

# Zertifizierung von Hochschulservern

Der Antrag zur Zertifizierung sollte in Absprache mit dem Rechenzentrum erfolgen.

In Anlehnung an den Leitfaden unseres Providers DFN wurden für die Hochschule folgende Festlegungen getroffen:

C = DE

ST = Brandenburg

L = Brandenburg an der Havel

O = Technische Hochschule Brandenburg

OU = kann leer bleiben

CN = ( Name des Servers Bsp. www.th-brandenburg.de )

E = ( Emailadresse des Admin )

Über folgende URL können Sie webbasiert ein "Serverzertifikat" erstellen und beantragen:

<https://pki.pca.dfn.de/dfn-pki/dfn-ca-global-g2/1380>

Möchten lokal selbst das Zertifikat generieren und dann nur den Request über obige URL einreichen, finden Sie im folgenden eine Anleitung.

## Erstellen eines Schlüsselpaares mit Hilfe von "openssl"

Die folgende Anleitung zeigt die wesentlichen Schritte für die Erstellung eines Schlüssels. Eine ausführliche Dokumentation im Umgang mit openssl ist den Manuals zu entnehmen. Es werden keine Parameter im einzelnen erklärt. Hier auch der Verweis auf die man-pages.

## Erzeugen eines "private key"

Hierbei handelt es sich um die Generierung des geheimen, nicht öffentlichen Schlüssels. Dieser darf niemals veröffentlicht werden und muss vor dem Zugriff durch Dritte geschützt werden.

## Schlüsselgenerierung mit PEM pass phrase ( Achtung: Start des Webserver erfordert Eingabe der "PEM pass phrase" )

```
#>openssl genrsa -des3 -out server.key 4096
```

## ODER Schlüsselgenerierung mit PEM pass phrase und einer zusätzlichen Zufallsdatei ( Achtung: Start des Webserver erfordert Eingabe der "PEM pass

**phrase" )**

```
#>openssl genrsa -des3 -rand /var/adm/messages -out server.key 4096
```

**Schlüsselgenerierung ohne PEM pass phrase ( automatischer Start des Webserver möglich )**

```
#>openssl genrsa -out server_np.key 4096
```

**Entfernen der PEM pass phrase eines zuvor mit PEM pass phrase generierten Schlüssels**

```
#>openssl rsa -in server.key -out server_np.key
```

**Erzeugen eines "public key"**

Der öffentliche Schlüssel wird aus dem "private key" generiert. Mit diesem Schlüsselpaar erfolgt dann bei einem Verbindungsaufbau zwischen Server und Client die Verschlüsselung der Daten. Für die Zertifizierung durch eine CA muss ein sogenannter Request erzeugt werden.

**Generierung des Request für das Zertifikat**

```
#>openssl req -new -key server_np.key -out server.csr
```

**Fingerprint des Modulus erzeugen - wird von der CA gefordert zur Überprüfung des Requests.**

```
#>openssl req -noout -modulus -in server.csr | openssl sha1 -c
```

**Request als Text ausgeben zur Ansicht**

```
#>openssl req -noout -text -in server.csr
```

Nach Vollendung aller Formalitäten bekommt man von der CA seinen signierten Schlüssel. Die Schlüssel sind dann in den Webserver einzubinden. Im folgenden kurz erläutert für den Apache - Webserver.

**Implementierung der Schlüssel in Apache + mod\_ssl**

Vorausgesetzt die Schlüssel sind auf dem Webserver gespeichert, beschränkt sich die Arbeit auf die Anpassung einiger Anweisungen in der Konfigurationsdatei httpd.conf.

```
# Server Certificate:
```

```
# öffentlicher key von der CA signiert
```

```
SSLCertificateFile <path>/<filename>
```

```
# Server Private Key:
```

# geheimer key

SSLCertificateKeyFile <path>/<filename>

# Server Certificate Chain:

# Zertifizierungskette, im Fall der THB-CA, werden in dieser Datei die Keys der CA hinterlegt, die notwendig sind um die Kette bis zur Wurzel zu verfolgen

# Bsp.: das Server Zertifikat wurde mit dem THB CA Server Key signiert, dieser vom THB Top Level CA.

# mit Hilfe folgender Anweisung: "cat <filename THB CA Server Key> <filename THB Top Level CA Key> > <filename>"

# wird eine Zertifizierungskette erstellt. Hat der Benutzer zuvor einmalig das Wurzelzertifikat der CA geladen, übermittelt der Server die restlichen Schlüssel.

# Damit kann der Client/Browser/Benutzer die Kette bis zur Wurzel verfolgen und vertraut damit der Verbindung.

SSLCertificateChainFile <path>/<filename>