# 1 X.509-Zertifikate mit OpenSSL erstellen

Dokument-ID: 108395_de_00
Dokument-Bezeichnung: AH DE X.509 CERT OPENSSL


Stellen Sie sicher, dass Sie immer mit der aktuellen Dokumentation arbeiten.
Diese steht unter der Adresse phoenixcontact.net/products zum Download bereit.

**Inhalt dieses Dokuments**

In diesem Kapitel wird die Erstellung von X.509-Zertifikaten mit dem Tool *OpenSSL* erläutert.



## 1.1 Einleitung

Die Registrierung von Zertifikaten erfordert eine Zertifizierungsstelle (Certification Authority; CA), die für einen bestimmten Zeitraum Public-Key-Zertifikate ausstellt. Eine CA kann eine private (interne) CA sein, die von Ihrer eigenen Organisation geführt wird, oder eine öffentliche CA. Eine öffentliche CA wird durch einen Drittanbieter geführt, dem Sie die Validierung der Identität der einzelnen Clients und Server, denen er ein Zertifikat ausstellt, anvertrauen.

Es stehen mehrere Tools zur Erstellung und Verwaltung von Zertifikaten zur Verfügung, wie z. B. *Microsoft Certification Authority (CA) Server*, *OpenSSL* und *XCA*.

Dieser Anwenderhinweis erläutert die Vorgehensweise zur Erstellung von X.509-Zertifikaten mit den Tools **OpenSSL** und **XCA**, um eine VPN-Verbindung mit den X.509-Zertifikaten als Authentifizierungsmethode einzurichten.

Dieses Dokument ist aufgrund des Umfangs nicht als vollständiges Benutzerhandbuch für die beschriebenen Tools geeignet. Dieses Dokument soll Ihnen helfen, mit den Tools vertraut zu werden und die benötigten Zertifikate in einem kurzen Zeitraum zu erstellen.

### 1.1.1 Einführung OpenSSL

OpenSSL ist für mehrere Plattformen erhältlich (Linux, UNIX, Windows) und kann im Internet heruntergeladen werden. Wir haben in diesem Fall *OpenSSL 1.1.0e* auf einer *Windows 7* Plattform verwendet. Weiterführende Informationen zu OpenSSL und die unterstützten Kommandozeilen-Optionen sind unter http://www.openssl.org zu finden.

OpenSSL bietet zahlreiche Möglichkeiten zur Festlegung der erforderlichen Optionen. Sie können sie in der Kommandozeile eingeben, sie in einer Konfigurationsdatei festlegen oder sie bei Aufforderung in einem bestimmten Fenster eingeben, wenn der Befehl *openssl* ausgeführt wird. Bei der Verwendung von Konfigurationsdateien können Sie entweder alle erforderlichen Parameter in einer Einzeldatei festlegen oder verschiedene Dateien verwenden, je nachdem, welche Art von Zertifikat Sie erstellen möchten. Die OpenSSL-Konfigurationsdatei, die bereits in OpenSSL vorhanden ist, hat die Bezeichnung *openssl.cnf*.

Bitte beachten: In Windows wird die Dateiendung *.cnf* ausgeblendet, selbst wenn Sie die Einstellung im *Windows Explorer* geändert haben sollten. Aus diesem Grund verwenden wir die Endung *.conf*.

In den folgenden Kapiteln werden wir erläutern, auf welche Weise OpenSSL eingerichtet werden muss, um die Funktion einer Zertifizierungsstelle (CA) zu erfüllen. Eine Zertifikatanfrage muss durch die CA signiert werden, um zu einem gültigen Zertifikat zu werden.

Sie können zum Erstellen der Zertifikate prinzipiell die Beispiele in den folgende Kapiteln anwenden. Sie müssen dazu lediglich die Anweisungen befolgen und die Parameter im Abschnitt *req_dn* der OpenSSL-Konfigurationsdatei *openssl.conf* (siehe Kapitel „OpenSSL-Konfigurationsdatei modifizieren“ auf Seite 4) an die Anforderungen Ihres Unternehmens entsprechend anpassen.

Es folgt eine kleine Legende mit **Dateiendungen**, die für die erstellten Dateien verwendet werden, sowie deren Bedeutung.

| Dateiendung | Erläuterung |
|---|---|
| **key** | Privater Schlüssel<br><br>Bei diesen Dateien müssen restriktive Berechtigungen gesetzt werden. |
| **csr** | Zertifikatanfrage (certificate request)<br><br>Die Anfrage wird durch die CA signiert, um das Zertifikat zu erstellen. Im Anschluss wird diese Datei nicht mehr benötigt und kann gelöscht werden. |
| **crt** | Zertifikat<br><br>Dieses Zertifikat kann öffentlich verbreitet werden. |
| **p12** | PKCS#12-Export des Zertifikats, der den zugehörigen privaten und öffentlichen Schlüssel enthält.<br><br>Die Exportdatei wird durch ein Passwort geschützt, um den privaten Schlüssel vor unbefugter Nutzung zu schützen.<br><br>Dieses Zertifikat **darf nicht** öffentlich verbreitet werden. |

## 1.2 CA-Umgebung vorbereiten

Als Erstes muss eine Verzeichnisstruktur erstellt werden, in der sämtliche Zertifikatsangelegenheiten verwaltet werden. In den folgenden Beispielen wird **C:\CA** als Root-Verzeichnis verwendet. Folgende Unterverzeichnisse müssen erstellt werden:

| Unterverzeichnis | Zweck |
|---|---|
| **.\certs** | Verzeichnis, in dem die Zertifikate abgelegt werden. |
| **.\newcerts** | Verzeichnis, in dem OpenSSL die erstellten Zertifikate im PEM-Format als *<cert serial number>.pem* (z. B. *07.pem*) ablegt. Dieses Verzeichnis wird von OpenSSL benötigt. |
| **.\private** | Verzeichnis zur Speicherung der privaten Schlüssel. Stellen Sie sicher, dass Sie bei diesem Verzeichnis restriktive Berechtigungen festlegen, sodass für Anwender mit den entsprechenden Privilegien ein Schreibschutz (*Read-Only*) besteht. |

Neben dem Verzeichnisbaum müssen die folgenden zwei Dateien (*index.txt* und *serial*) erstellt werden:

- **index.txt**: Diese Datei wird von OpenSSL als Zertifikate-"Datenbank" verwendet. Um die diese Datei zu erstellen, gehen Sie wie folgt vor:
  - Öffnen Sie eine DOS-Eingabeaufforderung.
  - Wechseln Sie in das CA-Root-Verzeichnis (in unserem Beispiel: *C:\CA*).
  - Führen Sie folgenden Befehl aus: *copy NUL: index.txt*
    Dieser Befehl erstellt die leere Datei *index.txt*.
- **serial**: Diese Datei enthält den Zähler (*Counter*) für Zertifikatseriennummern. Dieser Zähler zählt durch OpenSSL automatisch hoch, wenn der entsprechende Wert zum Erstellen eines Zertifikats verwendet wurde. Um diese Datei zu erstellen, gehen Sie wie folgt vor:
  - Öffnen Sie eine DOS-Eingabeaufforderung.
  - Wechseln Sie in das CA-Root-Verzeichnis (in unserem Beispiel: *C:\CA*).
  - Führen Sie folgenden Befehl aus: *echo 0001 > serial*
    Dieser Befehl erstellt die Datei *serial* mit der anfänglichen Seriennummer 0001.

## 1.3 OpenSSL-Konfigurationsdatei modifizieren

Wir haben die OpenSSL-Konfigurationsdatei mit *openssl.conf* benannt und sie im CA-Root-Verzeichnis (in unserem Beispiel: *C:\CA*) abgelegt. Die OpenSSL-Konfigurationsdatei besteht aus mehreren Abschnitten. Jeder Abschnitt wird für einen anderen Zweck verwendet. Die Abschnitte umfassen die folgenden Positionen:

- **ca, CA_default**: Legt die Konfiguration der Zertifizierungsstelle fest.
- **policy_any**: Definiert die Richtlinien für Anfragen.
- **req, req_dn**: Definiert die Anfrage-Standardwerte.

In unserem Beispiel weist die Konfigurationdatei (*openssl.conf*) die folgenden Einträge auf:

```
[ req ]
prompt                          = yes
default_bits                    = 4096
distinguished_name              = req_dn
x509_extensions                 = req_ext
string_mask                     = utf8only

[ ca ]
default_ca                      = CA_default

[ CA_default ]
dir                             = C:/CA
certs                           = $dir/certs
database                        = $dir/index.txt
new_certs_dir                   = $dir/newcerts

certificate                     = $dir/certs/ca.crt
serial                          = $dir/serial
private_key                     = $dir/private/ca.key

default_md                      = sha256
default_days                    = 365

x509_extensions                 = req_ext
policy                          = policy_any

[ req_dn ]
countryName                     = Länderkennung (2-stelliger Code)
countryName_default             = DE

organizationName                = Name der Organisation (Unternehmen)
organizationName_default        = PHOENIX CONTACT Cyber Security AG

organizationalUnitName          = Name der Organisationseinheit (Abteilung, Division)
organizationalUnitName_default  = Support

commonName                      = Common-Name (Hostname, IP oder Ihr Name)

# In unserem Beispiel nicht verwendet
#emailAddress                   = E-Mail-Adresse
#localityName                   = Name der Örtlichkeit (Stadt, Verwaltungsbezirk)
#stateOrProvinceName            = Name des Bundeslandes/Bundesstaats (vollständiger Name)

[ policy_any ]
countryName                     = supplied
organizationName                = supplied
organizationalUnitName          = optional
commonName                      = supplied
# In unserem Beispiel nicht verwendet
#emailAddress                   = optional
#localityName                   = optional
#stateOrProvinceName            = optional

[ req_ext ]
basicConstraints                = critical, CA:false

[ ca_ext ]
basicConstraints                = critical, CA:true, pathlen:0
keyUsage                        = critical, cRLSign, keyCertSign
```

| Abschnitt | Option | Beschreibung |
|---|---|---|
| **[ req ]** | Dieser Abschnitt wird bei Anfrage nach einem Zertifikat abgerufen, indem der Befehl *openssl* mit der Option **req** aufgerufen wird. | |
| | **prompt** | Wenn dieser Wert auf **no** gesetzt wird, werden die Eingabeaufforderung für die Zertifikatsfelder deaktiviert und nur Werte aus der Konfigurationsdatei direkt übernommen. Sie müssen diese Option aktivieren, um in der Lage sein zu können, den *common name* einzugeben oder die Standardwerte des eindeutigen Namens des Zertifikats für jedes angefragte Zertifikat ändern zu können. |
| | **default_bits** | Dieser Eintrag legt die standardmäßige Schlüsselgröße in Bits fest. Bei einer fehlenden Angabe werden 512 Bits verwendet. |
| | **distinguished_name** | Dies bezeichnet den Abschnitt, der die eindeutigen Namensfelder enthält, die bei der Generierung eines Zertifikats oder einer Zertifikatanfrage per Eingabeaufforderung angezeigt werden. In unserem Beispiel wurde dieser Abschnitt **[ req_dn ]** benannt. |
| | **x509_extensions** | Dies bezeichnet den Abschnitt der Konfigurationsdatei, in dem eine Liste der Endungen zum Hinzufügen zum Zertifikat enthalten ist, welches durch Anwendung des **-x509**-Parameters erzeugt wird. Er kann durch den Kommandozeilen-Parameter **-extensions** übersteuert werden. |
| | **string_mask** | Diese Option blendet die Verwendung bestimmter Zeichenfolge-Typen in bestimmten Feldern aus. Wenn die Option **utf8only** eingesetzt wird, werden ausschließlich UTF8-Strings verwendet: dies ist die PKIX-Empfehlung in RFC2459 nach 2003. |
| **[ ca ]** | Dieser Abschnitt wird abgerufen, wenn Zertifikatanfragen durch Aufrufen des Befehls *openssl* mit der Option **ca** signiert werden. | |
| | **default_ca** | Wenn die Kommandozeilen-Option **-name** angewendet wird, wird damit der zu verwendende Abschnitt benannt. Andernfalls muss der zu verwendende Abschnitt in der Option **default_ca** des Abschnitts **ca** der Konfigurationdatei in unserem Beispiel **[ CA_default ]** benannt werden. |

| | | |
|---|---|---|
| **[ CA_default ]** | Dieser Abschnitt wird abgerufen, wenn Zertifikatanfragen durch Aufrufen des Befehls *openssl* mit der Option **ca**, auf die die Option **default_ca** des Abschnitts **ca** Bezug nimmt, signiert werden. | |
| | **dir** | Root-Verzeichnis der CA-Umgebung. Wenn die Konfigurationsdatei in diesem Verzeichnis abgelegt wird und falls Sie sämtliche Befehle *openssl* aus diesem Verzeichnis ausführen, können Sie ganz einfach "dir = " angeben.. |
| | **certs** | Zertifikate-Ausgabeverzeichnis. |
| | **database** | Die zu verwendende Text-Datenbankdatei (Pflichtparameter). Diese Datei muss vorhanden sein, selbst wenn sie zu Anfang leer ist. |
| | **new_certs_dir** | Hier wird das Verzeichnis festgelegt, in dem neue Zertifikate abgelegt werden. Pflichtangabe. |
| | **certificate** | Speicherort und Dateiname des CA-Zertifikats. |
| | **serial** | Eine Textdatei, in der die nächste Seriennummer zur Verwendung im Hex-Format enthalten ist. Pflichtangabe. Diese Datei muss vorhanden sein und eine gültige Seriennummer enthalten. |
| | **private_key** | Speicherort und Dateiname der Datei, in der der private Schlüssel der CA enthalten ist. |
| | **default_md** | Diese Option legt den zu verwendenden Digest-Algorithmus fest. Jeder Digest, der durch den OpenSSL-Befehl *dgst* unterstützt wird, kann verwendet werden. |
| | **default_days** | Die Standardanzahl an Tagen, die das Zertifikat gültig ist. Dieser Standardwert kann durch den Kommandozeilen-Parameter **-days** übersteuert werden. |
| | **x509_extensions** | Dies bezeichnet den Abschnitt der Konfigurationsdatei, in dem eine Liste der Endungen zum Hinzufügen zum Zertifikat enthalten ist, welches durch Anwendung des **-x509**-Parameters erzeugt wird. Er kann durch den Kommandozeilen-Parameter **-extensions** übersteuert werden. |

| | | |
|---|---|---|
| **[ req_dn ]** | Dies bezeichnet die Parameter, die die eindeutigen Namensfelder enthalten, die bei der Generierung eines Zertifikats oder einer Zertifikatanfrage per Eingabeaufforderung angezeigt werden und auf die die Option **distinguished_name** des Abschnitts **req** Bezug nimmt. Wenn die Option **prompt** im Abschnitt **req** fehlt oder auf **yes** gesetzt ist, dann enthält der Abschnitt Informationen, die über Eingabeaufforderung den Feldern zugewiesen werden. <fieldname> bezeichnet den verwendeten Feldnamen, zum Beispiel *commonName* (oder CN). | |
| | **<fieldname> = "prompt"** | Die Zeichenfolge "prompt" dient dazu, den Anwender zur Eingabe der relevanten Details aufzufordern. |
| | **<fieldname>_default ="default field value"** | Wenn der Anwender nichts eingibt, wird der Standardwert verwendet; falls es keinen Standardwert gibt, wird das Feld ausgelassen. |
| **[ policy_any ]** | Diese Option legt die zu verwendende CA-"Richtlinie" fest und muss durch den Kommandozeilen-Parameter **–policy** spezifiziert werden. Dies ist ein Abschnitt in der Konfigurationsdatei, in dem entschieden wird, welche Felder Pflichtfelder sind oder mit dem CA-Zertifikat übereinstimmen müssen. Der Richtlinien-Abschnitt besteht aus einem Variablensatz, der den DN-Feldern des Zertifikats entspricht. Wenn der Wert "**match**" ist, muss der Feldwert mit dem gleichen Feld im CA-Zertifikat übereinstimmen. Wenn der Wert "**supplied**" ist, muss er im Feld vorhanden sein. Wenn der Wert "**optional**" ist, kann er im Feld vorhanden sein. Alle Felder, die im Richtlinien-Abschnitt nicht erwähnt werden, werden im Hintergrund und automatisch gelöscht. | |
| **[ …_ext ]** | Diese Abschnitte legen die X.509-Endungen fest und werden durch die Option **x509_extensions** innerhalb der Konfigurationsdatei (Abschnitt **[ req ]** und **[ CA_default ]**) referenziert. Sie können durch den Kommandozeilen-Parameter **-extensions** übersteuert werden. | |
| | basicConstraints | Dieser Flag dient zur Bestimmung, ob das Zertifikat als ein CA-Zertifikat verwendet werden kann. |

## 1.4 CA-Zertifikat und Schlüssel erstellen

Nachdem nun alle anfänglichen Konfigurationen abgeschlossen wurden, kann ein selbstsigniertes Zertifikat erstellt werden, das als unser CA-Zertifikat verwendet werden wird. Mit anderen Worten: Wir werden dieses Zertifikat zum Signieren anderer Zertifikatanfragen verwenden.

Wechseln Sie in das CA-Root-Verzeichnis. Von diesem Verzeichnis aus können wir sämtliche **openssl-Befehle** erteilen, da unsere OpenSSL-Konfigurationsdatei (*openssl.conf*) hier abgelegt ist.

Syntax zum Erstellen von CA-Zertifikat und privatem Schlüssel:

```
openssl req -new  -config <filename> -x509 -extensions <section> -keyout
<filename> -out <filename> -days <nn>
```

| Option | Beschreibung |
|---|---|
| **req** | Der *req*-Befehl dient in erster Linie zum Erstellen und Verarbeiten von Zertifikatanfragen. Er kann dafür selbstsignierte Zertifikate erstellen, wenn die Option **-x509** festgelegt wurde. |
| **-new** | Diese Option erzeugt eine neue Zertifikatanfrage. |
| **-config <filename>** | Dies ermöglicht die Festlegung einer alternativen Konfigurationsdatei. |
| **-x509** | Diese Option gibt ein selbstsigniertes Zertifikat statt einer Zertifikatanfrage aus. |
| **-extensions <section>** | Legt den Abschnitt in der openssl-Konfigurationsdatei (vorgegeben durch **-config <filename>**) fest, in dem die X.509-Zertifikatendungen definiert werden. |
| **-keyout <filename>** | Dateiname des privaten Schlüssels des CA. Obwohl dieser durch eine Passphrase geschützt ist, sollten Sie den Zugriff darauf beschränken, sodass nur autorisierte Anwender einen Lesezugriff haben. |

**Beispiel:**

```
C:\CA>openssl req -new -config openssl.conf -x509 -extensions ca_ext -keyout
private/ca.key -out certs/ca.crt -days 3640
Erzeugt einen RSA-Private-Key mit 4096 Bit
.....................................................++
.....................................................++,
mit dem ein neuer privater Schlüssel unter 'private/ca.key' geschrieben wird
Enter PEM pass phrase: - geben Sie eine sichere Passphrase zur Verwendung mit diesem
Schlüssel ein
Verifying - Enter PEM pass phrase: - geben Sie die Passphrase zur Bestätigung erneut ein
-----
Sie werden aufgefordert werden, die Informationen einzugeben, die in
Ihre Zertifikatanfrage eingebunden werden.
Die Informationen, die Sie eingeben müssen, werden als "Distinguished Name"
(eindeutiger Name) oder DN bezeichnet.
Sie werden eine ganze Reihe an Feldern sehen, von denen Sie jedoch einige frei lassen
können.
Bei bestimmten Feldern gibt es einen Standardwert.
Wenn Sie '.' eingeben, bleibt das Feld leer.
-----
Country Name (2-stelliger Code) [DE]: - wir haben den Standardwert beibehalten
Organization Name (Unternehmen) [PHOENIX CONTACT Cyber Security AG]: - wir
haben den Standardwert beibehalten
Organizational Unit Name (Abteilung, Division) [Support]: - wir haben den Standardwert
beibehalten
Common Name (Hostname, IP oder Ihr Name) []:CA – wir haben den Common-Name für
das CA-Zertifikat eingegeben

C:\CA>
```

Es werden zwei Dateien erstellt:

– **certs/ca.crt**: Dies ist das Zertifikat der CA; es kann öffentlich zur Verfügung gestellt werden und ist selbstverständlich für alle lesbar.

– **private/ca.key**: Dies ist der private Schlüssel der CA. Obwohl dieser durch eine Passphrase geschützt ist, sollten Sie den Zugriff darauf beschränken, sodass nur autorisierte Anwender einen Zugriff erlangen können.

## 1.5 Zertifikatanfrage für den mGuard erstellen

Um ein gültiges mGuard-Zertifikat zu erhalten, müssen Sie zuerst eine Zertifikatanfrage erstellen und diese anschließend mit dem CA-Zertifikat signieren (erläutert in Kapitel „Zertifikatanfrage des mGuards mit dem CA signieren“ auf Seite 13).

Syntax zum Erstellen einer Zertifikatanfrage für den mGuard:

```
openssl req –new -config <filename> -keyout <filename> -out <filename> -days <nn>
```

| Option | Beschreibung |
|---|---|
| **req** | Der *req*-Befehl dient in erster Linie zum Erstellen und Verarbeiten von Zertifikatanfragen. |
| **-new** | Diese Option erzeugt eine neue Zertifikatanfrage. |
| **-config <filename>** | Dies ermöglicht die Festlegung einer alternativen Konfigurationsdatei. |
| **-keyout <filename>** | Dateiname des privaten Schlüssels des mGuards. Obwohl dieser durch eine Passphrase geschützt ist, sollten Sie den Zugriff darauf beschränken, sodass nur autorisierte Anwender einen Lesezugriff haben. |
| **-out <filename>** | Dateiname des mGuard-Zertifikats. |
| **-days <nn>** | Die Anzahl der Tage, die das Zertifkat gültig bleiben soll. |

**Beispiel:**

```
C:\CA>openssl req -new -config openssl.conf -keyout private/mGuard.key -out
mGuard.csr -days 364
Erzeugt einen RSA-Private-Key mit 4096 Bit
.....++
.....
+,
mit dem ein neuer privater Schlüssel unter 'private/mGuard.key' geschrieben wird.
Enter PEM pass phrase: - geben Sie eine sichere Passphrase zur Verwendung mit diesem
Schlüssel ein
Verifying - Enter PEM pass phrase: - geben Sie die Passphrase zur Bestätigung erneut ein
-----
Sie werden aufgefordert werden, die Informationen einzugeben, die in
Ihre Zertifikatanfrage eingebunden werden.
Die Informationen, die Sie eingeben müssen, werden als "Distinguished Name"
(eindeutiger Name) oder DN bezeichnet.
Sie werden eine ganze Reihe an Feldern sehen, von denen Sie jedoch einige frei lassen
können.
Bei bestimmten Feldern gibt es einen Standardwert.
Wenn Sie '.' eingeben, bleibt das Feld leer.
-----
Country Name (2-stelliger Code) [DE]: - wir haben den Standardwert beibehalten
Organization Name (Unternehmen) [PHOENIX CONTACT Cyber Security AG]: - wir
haben den Standardwert beibehalten
Organizational Unit Name (Abteilung, Division) [Support]: - wir haben den Standardwert
beibehalten
Common Name (Hostname, IP oder Ihr Name) []:mGuard – geben Sie den Common-
Name für das mGuard-Zertifikat ein


C:\CA>
```

Es werden zwei Dateien erstellt:

- **mGuard.csr**: Dies ist die Zertifikatanfrage, die durch das CA-Zertifikat signiert werden muss.
- **private/mGuard.key**: Dies ist der private Schlüssel, der nicht mit einer Passphrase geschützt wird.

## 1.6 Zertifikatanfrage des mGuards mit dem CA signieren

Die Zertifikatanfrage des mGuards muss durch die CA signiert werden, um ein gültiges Zertifikat zu werden.

Syntax zur Signierung der Zertifikatanfrage des mGuards mit dem CA:

```
openssl ca –config <filename> -out <filename> -infiles <filename>
```

| Option | Beschreibung |
|---|---|
| **ca** | Der Befehl *ca* ist eine minimale CA-Anwendung. Mit ihm können Zertifikatanfragen auf vielfältige Weise signiert und CRLs (Zertifikatssperrlisten) erzeugt werden; er unterhält des Weiteren eine Textdatenbank mit ausgestellten Zertifikaten und deren Status. |
| **-config <filename>** | Dies ermöglicht die Festlegung einer alternativen Konfigurationsdatei. |
| **-out <filename>** | Dateiname des signierten mGuard-Zertifikats. |
| **-infiles <filename>** | Dateiname der Zertifikatanfrage des mGuard. Dies muss die letzte Option sein. |

**Beispiel:**

```
C:\CA>openssl ca -config openssl.conf -out certs/mGuard.crt -infiles mGuard.csr
Verwendet die Konfiguration von openssl.conf
Enter pass phrase for C:/CA/private/ca.key: - geben Sie die Passphrase des privaten Schlüssels von CA ein
Stellen Sie sicher, dass die Anfrage mit der Signatur übereinstimmt
Signatur ist OK
Der "Distinguished Name" des Subjekts lautet wie folgt:
countryName           :PRINTABLE:'DE'
organizationName      :ASN.1 12:'PHOENIX CONTACT Cyber Security AG'
organizationalUnitName:ASN.1 12:'Support'
commonName            :ASN.1 12:'mGuard'
Das Zertfikat muss bis zum 7. Juli 2018, 09:02:23 GMT (365 Tage) ausgestellt werden
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated


C:\CA>
```

Nachdem alle diese Schritte abgeschlossen wurden, werden zwei neue Dateien erstellt:

– **certs/mGuard.crt**: Dies ist das Zertifikat des mGuards, das öffentlich zur Verfügung gestellt werden kann.
– **newcerts/01.pem**: Dies ist genau das gleiche Zertifikat, jedoch mit der Seriennummer des Zertifikats (Hex-Zahl) als Dateiname. Bei nachfolgenden Anfragen wird die Zahl um 1 erhöht. Diese Datei wird nicht mehr benötigt und kann gelöscht werden.

Nun können Sie die Zertifikatanfrage des mGuards löschen (*mGuard.csr*). Diese wird nicht mehr benötigt.

## 1.7 PKCS#12-Datei von mGuard erstellen (Maschinenzertifikat)

Diese Datei kombiniert den privaten und öffentlichen Schlüssel und ist das Maschinenzertifikat des mGuards, das über das Menü **Authentifizierung >> Zertifikate >> Maschinenezertifikate** importiert werden muss. Es erscheint eine Eingabeaufforderung, in der Sie ein Passwort eingeben müssen, durch das der PKCS#12-Export des Zertifikats vor unbefugter Nutzung geschützt wird.

Es folgt die Syntax zum Erstellen des mGuard-Maschinenzertifikats:

```
openssl pkcs12 -export -in <filename> -inkey <filename> -out <filename>
```

| Option | Beschreibung |
|---|---|
| **pkcs12** | Der *pkcs12*-Befehl ermöglicht das Erstellen und Zerteilen (Parsen) von PKCS#12-Dateien. |
| **-export** | Mit dieser Option wird festgelegt, dass eine PKCS#12-Datei erstellt und nicht zerteilt (geparst) wird. |
| **-in <filename>** | Der Dateiname, aus dem das Zertifikat ausgelesen wird. Das Format der Datei muss PEM sein. Dies ist das Zertifikat des mGuards, das Sie im vorherigen Schritt erstellt haben. |
| **-inkey <filename>** | Datei, aus der der private Schlüssel ausgelesen wird. Dies ist die Datei, in der der private Schlüssel des Zertifikats des mGuards enthalten ist. |
| **-out <filename>** | Der Dateiname, in den Zertifikate und private Schlüssel geschrieben werden. Sie werden alle im PEM-Format geschrieben. |

**Beispiel:**

```
C:\CA>openssl pkcs12 -export -in certs/mGuard.crt -inkey private/mGuard.key -out
certs/mGuard.p12

Enter pass phrase for private/mGuard.key: - geben Sie das Passwort des privaten
Schlüssels von mGuard ein

Enter Export Password: - geben Sie eine sichere Passphrase zur Verwendung für diesen
Export ein

Verifying - Enter Export Password: - geben Sie zur Bestätigung erneut die Passphrase ein


C:\CA>
```

Dieser Befehl erstellt eine Datei mit der Bezeichnung **certs/mGuard.p12**, in der der öffentliche und private Schlüssel des mGuard-Zertifikats enthalten ist. Die Datei ist durch das eingegebene Passwort geschützt.

## 1.8 Beispiel: VPN-Verbindung zwischen zwei mGuard-Geräten

Wir gehen davon aus, dass Sie die CA-Umgebung bereits eingerichtet, die Konfigurationsdatei von OpenSSL (*openssl.conf*) konfiguriert sowie CA-Zertifikat und Schlüssel erstellt haben. (So wie in den vorherigen Kapiteln beschrieben.)

**Schritt 1: Erstellen Sie eine Zertifikatanfrage für jeden mGuard**

**mGuard 1**

```
openssl req -new -config openssl.conf -keyout private/mGuard1.key -out
mGuard1.csr -days 364
```

**mGuard 2**

```
openssl req -new -config openssl.conf -keyout private/mGuard2.key -out
mGuard2.csr -days 364
```

**Schritt 2: Signieren Sie jede Zertifikatanfrage mit dem CA**

**mGuard 1**

```
openssl ca -config openssl.conf -out certs/mGuard1.crt -infiles mGuard1.csr
```

**mGuard 2**

```
openssl ca -config openssl.conf -out certs/mGuard2.crt -infiles mGuard2.csr
```

Die zwei Zertifikate **certs/mGuard1.crt** und **certs/mGuard2.crt** werden erstellt. **mGuard1.crt** muss bei mGuard **2** als Verbindungszertifikat über das Menü **IPsec VPN >> Verbindungen >> Authentifizierung** importiert werden. **mGuard2.crt** dementsprechend bei **mGuard 1**.

**Schritt 3: Erhalten Sie das Maschinenzertifikat für jeden mGuard**

**mGuard 1**

```
openssl pkcs12 -export -in certs/mGuard1.crt -inkey private/mGuard1.key -out
certs/mGuard1.p12
```

**mGuard 2**

```
openssl pkcs12 -export -in certs/mGuard2.crt -inkey private/mGuard2.key -out
certs/mGuard2.p12
```

Die zwei Exporte **certs/mGuard1.p12** und **certs/mGuard2.p12** werden erstellt.

**mGuard1.p12** muss bei mGuard **1** als Maschinenzertifikat über das Menü **Authentifizierung >> Zertifikate >> Maschinenzertifikate** importiert werden. **mGuard2.p12** dementsprechend bei mGuard **2**.