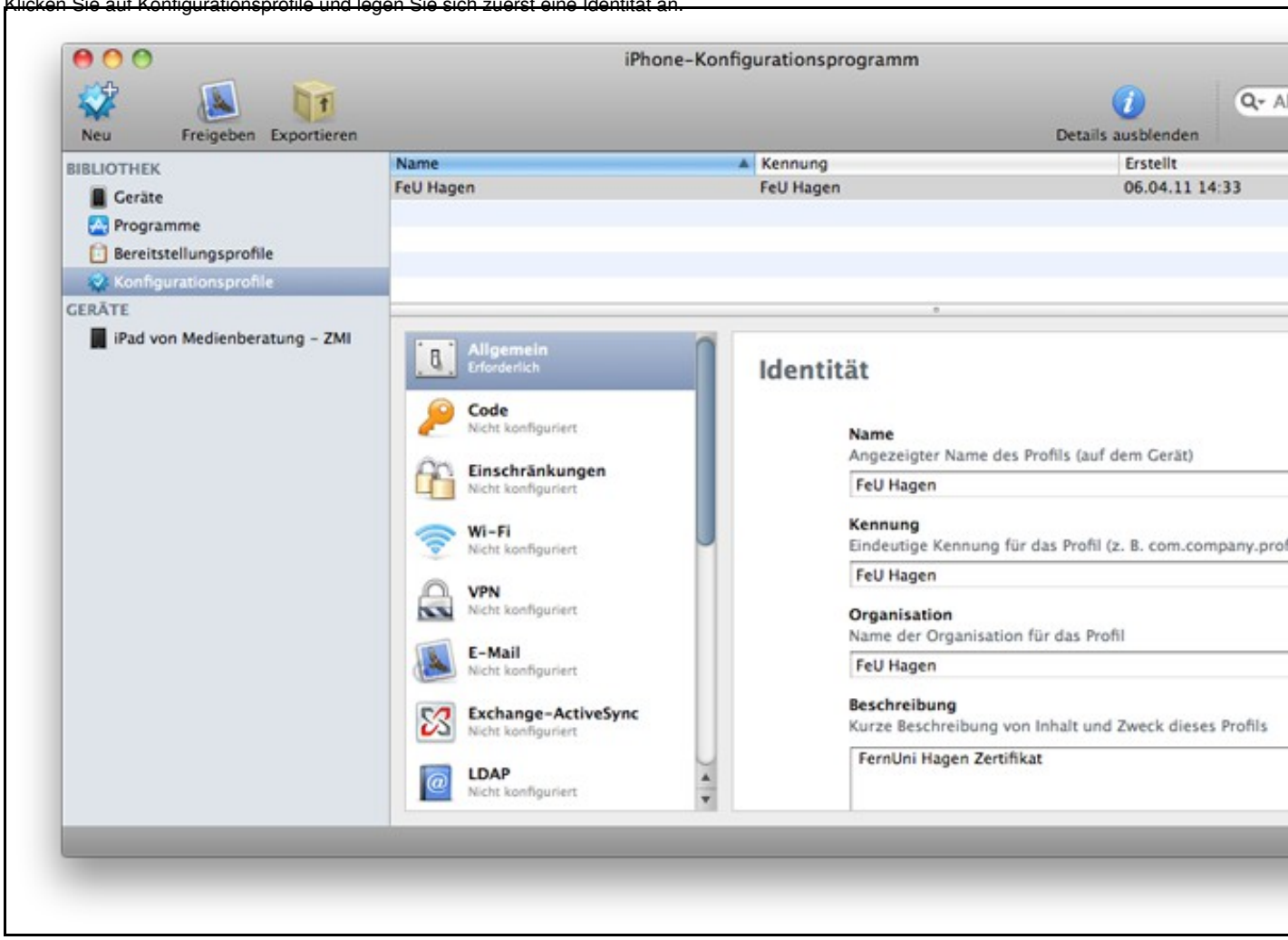
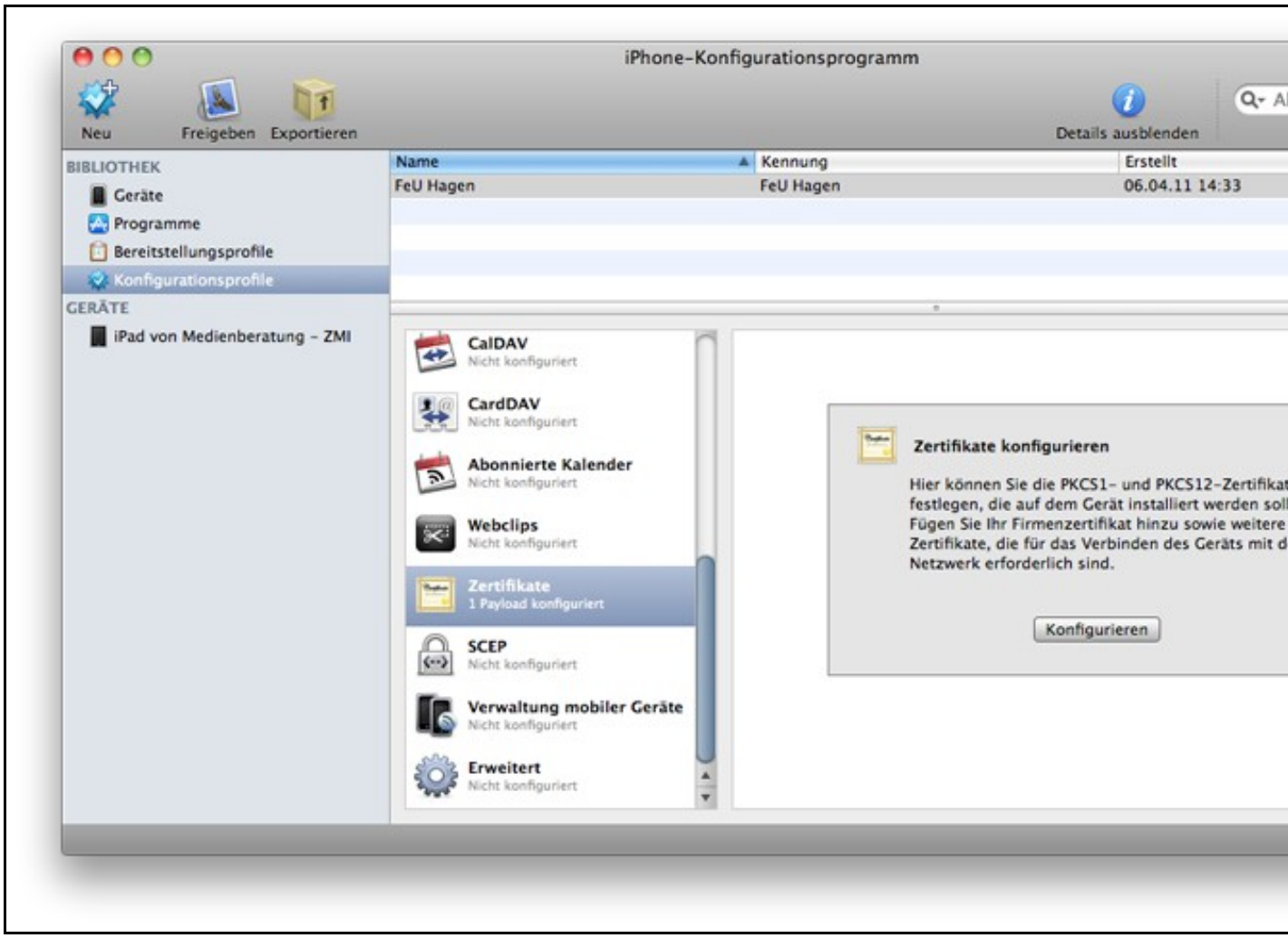
Table of Contents

1 Installation von Zertifikaten auf dem iPad

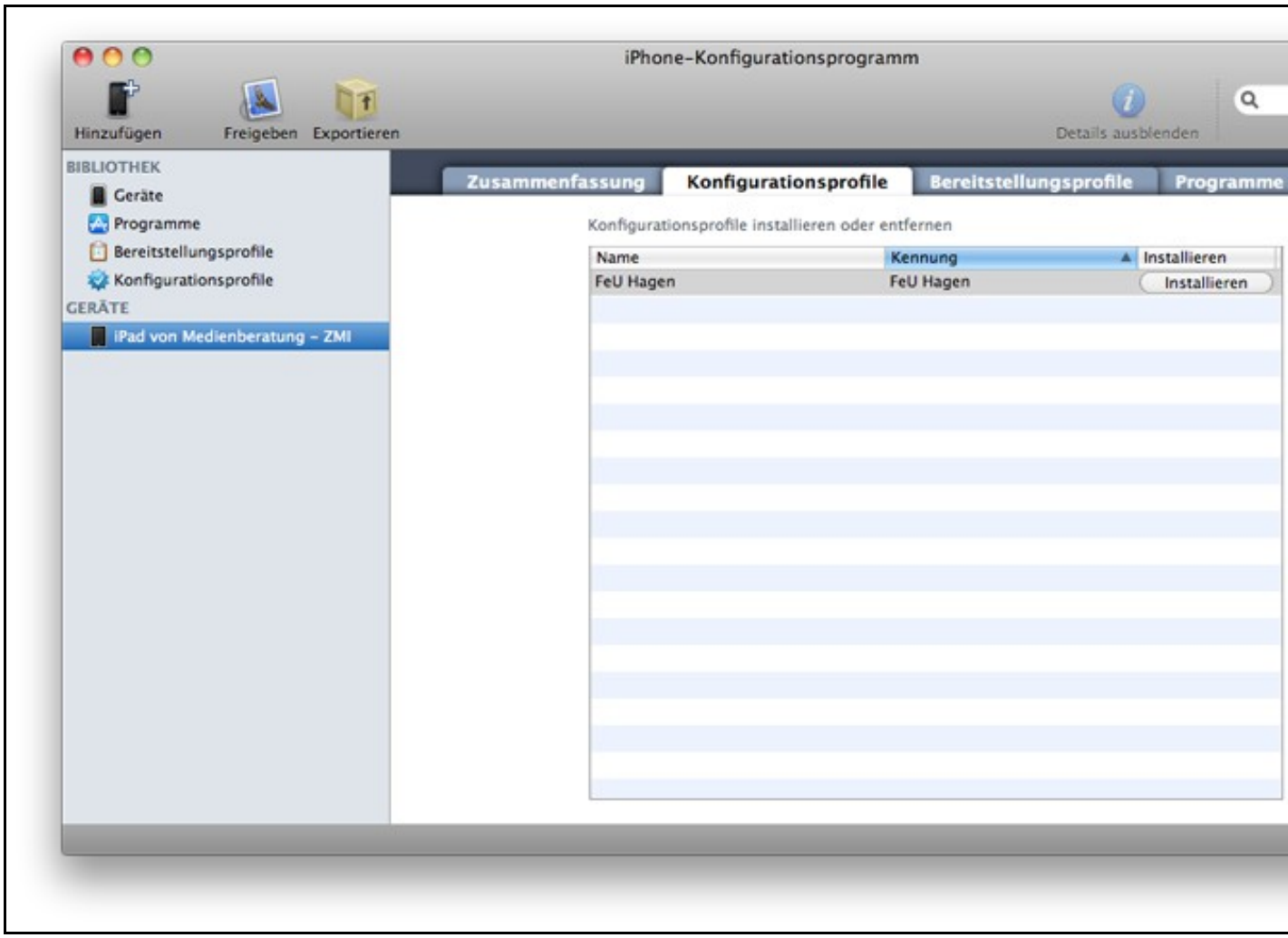
1. Für die Installation des Zertifkats auf dem iPad benötigen Sie das iPhone Konfigurationsprogramm, welches Apple kostenlos zur Verfügung stellt. Laden Sie sich dieses Programm aus dem Internet herunter und installieren Sie es auf Ihrem PC/Mac, mit dem Sie Ihr iPad synchronisieren.
2. Ihr installiertes Zertifikat können Sie aus dem Browser als eine PKCS 12-Datei (.p12) exportieren.
3. Öffnen Sie das iPhone-Konfigurationsprogramm und schließen Sie Ihr iPad an Ihren PC/Mac an.
4. Klicken Sie auf Konfigurationsprofile und legen Sie sich zuerst eine Identität an.



5. Klicken Sie anschließend auf Zertifikate und konfigurieren Sie Ihre Zertifkate. Wählen Sie im folgenden Upload-Dialog die zuvor exportierte .p12-Datei aus.



6. Wählen Sie anschließend im Konfigurationsprogramm Ihr iPad aus und klicken Sie dort auf den Reiter "Konfigurationsprofile". Starten Sie die Installation durch Klick auf den Button "Installieren". Bestätigen Sie die Installation auf dem iPad.



7. Das Zertifkat können Sie jederzeit über das iPhone Konfigurationsprogramm oder direkt auf dem iPad unter "Einstellungen" - "Allgemein" - "Profil" einsehen oder wieder löschen.

2 Serverzertifikat beantragen

1. REDIRECT

3 Signieren von Applets

4 Einleitung

Die Verwendung von Java-Applets ist eine der ältesten Möglichkeiten Interaktivität in eine Web-Seite zu bringen (und war aus diesem Grund in den 90er Jahren ein Grund für die schnelle Verbreitung von Java). Hinter einem Applet steckt Java-(Byte-)Code, der über das HTTP-Protokoll auf den Rechner des Benutzers geladen und dort ausgeführt wird. Aufgrund der mit diesem Verfahren verbundenen Risiken, wurden die Sicherheitsbestimmungen der Java Runtime Engine in den letzten Jahren immer stärker erhöht, womit gleichzeitig die reibungslose Ausführung des Applets erschwert wird.

Ein Ausweg aus dieser immer enger werden Zwickmühle ist die Möglichkeit *Applets* mit Hilfe eines Zertifikats zu *signieren*. Die folgende Beschreibung zeigt die hierfür nötigten Schritte.

5 Liste der benötigten Zutaten

- Die Java Standard Edition (**Java SE**). Die Java SE enthält die Tools *keytool*, *jar* und *jarsigner*.

Das Java SE gibts auf den Seiten von Oracle kostenlos zum Download.

- Das zu signierende Applet, bspw. als *MeinApplet.jar*.

6 Erstellung des Signatur-Zertifikats

Zur Signatur des Applets wird vorab ein Sicherheits-Zertifikat benötigt, welches sich Mitarbeitende der FernUniversität über die hochschulweite *Certification Authority* (CA) ausstellen lassen können. Die im weiteren Verlauf gezeigten Arbeitsschritte müssen glücklicherweise nur *einmal* durchgeführt werden, d.h. ein auf diese Weise erstelltes Zertifikat kann für die Signierung mehrerer Applets verwendet werden.

So wird's gemacht:

6.1 Erstellung des Zertifikatsschlüssels

Die Anleitung ist an dieser Stelle nicht mehr aktuell !!!

Bitte beantragen Sie das benötigte Zertifikat nach folgender Anleitung: Code Signing Zertifikat beantragen

~~Setzen Sie in einem Windows *Command Prompt* (unter Unix auf Shell-Ebene) den folgenden Befehl ab.~~

```
keytool -genkey -keyalg rsa -alias meinZertifikat
```

Der Alias *meinZertifikat* bezeichnet dabei einen (frei wählbaren) Namen, unter dem das Zertifikat später in Ihrem lokalen Java-Zertifikats-Speicher abgelegt werden wird.

Der keytool-Befehl erzeugt eine Reihe von Abfragen, die wie im folgenden gezeigt beantwortet werden sollten.

```
Wie lautet Ihr Vor- und Nachname?
  [Unknown]:  GRP: CodeSigning Abt. Basisanwendungen
Wie lautet der Name Ihrer organisatorischen Einheit?
  [Unknown]:  Zentrum fuer Digitalisierung und IT
Wie lautet der Name Ihrer Organisation?
  [Unknown]:  FernUniversitaet in Hagen
Wie lautet der Name Ihrer Stadt oder Gemeinde?
  [Unknown]:  Hagen
Wie lautet der Name Ihres Bundeslands?
  [Unknown]:  Nordrhein-Westfalen
Wie lautet der Ländercode (zwei Buchstaben) für diese Einheit?
  [Unknown]:  DE
Ist CN=GRP: CodeSigning Abt. Basisanwendungen, OU=Zentrum fuer Digitalisierung und IT, O=FernUniversitaet in Hagen, L=Hagen,
ST=Nordrhein-Westfalen, C=DE  richtig?
  [Nein]:  Ja
```

Zu beachten!

- Als Vor- und Nachname (CN) verwenden **Sie in jedem Fall die Zeichenkette *GRP: CodeSigning*** gefolgt vom Namen Ihrer Abteilung oder Ihres Lehrgebiets.
- Als organisatorische Einheit (OU) versenden Sie den Namen Ihrer zentralen Einrichtung oder Ihrer Fakultät.
- Benutzen Sie **keine Umlaute**.

6.2 Erstellung der Zertifikats-Anforderung (CSR)

```
keytool -certreq -alias meinZertifikat -file MeinCSR.csr
```

generiert auf Basis des eingeführten Alias *meinZertifikat* die Zertifikats-Anforderung in der Datei *MeinCSR.csr*.

6.3 Ausstellung des Zertifikats

Zur Ausstellung des Zertifikats öffnen Sie in Ihrem Webbrowser die Seite https://pki.pca.dfn.de/fernuni-hagen-ca-g2/pub.

- Klicken Sie auf die Karteikarte **Serverzertifikat**.
- Füllen Sie das auf dieser Seite befindliche Formular aus, und wählen Sie als **Zertifikatsprofil** den Eintrag **User**!

Der Ausstellungsprozess endet mit der Erstellung eines PDF-Dokuments, welches **persönlich** (d.h. ausgedruckt und unterschrieben) bei unserer Certification Authority (Herr Heikamp / Herr Löffler, *Zentrum für Digitalisierung und IT, AVZ, Raum A212*) abgegeben werden muss. Dieser ändert den Eintrag User auf "Code-Signing".

Weisen Sie die Kollegen bei der Gelegenheit sicherheitshalber nochmal darauf hin, dass das Zertifikat für *Code-Signing* benötigt wird. Das Zertifikat wird Ihnen im Anschluss per E-Mail zugeschickt.

6.4 Import der CA-Zertifikate in den Keystore

Laden Sie über die Seite https://pki.pca.dfn.de/fernuni-hagen-ca-g2/pub jeweils das Wurzelzertifikat, das DFN-PCA-Zertifikat, und das DFN-CA-Global-G2-Zertifikat einzeln als .cer Dateien herunter.

Laden Sie die Wurzelzertifikate einzeln als .cer Dateien herunter.

Anschließend werden die Zertifikate über den folgenden Befehl einzeln in den Keystore eingebunden:

```
keytool ?import ?alias <Name> -file <Dateipfad>.cer
```

Die Abfragen nach der Vertrauenswürdigkeit werden mit "Ja" beantwortet.

Erst nach diesem Schritt ist es möglich das eigene Zertifikat mit dem ausgestellten Zertifikat zu verbinden.

6.5 Import des Zertifikats

Der Import des ausgestellten Zertifkats in den Java-Zertifikatsspeicher ist dafür relativ trivial.

```
keytool -import -alias meinZertifikat -trustcacerts -file cert.p7b
```

erzeugt die Verbindung des neuen Zertifikats (aus der Datei *cert.p7b*) mit dem mittlerweile bekannten Alias *meinZertifikat*.

7 Signieren der Applet-Datei

- Stellen Sie sicher, dass sich Ihre Applet-Datei, in unserem Beispiel (s.o.) *MeinApplet.jar*, in einem ansonsten leeren Verzeichnis befindet.
- Erstellen Sie in diesem Verzeichnis eine neue Datei *manifest-addons* mit den folgendem zwei Zeilen als Inhalt:

```
Application-Name: Applet-Name
Permissions: all-permissions
```

Application-Name ein frei wählbarer Name, der dem Benutzer später beim Start des Applets angezeigt wird. Möglich wäre bspw. ein Eintrag wie *Mein Applet (FernUni-signiert)*.

Über *Permissions* bestimmen Sie die Rechte-Matrix Ihres Applets. *all-permissions* lässt dem Applet dabei alle möglichen Freiheiten, u.a. die Möglichkeit einer Netzwerkverbindung zu externen Servern, sowie den Zugriff auf das lokale Datei-System des Rechners, auf dem es ausgeführt wird!

Erweitern Sie nun Ihre Applet-Datei *MeinApplet.jar* durch die oben festgelegten Manifest-Ergänzungen

```
jar ufm MeinApplet.jar manifest-addons
```

Als letzte Maßnahme wird das Gesamt-Paket nun signiert und in einer neuen Datei *MeinAppletSigned.jar* abgelegt.

```
jarsigner -verbose -tsa http://zeitstempel.dfn.de/ -signedjar MeinAppletSigned.jar MeinApplet.jar meinZertifikat
```

Das Einfügen eines Timestamps über die Option *-tsa* birgt den Vorteil, dass dem zum signieren benutzten Zertifikat die Gültigkeit zum Zeitpunkt der Applet-Signierung bescheinigt wird. Hierdurch entfällt ein evtl. später regelmäßig benötigtes "nachsignieren". Die Bereitstellung des Timestamp übernimmt freundlicherweise der DFN-Verein als Service unter der o.a. Adresse.

8 Änderungen in der HTML-Umgebung

Ersetzen Sie auf Ihrem Server nun die Datei *MeinApplet.jar* durch die signierte Variante *MeinAppletSigned.jar*.

Suchen Sie jetzt in der korrespondierenden HTML-Datei die Stelle, an der das Applet eingebunden wird. Prinzipiell sollte sie ungefähr folgendes Aussehen haben:

```
<applet code="MeinApplet.class"
        archive="MeinApplet.jar"
        codebase="...">
 (...)
</applet>
```

Passen Sie diese Umgebung nun an die signierte Applet-Datei an.

- Ändern Sie den Wert des *archive*-Parameters in *MeinAppletSigned.jar*.
- Fügen Sie die Zeile

```
<param name="Permissions" value="all-permissions">
```

in die <applet>-Umgebung ein. Der *value*-Wert muss dabei mit der Angabe in der o.a. Datei *manifest-addons* identisch sein, in unserem Fall also auf *all-permissions* gesetzt werden.

Das Ganze noch einmal im Überblick zusammengefasst:

```
<applet code="MeinApplet.class"
        archive="MeinAppletSigned.jar"
        codebase="...">
<param name="Permissions" value="all-permissions">
 (...)
</applet>
```

Laden Sie nun die o.a. HTML-Seite in Ihren Webbrowser, und kontrollieren Sie, ob das signierte Applet wie gewünscht geladen wird.