Kurzanleitung zur Schlüsselgenerierung mit OpenSSL

Im Folgenden wird bespielhaft beschrieben, wie mit Hilfe des Kommandozeilen-Tools OpenSSL ein valider Zertifikatsrequests mit Schlüssellänge 4096 Bit und ein SHA256-Hashwert vom public-Key im Zertifikatsrequest erzeugt werden kann.
Die Konfiguration mit den Angaben zu "Unternehmen", "IK/BN" und "Ansprechpartner" sowie ggf. auch der "TrustCenter Name" sind entsprechend dem Antrag anzupassen.
Die optionalen Zahlen bei den Key-Dateinamen entsprechen dem IK für eine Zuordnung bei mehreren Schlüsseldateien.
Für die beschriebene Vorgehensweise ist die Version 3.0.x (oder neuer) von OpenSSL erforderlich. Bitte beachten Sie, dass wir für OpenSSL keinen Support anbieten und wir diese Kurzanleitung freundlicherweise ohne Gewähr bereitstellen.

1. Privaten Schlüssel generieren:
```
openssl genrsa -out 123456789.private.key 4096
```

2. Zertifikatsrequest erstellen:
```
openssl req -new -key 123456789.private.key -sha256 -sigopt rsa_padding_mode:pss
-sigopt rsa_pss_saltlen:32 -out 123456789.p10 -config itsg.config -outform DER
```

Die Konfigurationdatei "itsg.config" enthält folgenden Inhalt mit Feldbezeichnungen zum Verständnis und Beispielangaben. Die fünf Feldbezeichnungen sind fett markiert und die rot markierten Beispielangaben sind entsprechend den Antragsdaten anzupassen:

```
[ req ]
default_bits       = 4096
distinguished_name = req_DN
string_mask        = nombstr

[ req_DN ]
countryName                     = "1. Country Name  (Wert DE)"
countryName_default             = "DE"
countryName_min                 = 2
countryName_max                 = 2
0.organizationName              = "2. Organization  (TrustCenter Name)"
0.organizationName_default      = "ITSG TrustCenter fuer sonstige
                                  Leistungserbringer"
0.organizationalUnitName        = "3. Organizational Unit Name (Unternehmen)"
0.organizationalUnitName_default  = "Muster GmbH"
1.organizationalUnitName        = "4. Organizational Unit Name (Betriebs-Nr.)"
1.organizationalUnitName_max    = 80
1.organizationalUnitName_default  = "IK123456789"
commonName                      = "5. Common Name (Kontaktperson)"
commonName_max                  = 80
commonName_default              = "Max Mustermann"
```

Wichtiger Hinweis:
Im Feld IK müssen die Zeichen IK vor den Zahlen angegegeben werden, z.B. IK123456789.
Für die Felder Unternehmen und Kontaktperson sind nur folgende Zeichen zulässig: A-Z, a-z, 0-9, /, -, Punkt, ( ), Leerzeichen.
Umlaute müssen ersetzt werden, z.B. ö in oe. Alternativ zum Sonderzeichen & kann "und" angegeben werden.

3. Öffentlichen Schlüssel aus dem Request extrahieren:
```
openssl req -in 123456789.p10 -inform DER -pubkey -noout -out 123456789.pub.key
```

4. Werte des öffentlichen Schlüssels (ohne Algorithmus-Angabe) extrahieren:
openssl asn1parse -in 123456789.pub.key -strparse 19 -out 12345678pubKeyValues.key

Die Option "-strparse 19" extrahiert die Schüsselangaben (BIT STRING an Position 19) aus dem ASN1:

```
openssl asn1parse -in 123456789.pub.key
    0:d=0  hl=4 l= 546 cons: SEQUENCE
    4:d=1  hl=2 l=  13 cons: SEQUENCE
    6:d=2  hl=2 l=   9 prim: OBJECT            :rsaEncryption
   17:d=2  hl=2 l=   0 prim: NULL
   19:d=1  hl=4 l= 527 prim: BIT STRING
```

5. Folgende Befehlszeile gibt den SHA256-Hashwert des öffentlichen Schlüssels in einer txt-Datei aus:
openssl dgst -sha256 -out 12345678pubkey_sha256.txt 12345678pubKeyValues.key

Zusatzinfo zur Überprüfung des Zertifikatsrequests:
Folgender Befehl kann verwendet werden, um den Inhalt eines Zertifikatsrequests textuell auszugeben und zu prüfen:

```
openssl req -text -config itsg.config -in 123456789.p10 -nameopt multiline -noout

Certificate Request:
    Data:
        Version: 1 (0x0)
        Subject:
            countryName               = DE
            organizationName          = ITSG TrustCenter fuer sonstige
                                        Leistungserbringer
            organizationalUnitName    = Muster GmbH
            organizationalUnitName    = IK123456789
            commonName                = Max Mustermann
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (4096 bit)
                Modulus:
                    00:b4:4c:91:49:a8:45:c0:df:af:d5:76:6a:25:3f:
                    0f:24:23:a3:5d:ff:f7:06:b8:e4:1f:4d:59:91:2a:
                    bd:65:ad:aa:74:30:17:df:c1:8d:07:d1:81:56:a3:
                    83:9e:22:82:92:2d:...
                Exponent: 65537 (0x10001)
        Attributes:
            a0:00
    Signature Algorithm: rsassaPss
         Hash Algorithm: sha256
         Mask Algorithm: mgf1 with sha256
         Salt Length: 0x20
         Trailer Field: 0xBC (default)

         7d:3b:22:7c:aa:ba:d4:6e:0e:72:d5:fb:25:c9:8f:84:cc:5c:
         94:16:ce:fa:7e:22:89:64:90:88:d8:b1:ba:73:9d:8c:ad:aa:
         3f:e7:f2:cc:4e:d1:dd:34:a9:7a:ec:07:98:fc:53:62:95:19:
         0e:c9:f3:d2:ea:9c:0a:e5:94:d3:1d:d7:5e:15:9d:4d:f1:f2:
         9f:54:44:a5:41:7c:...
```