designing e-health

Generieren von selbstsignierten Software-Zertifikaten

Version 3.0

Wien, Oktober 2023

Inhaltsverzeichnis

1 Generieren von selbstsignierten SW-Zertifikaten 3
2 Generieren eines RSA-Schlüsselpaars, eines (selbstsignierten) Zertifikats und einer PKCS12-Datei 4
2.1 Generieren des privaten Schlüssels und des selbstsignierten Zertifikats (enthält den öffentlichen Schlüssel)....4
2.2 Generieren der PKCS12-Datei (optional) 5
3 Beispiel openssl.cnf 6

1 Generieren von selbstsignierten SW-Zertifikaten

In Krankenanstalten und bei deren Softwareherstellern können zur Authentifikation von SOAP-Requests spezielle, von der Krankenanstalt selbst signierte Softwarezertifikate ausgestellt werden.
Genauere Informationen zu den Anforderungen an die Zertifikate sowie den genauen Prozess zur Registrierung von Software-Zertifikaten finden Sie auf www.chipkarte.at im Bereich „Software-Zertifikate“.

Inhalt dieser Beschreibung ist das Generieren der Zertifikate mit dem OpenSource Tool *OPENSSL*.

Im Zuge der Ausstellung des Zertifikats (sofern keine eigene OPENSSL-Konfiguration eingesetzt wird) wird von OPENSSL interaktiv eine Angabe zu

- „Organization Name“
- „Organizational Unit Name“
- „Common Name“

erfordert.
Die Angabe dieser Daten ist optional und wird technisch nicht geprüft. Aus Sicht der Nachvollziehbarkeit und Zuordnung sind folgende Grundinformationen anzugeben:

- Name der ausstellenden Krankenanstalt (zum Beispiel im Parameter „O“)
- Haupt-Vertragspartnernummer (zum Beispiel im Parameter „CN“)

Hinweis: Es dürfen keine Sonderzeichen und Umlaute verwendet werden.

2 Generieren eines RSA-Schlüsselpaars, eines (selbstsignierten) Zertifikats und einer PKCS12-Datei

2.1 Generieren des privaten Schlüssels und des selbstsignierten Zertifikats (enthält den öffentlichen Schlüssel)

Die Variable <*VPNR*> ist durch die Vertragspartnernummer der Admin-Karte bzw. die entsprechende Haupt-VPNR der Krankenanstalt, für die das Zertifikat ausgestellt werden soll, zu ersetzen.

Hinweis: Die Gültigkeitsdauer kann z.B. mit 365 Tagen (1 Jahr) festgelegt werden.

openssl req -x509 -newkey rsa:2048 *–config <openssl.cnf>* -keyout <vpnr>.key -out <vpnr>.cer -days 365

```
Generating a 2048 bit RSA private key
.....................................+++
.......+++
writing new private key to '***.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AT
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Krankenanstalt XYZ
Organizational Unit Name (eg, section) []:TestSupport
Common Name (eg, YOUR name) []:testrsa
Email Address []:
```

Ein Beispiel für <openssl.cnf> ist im Anhang unter Kapitel *3 Beispiel openssl.cnf* zu finden.

2.2 Generieren der PKCS12-Datei (optional)

Diese enthält sowohl den privaten wie auch den öffentlichen Schlüssel als sogenanntes Schlüsselpaar. Diese kann unkompliziert in Java-Keystores eingespielt werden.

openssl pkcs12 -export -inkey <vpnr>.key -in <vpnr>.cer -out <vpnr>.p12

```
Enter pass phrase for testrsa.key:
Enter Export Password:
Verifying - Enter Export Password:
```

Im Verzeichnis befinden sich anschließend folgenden Dateien:

<vpnr>.p12	PKCS12-Datei zum Import in einen Keystore
<vpnr>.cer	das (selbstsignierte) Zertifikat nach PKCS7 (in diesem findet sich der öffentliche Schlüssel)
<vpnr>.key	der private Schlüssel

Zum Einspielen im Rechenzentrum des e-card-Systems wird das **<vpnr>.cer File** benötigt.
Dieses muss als Zip-File verpackt über das Web-Formular eingemeldet werden.

3 Beispiel openssl.cnf

```
#
# OpenSSL configuration file for
# Certification by SVC
# Version 1.8
#
RANDFILE               = ./random.rnd
[ req ]
default_bits           = 2048
default_keyfile        = ./Test.key
distinguished_name     = req_distinguished_name
attributes             = req_attributes
x509_extensions        = v3_ca
default_md             = sha256

string_mask            = nombstr

[ req_distinguished_name ]
 C   = AT
 ST  = Bundesland
 L   = Stadt
 O   = Verbund
 OU  = VPNR_Vertragspartnernummer
 CN  = Krankenanstaltsbezeichnung



[ req_attributes ]

[ v3_ca ]
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid:always,issuer:always
basicConstraints       = CA:false
keyUsage               = nonRepudiation, digitalSignature, keyEncipherment
```