

Questions about the autack message

Are there any sample autack available?
http://www.tullverket.se/download/18.3dbb41be12fafdeeda880001205/example_interchange_CUSDEC_or_CUSRES_and_AUTACK_PKI.pdf

What is USH[2].0534 "Security Reference Number"?
Security Reference Number is an unique internal reference number from security header (USH) to find corresponding USY-segments. It is basically a counter, SRN01, SRN02, SRN03 generated by the sender, however only one value (e.g. SRN01) will be used since only one signature is supported

What is USH[2].0511 "Security party identification"?
Security party identification is your eori-number (including the country prefix).
It is not the certificate serial number and it should not be be base64- or hex-encoded.

What is USC[4].0536 "Certificate reference"?
It is the certificate serial number, that is the base 64 encoded byte representation of the serial number of your certificate. A common error is to use the hex encoded representation shown in some certificate viewers.
Another common error is to use the eori number from subject distinguished name. (SERIALNUMBER=, OU=, O=,C=,CN=..)

What is USC[4].0538 "Key name"?
It is an base64 encoded byte representation of the key identifier in the authority key identifier field of the certificate. This is used by swedish customs to determine which issuer your certificate should be validated against.
For example the base 64 representation of the authority key identifier id issued by "CN=Swedish Customs Public CA 1.0, SERIALNUMBER=SE2021000969, OU=Only for authorized use, OU=Public Intermediate Certificate Authority, OU=Swedish Customs, O=Tullverket, C=SE" will look like:
PS1IRSLAUC0cmps4EQMSDUL77f8=

Why do we need certificate serial number, authority key identifier and checksum? Shouldn't digital signature be enough?
The validation is performed by the following steps:

1. Calculating the message checksum (SHA-256). This checksum is also compared with the unencrypted checksum in the autack.
2. Retrieving the public key for the certificate indicated by the serial number, issued by the authority indicated by authority key identifier.
3. Decrypting the digital signature with the public key indicated in certificate serial number issued by indicated authority key identifier.
4. Comparing the decrypted digital signature with the message checksum.

Questions about the certificate serial number

How do we read the certificate serial number from our certificate?
Exactly how it is done depends on which api you are using, using, but it may look something like this in Java:

```
BigInteger serial = x509Certificate.getSerialNumber().
new Base64Encoder().encode(serial.toByteArray());
```

or something like this in C#

```
System.Convert.ToBase64String(X509Certificate.GetSerialNumber());
```

The certificate serial number is not the same as SERIALNUMBER the subject distinguished name field.

Which encoding should be used for the certificate serial number.
The byte representation of the serial number should be base 64 encoded. A common error is to use hex encoding instead of base64 encoding since the serial number is shown as hex in most certificate viewers. In Unix a hex encoded value may be changed to base64 using:

```
echo "00 97 95 45 e3 c4 c2 59 ef " | xxd -r -p | openssl base64
AJeVRePEwlnv
```

Questions about the authority key identifier

What is the authority key identifier field and how is it used?

The certificate field authorityKeyIdentifier is used to uniquely identify the parent certificate used to sign a certificate. The authorityKeyIdentifier field in a certificate is the same as the subjectKeyIdentifier field of the issuer certificate. This field is divided into the following subfields (see RFC 5280):

- keyIdentifier - Contains a checksum of the parent certificate's public key.
- authorityCertIssuer - Issuer of the parent certificate
- authorityCertSerialNumber - Certificate serial number for parent certificate

The subfield keyIdentifier is used for signature certificates issued by Swedish Customs. To verify the signature, the recipient must have access to the sender's signature certificate. In the Swedish Customs' implementation of the EDIFACT format, no signature certificate is included in the AUTACK message. However, the AUTACK message includes certificate serial number, and a reference to the CA, authorityKeyIdentifier [keyIdentifier]. The fields serialNumber and authorityKeyIdentifier [keyIdentifier] uniquely identify the certificate used to sign the document which is necessarily to select the correct certificate.
See also section 4.2.1.1 of http://www.rfc-editor.org/rfc/rfc5280.txt

How do we read authority key identifier from our certificate?

Read the keyIdentifier for AuthorityKeyIdentifier from the certificate issued to your company by Swedish customs and base 64 encode this value. Authority key identifier is found in extension 2.5.29.35. Exactly how this is done depends on which api you are using, using, but it may look something like this:

```
byte[] extvalue =
x509Certificate.getExtensionValue("2.5.29.35");
DEROctetString oct = (DEROctetString) (new ASN1InputStream(new
ByteArrayInputStream(extvalue)).readObject());
AuthorityKeyIdentifier keyId = new
AuthorityKeyIdentifier((ASN1Sequence) new ASN1InputStream(new
ByteArrayInputStream(oct.getOctets())).readObject());
base64Encode(keyId.getKeyIdentifier());
```

Which encoding should be used for the authority key identifier id?

The byte representation of the authority key identifier id should be base 64 encoded. A common error is to use hex encoding instead of base64 encoding since the authority key identifier is shown as hex in most certificate viewers.
In Unix a hex encoded value may be converted to base64 using:

```
echo "3d 2d 48 45 22 c0 50 2d 1c 9a 9b 38 11 03 12 0d 42 fb ed
ff" | xxd -r -p | openssl base64
PS1IRSLAUC0cmps4EQMSDUL77f8=
```

Questions about documentation

Where do I find documentation for the PKI-based security concept?

You will find "Guidelines and Instructions on Security for Electronic Data Interchange (EDI)" in the URL
http://www.tullverket.se/download/18.2bfaa48c13036155bb680003360/Riktlinjer_sakerhet_EDI_version_2.0+EN.pdf

Where do I find the Swedish customs public certificates, issuer certificates and root certificates?

http://www.tullverket.se/download/18.293aa46d1353a39251fb4/edi_nya_rot_och_ca_certifikat.pdf

Where do I find the autack specifications?
http://www.tullverket.se/innehallao/t/tulldataregelverkspec/tulldataregelverktekniskaspecifikationer/sctssc.4.7f03eba412c3a2572d080001122.html