Zürcher
Hochschule
Winterthur

LDAP-Server für X.509 Zertifikate

Projektarbeit Sommersemester 2000

Dozent: Dr. Andreas Steffen
Studenten: Markus Grieder It3a
Stephan Zehnder It3a

Datum: 14. März – 19.Mai 2000
Erstellt am: 19.Mai 2000

Inhaltsverzeichnis

1 Zusammenfassung

Unsere Projektarbeit des Sommersemesters 2000 trägt den Titel „LDAP-Server für X.509 Zertifikate“. Bei der Arbeit betreut wurden wir durch Dr. Andreas Steffen.

Es ging in dieser Arbeit darum, für die Zürcher Hochschule Winterthur (ZHW), eine LDAP-Verzeichnisstruktur auf der Plattform des Netscape Directory Servers aufzubauen. Dieser LDAP-Server soll „public keys“ in der Form von standardisierten X.509 Zertifikaten zur Verfügung stellen und so der ZHW in Zukunft sicheren E-Mail Verkehr ermöglichen.

In diesem Dokument ist nebst der eigentlichen Dokumentation der Projektarbeit auch noch eine Einführung in LDAP und X.509 enthalten. Wir möchten darauf hinweisen, dass wir SSL und X.509 nicht im Detail angeschaut und in diesem Dokument aus diesem Grund auch nicht viel darüber geschrieben haben.
Auch über eine Teilaufgabe unseres Projektes, das Installieren von OpenSSL und das Erstellen der Schlüssel, haben wir nicht viel geschrieben. Statt dessen verweisen wir auf entsprechende Dokumente.

Das erste Ziel war, unser Wissen über LDAP und X.509 zu vertiefen. Doch schon sehr bald konnten wir damit beginnen, Zertifikate mit OpenSSL zu erstellen und den Netscape Directory Server zu installieren. Danach ging es ans Planen und Implementieren des LDAP-Schemas. Auch ein grosser Anteil an Arbeit und Zeit, hatte das Studieren von OpenCA. OpenCA ist ein Tool welches das Erstellen und Verwalten von Zertifikaten mit einer eigenen Zertifizierungsstelle ermöglicht.

Wir glauben das sehr viel von unserer Arbeit übernommen werden kann, um hier an der ZHW zukünftig mit Zertifikaten arbeiten zu können. Das von uns entworfene Schema muss natürlich schon noch genau an die Bedürfnisse der Schule angepasst werden.
Ein Problem ist, dass uns keine ausgereifte Software bekannt war, um einem Benutzer eine komfortable Abfrage des LDAP-Servers zu ermöglichen. Es wäre sicherlich sinnvoll, hier eine webbasierte Lösung zu entwickeln, oder zu suchen, damit der LDAP-Dienst auch voll ausgenützt wird.
Einzelne Softwarelösungen für PKI-Anwendungen (public key infrastructure) gibt es erst seit kurzem und die Entwicklung in diesem Bereich ist noch stark in Bewegung. Vor allem fehlt es aber noch an Anwendungen. Es ist also zu erwarten, dass es in zwei Jahren schon wieder ganz anders aussieht.

2 Aufgabenstellung

2.1 Beschreibung

Durch die Verlagerung wichtiger Geschäftsprozesse auf das Internet steigen die Anforderungen an die Transaktionssicherheit. Im privaten Bereich wird mit zunehmender Zahl der persönlichen Anwendungen, zusätzlich zu den unabdingbaren Sicherheitsaspekten, auch auf die Wahrung der Privatsphäre grossen Wert gelegt.

Moderne Lösungen, welche die verlangte Sicherheit und Vertraulichkeit gewährleisten können, beruhen ausnahmslos auf einer Public Key Infrastruktur (PKI). Damit sich diese Systeme jedoch auf breiter und globaler Basis durchsetzen können, muss der „public key" jedes Internet-Teilnehmers in einer benutzerfreundlichen Art und Weise online verfügbar sein. Dies kann mit einem LDAP Directory-Server, der die „public keys" in der Form von standardisierten X.509 Zertifikaten zur Verfügung stellt, realisiert werden.

In dieser Arbeit soll für die Zürcher Hochschule Winterthur (ZHW) auf der Plattform des Netscape Directory Servers eine LDAP-Verzeichnisstruktur aufgebaut werden, die es den Benützern erlauben wird, per Mausklick den „public key" des gewünschten Kommunikationspartners in ihren Netscape Browser, respektive Microsoft Internet Explorer zu laden und damit eine sichere, verschlüsselte E-Mail Kommunikation ermöglicht.

Zusätzlich werden es die digitalen ZHW-Zertifikate SSL-basierten Servern erlauben, die Zugriffsberechtigung der Benützer zu verifizieren. Das gleiche gilt für Security Gateways, die berechtigten externen Hosts den Zugriff auf das Schulnetz via IPsec-basierten Internet-Tunnels ermöglichen können.

2.2 Aufgaben

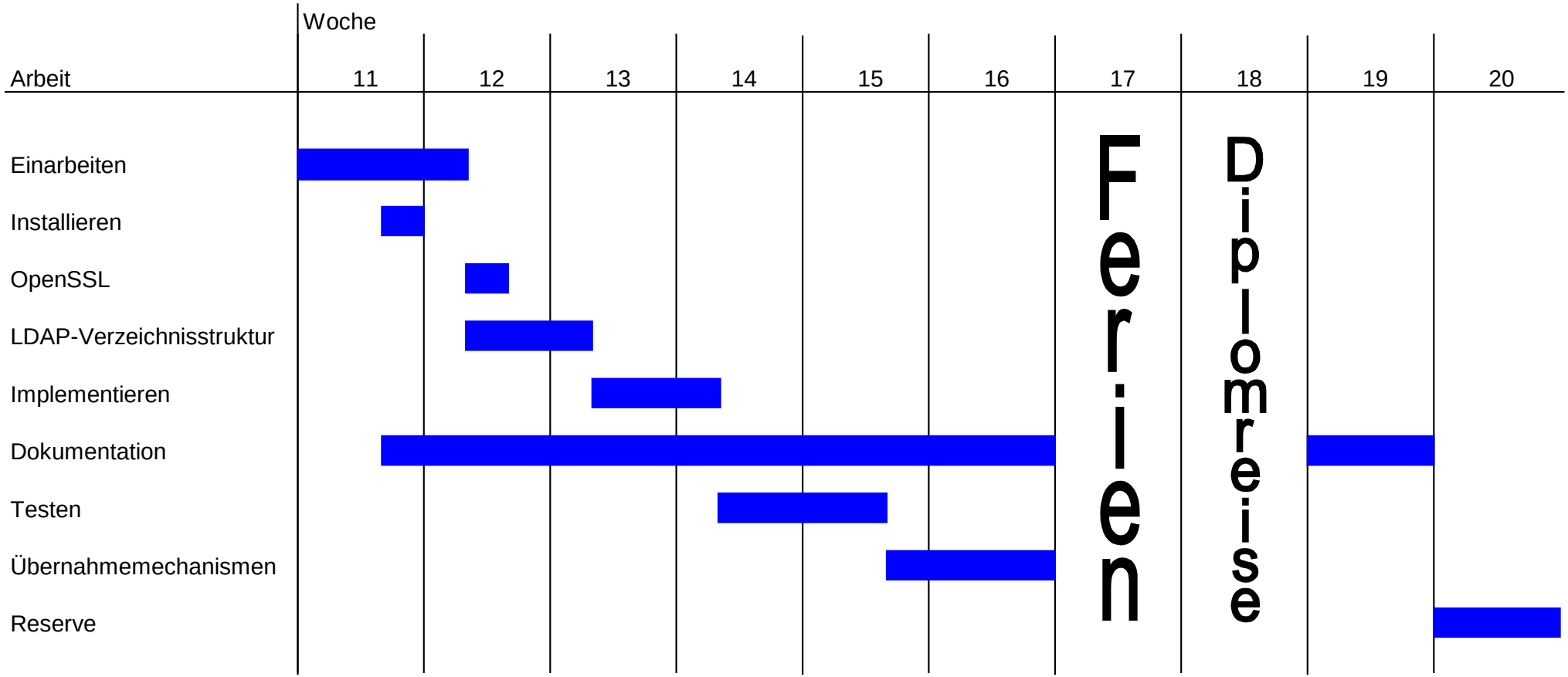
- Studium des Aufbaus und der Abfragemöglichkeiten von LDAP-Verzeichnissen.
- Erstellen eines X.509 ZHW-Root Zertifikats und einer Anzahl von Benutzerzertifikaten.mit
- Hilfe des OpenSSL Tool Kits.
- Installation des Netscape Directory Server 4.11 wahlweise unter Linux 2.2.x oder unter
- Windows NT 4.0.
- Abklärung des Übernahmemechanismus von X.509 E-Mail Zertifikaten im Netscape
- Browser 4.x, respektive Microsoft Internet Explorer 5.x.
- Erstellen einer LDAP-Verzeichnungsstruktur für die ZHW
- Implementation und Austesten des Konzeptes
- Dokumentation der Projektarbeit

2.3 Infrastruktur / Tools

Rechner: 2 PCs mit Dual-Boot: SuSE Linux 6.3 / Windows NT 4.0
SW-Tools: Netscape Directory Server 4.11, OpenSSL Tool Kit, OpenCA

3 Projektplanung

3.1 Zeitplan



3.2 Erläuterungen zum Zeitplan

3.2.1 Einarbeiten

• Unterlagen von IBM durchlesen	MG	SZ
• X.509 studieren	MG	
• OpenSSL studieren		SZ

3.2.2 Installieren

• Installieren des Netscape Directory Servers	MG	

3.2.3 OpenSSL

• Generieren eines Root-Keys		SZ
• Generieren verschiedener User-Keys		SZ

3.2.4 LDAP-Verzeichnisstruktur erstellen

• Abklären der bereits vorhanden Verzeichnistruktur	MG	SZ
• Abklären der Anforderungen	MG	SZ
• Gesamtes Schema aufstellen		SZ

3.2.5 Implementieren

• Aufbauen des Directory Services mit dem Netscape Directory Server		SZ
• Aufbauen und testen der OpenCA Plattform	MG	

3.2.6 Übernahmemechanismen

• Überprüfen der Möglichkeit mit MIME-Type Ist wahrscheinlich bereits in der OpenCA Plattform vorhanden.	MG	
• Client-Authentifikation bei SSL testen.	MG	

3.2.7 Dokumentation

• Einführung zu LDAP und X.509		SZ
• Übernahme der Zertifikate über LDAP / Web mit den Clients Netscape / Internet Explorer dokumentieren.	MG	
• LDAP-Struktur		SZ
• Installation	MG	
• Generieren der Zertifikate dokumentieren.		SZ
• Einlesen der Zertifikate über LDIF dokumentieren.		SZ
• Wie man das Schema implementiert		SZ
• OpenCA	MG	
• Zusammenfassung, Schlusswort, Ausblick	MG	SZ

3.2.8 Testen

Da uns keine ausgereifte Software bekannt war, die volle Abfragemöglichkeiten an einen LDAP-Server zu Verfügung stellt, konnten wir nicht ausführlich testen. Dies betrifft besonders das Testen der Rechte.

Das Kürzel in der rechten Spalte bezeichnet die verantwortliche Person

4 Einführung in LDAP

4.1 Was ist ein Verzeichnisdienst?

Ein Verzeichnisdienst ist ähnlich einer Datenbank, die Informationen in einem Verzeichnisdienst werden aber viel häufiger gelesen als geschrieben. Ein Verzeichnisdienst ist deshalb optimiert für eine schnelle Suche und das Lesen von Informationen. Man kann also sagen, dass ein Verzeichnisdienst sehr schnellen Zugriff auf Informationen erlaubt. Die Daten eines Verzeichnisdienstes können über mehrere Computer verteilt werden.

4.2 Weshalb ein Verzeichnisdienst?

Der Sinn eines Verzeichnisdienstes ist es, aktuelle Informationen an einem einzigen Ort abgespeichert zu haben. Die Informationen sollen nicht in verschiedenen Datenbanken abgespeichert werden, wenn möglich noch mit unterschiedlich aktuellen Einträgen, so dass die Informationen nicht übereinstimmen. Wenn etwas ändert, soll man es in einer Datenbank anpassen und nicht an verschiedenen Orten. Ausserdem sollen die Informationen allen zugänglich sein, natürlich entsprechend definierbarer Zugriffsrechte. Das Resultat ist effizientere und billigere Informationsverwaltung.

4.3 Was ist LDAP?

LDAP stammt ursprünglich von X.500 ab, welches das Directory Access Protocol DAP definiert. DAP läuft über den gesamten OSI-Stack. LDAP ist ein Verzeichnisdienst-Protokoll das direkt über TCP/IP läuft und die meisten Funktionalitäten von X.500 unterstützt. In einem LDAP-Verzeichnisdienst können Entitäten, die verschiedene Attribute und einen eindeutigen Namen, den `distinguished name dn` haben, abgespeichert werden. Jedes Attribut hat einen Typ und einen oder mehrere Werte.
Die Einträge sind hierarchisch in einer Baumstruktur angeordnet, welche typischerweise zum Beispiel die Organisationsstruktur einer Firma repräsentiert. Entitäten, welche Länder repräsentieren stehen an der Spitze, darunter stehen Entitäten welche Organisationen repräsentieren und weiter unten können Personen, Abteilungen und zum Beispiel Drucker stehen.
Durch die Werte des Attributs mit dem Namen `objectclass` wird angeben, welche Attribute benötigt und welche erlaubt sind.
Der LDAP Verzeichnis Dienst basiert auf einem Client/Server Modell. Ein LDAP-Client macht eine Anfrage auf einen LDAP-Server. Der Server liefert die Antwort oder einen Referenz auf weitere Informationen zurück

4.4 Schema und Objekte

Beim Netscape Directory Server sind bereits viele Objektklassen und Attribute vordefiniert. Es ist jedoch möglich beliebige weitere Objektklassen und Attribute hinzuzufügen.Das Schema bezeichnet das Format und die Eigenschaften der definierten Objektklassen und Attribute.
Die Entitäten in LDAP sind in Form von Objekten abgespeichert. Dies wurde so gemacht, weil es die Abspeicherung von allen möglichen Formen von Daten erlaubt. Ein LDAP-Server ist darum flexibler als eine relationale Datenbank, vor allem beim Hinzufügen und Löschen von Attributen. Es bestehen auch keine Längenbeschränkungen für die Eingaben.
Die meisten Werte in einem LDAP-Record sind Text basiert, es gibt jedoch auch Binärdaten. Meistens sind Binärdaten zum Beispiel im jpeg Format gespeicherte Bilder. Um grosse Binärobjekte in die Datenbank einzufügen wird jedoch empfohlen, diese auf einem Web-Server abzuspeichern und nur den Link darauf in die Datenbank aufzunehmen.

4.5 Die vier Objekttypen

4.5.1 Personen Objekte

Personen Objekte sind Einträge die gebraucht werden um Personen in einer Organisation zu beschreiben. Beispiele von mögliche Klassen sind `organizationalPerson` oder `inetOrgPerson`.

4.5.2 Organizational Unit Objekte

Dieser Objekttyp erlaubt es Verzweigungen im Verzeichnis-Informations-Baum zu erstellen. Die Organizational Unit Objekte sind in der Objektklasse `organizationalUnit` und durch das Attribut ou definiert. Das ou Attribut ist vielfach in Objektklassen ausserhalb von `organizationalUnit` definiert wie zum Beispiel in der `organizationalPerson` Klasse.

4.5.3 Group Objekte

Der Standart definiert nur statische Gruppen. Objekte von statischen Gruppen enthalten das Attribut `uniquemember`. Dieses Attribut enthält einen oder mehrere distinguished names (dn) Werte von Member-Entitäten. Normalerweise umfasst eine Gruppe verschiedene Personen, es können aber alle möglichen Arten von Objekten zu Gruppen zusammengefasst werden. Die Objekt-Klasse heisst `groupOfUniqueMembers`.
Wir haben zusätzlich noch die von Netscape definierten dynamischen Gruppen verwendet. Diese Objekte sind in der Objektklasse `groupOfUrls` definiert. Sie enthält nur die URL für eine oder mehrere LDAP-Abfrage,welche die Mitglieder definieren und kein Attribut `uniquemember`.

4.5.4 Domain Objekte

Weil diese Objekte die Wurzel des Verzeichnis-Informations-Baumes bilden, nennt man sie root level objects. Die Objekte sind definiert durch die Klasse `domainObject` und beinhalten die Attribute `organization`, `country`, `location` und `domain`.

4.6 Was ist LDIF?

LDIF ist die Abkürzung für LDAP Data Interchange Format. Es ist ein textbasierstes Format welches die Informationen durch mnemonics darstellt. Es wird sowohl verwendet um Daten für die Benützer darzustellen als auch Daten in den LDAP-Server zu laden.

4.7 Beispiele von LDAP-URLs

Alle Informationen über Andreas Steffen:
„ldap://ksy112.zhwin.ch/o=zhwin.ch??sub?(cn=Andreas Steffen)“

Alle Informationen über uid=6157 (uid=UserIdentifier):
„ldap://ksy112.zhwin.ch/o=zhwin.ch??sub?(uid=6157)“

Die E-Mail von Andreas Steffen:
„ldap://ksy112.zhwin.ch/o=zhwin.ch?mail?sub?(cn=Andreas Steffen)“

Die uid und die E-Mail von Andreas Steffen:
„ldap://ksy112.zhwin.ch/o=zhwin.ch?uid,mail?sub?(cn=Andreas Steffen)“

Das User-Zertifikat von Andreas Steffen
„ldap://ksy112.zhwin.ch/o=zhwin.ch?userCertificate?sub?(cn=Andreas Steffen)“

Alle Informationen, bei welchen der cn (CommmonName) den String „Steffen“ enthält:
„ldap://ksy112.zhwin.ch/o=zhwin.ch??sub?(cn=*Steffen)“

Den cn und das Use-Zertifikat der Einträge, bei welchen der cn den String „Steffen“ enthält:
„ldap://ksy112.zhwin.ch/o=zhwin.ch?cn,userCertificate?sub?(cn=*Steffen)“

5 Einführung in X.509

5.1 Einleitung

E-Mail ist heute zu einem der wichtigsten Kommunikationsmittel geworden und die Verwendung wird in Zukunft noch zunehmen. Die Sicherheit bei der E-Mail Kommunikation weist jedoch einige Lücken auf. So kann man den Absender nicht erkennen, auch nachträgliche Änderungen kann man nicht feststellen. Ein weiteres Problem ist die Möglichkeit, dass eine E-Mail von Drittpersonen gelesen werden könnte.

5.2 S/MIME

S/MIME ist ein Standard der sicheres Kommunizieren per E-Mail ermöglicht. S/MIME wurde von verschiedenen Firmen, aber vor allem durch die RSA Data Security entworfen. S/MIME erweitert das MIME-Format (Multipurpose Internet Mail Extensions) um kryptographische Dienste, also Verschlüsselung und digitale Signaturen.

5.3 Authentifikation

Dank dem Public-Key-Verfahren ist das Problem, dass man mit jedem Partner einen geheimen Schlüssel vereinbaren muss behoben. Beim Public-Key-Verfahren verwendet man den öffentlichen Schlüssel des Empfängers. Doch man kennt nicht jeden auf dem Netz und nicht jedem kann man trauen. Das Problem ist, wie kann ich feststellen, ob der andere auch derjenige ist, für den er sich ausgibt. Um dieses Problem zu lösen verwendet S/MIME den X.509-Zertifikatstandart.

5.4 X.509-Zertifikate

Ein X.509-Zertifikat kann mehrere Verwendungszwecke haben, so zum Beispiel für sichere E-Mail, sichere Online-Verbindungen oder zur Signatur von Objekten. X.509-Zertifikate enthalten neben dem Namen des Inhabers auch noch Angaben zur Organisation, seine E-Mail-Adresse sowie eine Gültigkeitsdauer.
Um ein Zertifikat zu erhalten, muss der Benutzer einen Antrag an eine CA stellen. Wenn er sich gegenüber der CA ausgewiesen hat, unterschreibt diese sein Zertifikat mit ihrem privaten Schlüssel. Bei den grösseren CA's (Verisign, Swisskey) gibt es verschiedene Klassen von Zertifikaten, mit unterschiedlichen Authentifizierungsprozessen. Bei Klassen mit einer grösseren Sicherheit kann es nötig sein, persönlich mit einem Ausweis bei der CA vorbeizugehen.
Technisch gesehen ist ein Zertifikat eine Datei, die den öffentlichen Schlüssel eines Benutzers, Angaben zu seiner Person und eine digitale Signatur einer Certificate Authority (CA) enthält. Diese Signatur kann jeder Benutzer mit Hilfe eines Wurzelzertifikats der CA auf seine Echtheit überprüfen. Ein Wurzelzertifikat ist von der CA selbst unterschrieben, für die wichtigsten CA's sind sie in den Browsern bereits enthalten. Die Tatsache, dass Wurzelzertifikate von der CA selbst unterschrieben werden, heisst natürlich, dass man dieser CA vertrauen muss.

5.5 Wie bekommt man den öffentlichen Schlüssel einer Person?

Um eine verschlüsselte Nachricht an eine Person zu schicken, braucht man deren öffentlichen Schlüssel. Um diesen zu erhalten gibt es verschiedene Möglichkeiten. Man kann zum Beispiel die andere Person darum bitten, eine signierte E-Mail zu senden, dadurch erhält man seinen öffentlichen Schlüssel, der im Zertifikat enthalten ist. Empfangene Schlüssel werden bei den meisten Browsern automatisch ins Adressbuch aufgenommen. Eine andere Möglichkeit ist, den Schlüssel über einen LDAP-Server zu beziehen. Einen solchen LDAP-Server haben wir in unserer Projektarbeit eingerichtet.

6 Zertifikate

6.1 OpenSSL installieren

Für die Beschreibung der OpenSSL-Installation beachten Sie bitte den folgenden Link:
„www.pca.dfn.de/dfnpca/certify/ssl/handbuch/ossl095„.

Die Konfigurationsdatei „openssl.cnf“, in welchem die Voreinstellungen für die Zertifikaterstellung gemacht werden, finden sie im Anhang. Eine CRL haben wir im ersten Schritt noch nicht vorgesehen, deshalb sind die entsprechenden Einträge in der Datei als Kommentar markiert.

6.2 Erstellen eines Zertifikats

Das Erstellen der Zertifikate ist im OpenSSL-Handbuch „www.pca.dfn.de/dfnpca/certify/ssl/handbuch“ beschrieben. Es empfiehlt sich das Listing der Konfigurationsdatei „openssl.cnf“, welche viel Kommentar und die Voreinstellungen zum Erstellen eines Zertifikats enthält, im Anhang anzuschauen.
Wir erstellten ein Rootzertifikat mit einer Schlüssellänge von 2048Bits. Die Benutzerzertifikate haben eine Schlüssellänge von 1024. Eine CRL (Certificate Revokation List) haben wir keine angegeben.

6.3 Erstellen einer *.p12 Datei

Das p12-Format ist ein Format für das Einlesen von Zertifikaten in den Browser.
Hier die Befehle um eine p12-Datei zu erzeugen:

```
openssl pkcs12 -chain -export -name „Name“ -in Zertifikat.pem -inkey privaterSchlüssel.pem -out name.p12
```

Beispiel.:

```
openssl pkcs12 -chain -export -name „Stephan Zehnder“ -in ZehnderCert.pem
-inkey private/Zehnderkey.pem -out zehnder.p12
```

6.4 Einlesen eines Zertifikats über LDIF

6.4.1 Einleitung

Wenn man mit OpenSSL ein Zertifikat erstellt, wird eine *.pem Datei erzeugt. Diese Datei kann aber nicht direkt ins LDIF-Format übernommen werden, sondern muss zuerst in ein Base64 Format umgewandelt werden. Das pem-Format ist zwar schon ein Base64-Format, es enthält aber noch das ganze Zertifikat in Textform. Aus diesem Grund muss die *.pem Datei zuerst in eine *.asc Datei umgewandelt werden. Eine Beispiel *.ldif Datei finden sie im Anhang. Es wäre auch möglich, das Zertifikat aus der pem-Datei zu kopieren und so in die LDIF-Datei kopiert werden. Dieser Schritt könnte natürlich auch automatisiert werden durch ein Skript.

6.4.2 Umwandeln einer *.pem Datei in eine *.asc Datei

Das Umwandeln geschieht in zwei Schritten. Erstens wird mittels x509 die *.pem Datei in eine *.der Datei umgewandelt. Im zweiten Schritt wird die *.der Datei mit mmencode in eine *.asc Datei umgewandelt.

6.4.3 Einfügen eines Zertifikats in die *.ldif Datei

Das Zertifikat wird mittels des Attributes `userCertificate` in die LDIF Datei eingetragen. Es muss dabei der Subtype binary verwendet werden.
Hier ein Ausschnitt einer LDIF-Datei mit eingetragenem Zertifikat:

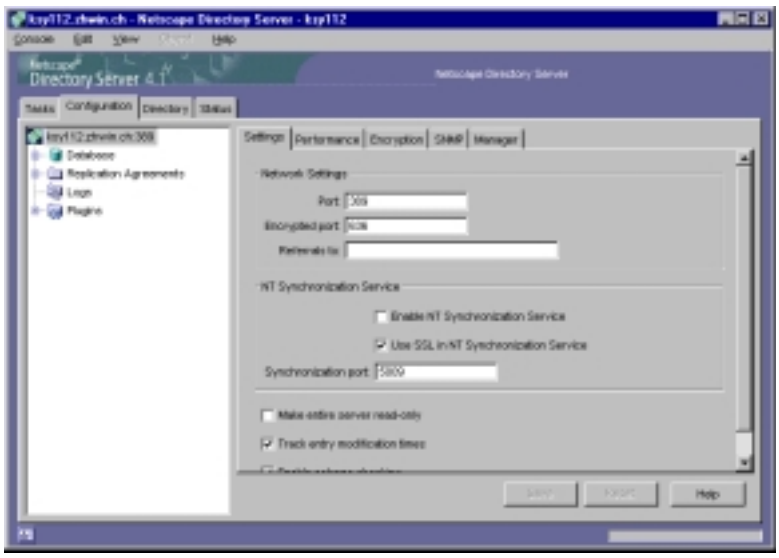
```
userCertificate;binary:: MIIE8DCCA9igAwIBAgIBAjANBgkqhkiG9w0BAQQFADCBwTELM
 AkGA1UEBhMCQ0gxEDAOBgNVBAgTB1p1ZXJpY2gxEzARBgNVBAcTCldpbnRlcnRodXIxJzAlB
 gNVBAoTHlp1ZXJjaGVyIEhvY2hzY2h1bGUgV2ludGVydGh1cjEgMB4GA1UECxMXSW5mb3Jt
 YXRpb25zdGVjaG5vbG9naWUxFDASBgNVBAMTC1pIVyBSb290IENBMSowKAYJKoZIhvcNAQkBFh
```

```
tzdGVwaGFuLnplaG5kZXJAYmlnZm9vdC5jb20wHhcNMDAwMzIxMTUwMTAxWhcNMDEwMzIxMTUw
MTAxWjCBujELMAkGA1UEBhMCQ0gxEDAOBgNVBAgTB1p1ZXJpY2gxEzARBgNVBAcTCldpbnRlcn
RodXIxJzAlBgNVBAoTHlp1ZXJjaGVyIEhvY2hzY2h1bGUgV2ludGVydGh1cjEgMB4GA1UECxMX
SW5mb3JtYXRpb25zdGVjaG5vbG9naWUxFzAVBgNVBAMTDkdyaWVkZXIgTWFya3VzMSAwHgYJKo
ZIhvcNAQkBFhFpN2dyaWVkZUB6aHdpbi5jaDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCg
```

Zu beachten ist, dass zwischen dem Attribut und dem Subtype ein Strichpunkt und hinter dem Subtype zwei Doppelpunkte stehen. Ausserdem ist vor jeder Zeile des Zertifikats ein Space einzufügen, oder das Zertifikat auf eine Zeile geschrieben werden.

6.4.4 Übernahme des Zertifikates in den LDAP-Server

Logen sie sich als Administrator ein und gehen sie auf den Directory Server (lesen sie dazu das Kapitel 7.4.1 und 7.4.2). Schauen sie dann unbedingt, dass das Register „Configuration" offen ist, nur dann können sie unter „Console", „Import" eine LDIF-Datei einlesen.



6.5 Übernahmemechanismus

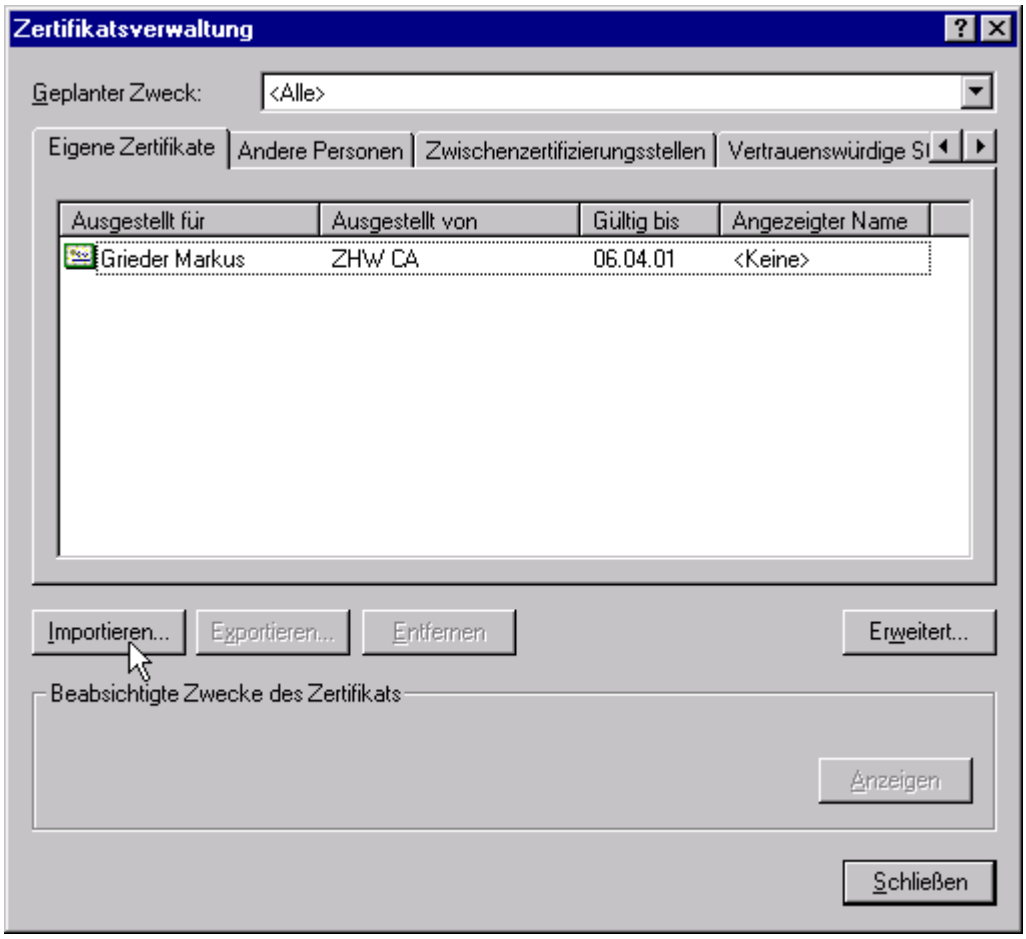
6.5.1 Internet Explorer

Damit die Zertifikate zum Beispiel für den sicheren E-Mail-Verkehr gebraucht werden können, welche eine hohe Verschlüsselung erfordern, muss zuerst das „High Encryption Pack"[1] von Microsoft installiert werden. Damit werden die Verschlüsselungsroutinen von 56-Bit auf 128-Bit aufgerüstet.
Die folgenden Beschreibung beziehen sich alle auf den Internet Explorer Version 5.x.

6.5.1.1 Eigene Zertifikate

Ein eigenes Zertifikat wird am einfachsten im PKSC#12 Format importiert. Unter Windows ist normalerweise die Endung *.p12 oder *.pfx mit dem Internet Explorer verbunden, womit beim Öffnen der Datei automatisch der Import-Wizard gestartet wird. Der Import eines Zertifikats kann jedoch auch über den Dialog Internetoptionen vom Internet Explorer oder der Systemsteuerung gestartet werden, indem im Register „Inhalt" mit der Schaltfläche „Zertifikate" die Zertifikatsverwaltung geöffnet wird und die Schaltfläche „Importieren" gedrückt wird.

[1] „High Encryption Pack" von Microsoft. Update des Internet Explorer von 56-Bit auf 128-Bit.
„http://www.microsoft.com/windows/ie_intl/de/security/sgc.htm„ (deutsche Update)
„http://www.microsoft.com/windows/ie/download/128bit/intro.htm„ (andere Sprachen)



Wenn der Importvorgang über die Zertifikatsverwaltung gestartet wurde, muss zuerst der Dateinamen des Zertifikat angegeben werden. Beim Durchsuchen des Dateisystems in diesem Dialog werden standardmässig nur Dateien mit der Endung *.pfx angezeigt. Dennoch werden alle Zertifikate in den Formaten PKSC#12 + PKSC#7 und DER problemlos importiert, egal welche Endungen die Dateien haben.
Je nach Typ des Zertifikat wird dieses nun automatisch in einem der Zertifikatsspeicher wie „Eigene Zertifikate“ oder „Andere Personen“ abgelegt.

6.5.1.2 Fremde Zertifikate

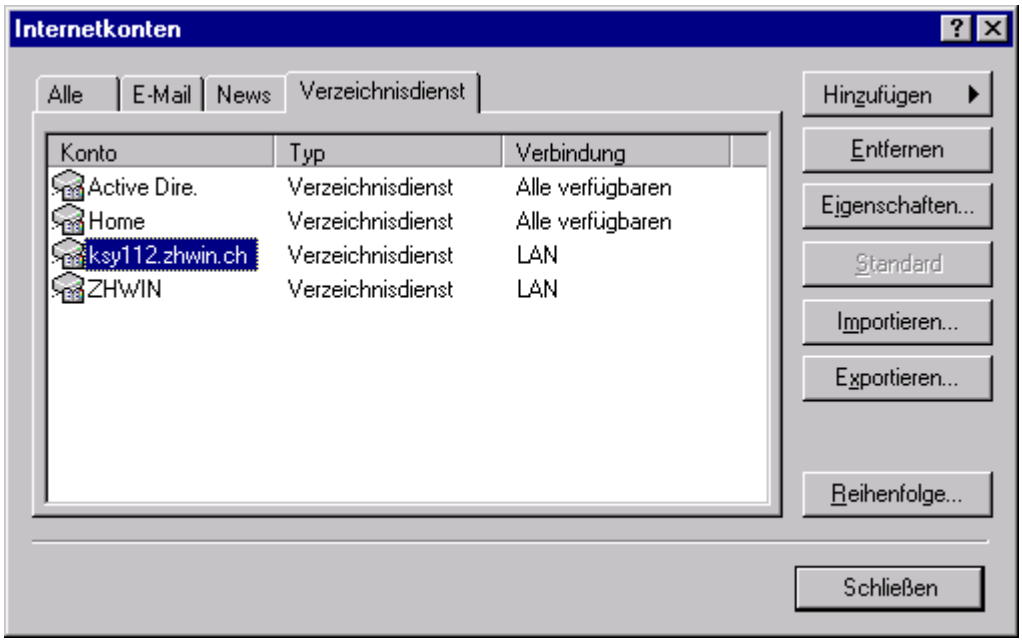
Ein Stamm-Zertifikat oder ein Zertifikat einer anderen Person, welche als Datei im DER oder PKSC#7 Format vorliegt, kann mit dem gleichen Vorgehen wie bei dem eigenen Zertifikat importiert werden. Das Stamm-Zertifikat ist nach dem Importieren automatisch für die E-Mail-Verschlüsselung, für welche es gedacht ist, gebrauchbar. Dies im Unterschied zu Netscape, wo zuerst die Einsatzmöglichkeiten aktiviert werden müssen.

6.5.1.2.1 Import über Webseiten

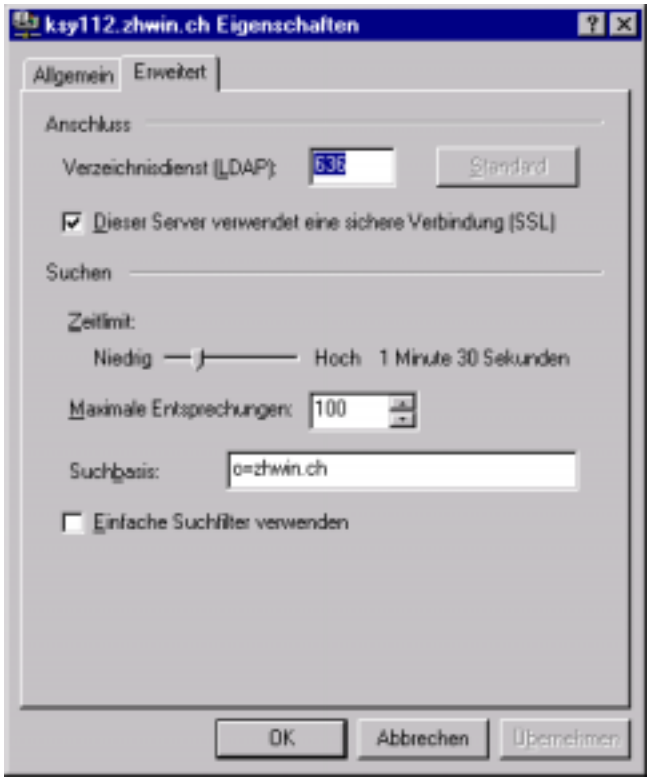
Damit der Internet Explorer bei einer Anfrage ein DER-codierten Zertifikat als solches erkennt, muss der Mime-Typ application/pkix-cert zurückgeliefert werden. Bei einem PKSC#12 Zertifikat erwartet er den Mime-Typ application/x-pkcs12. Bei einem PKSC#7 Zertifikat erwartet er den Mime-Typ application/x-pkcs7-mime. Dieses Verhalten ist leider unterschiedlich zum Netscape.

6.5.1.2.2 Import über LDAP

Das importieren eines fremden Zertifikats über LDAP erfolgt mit Hilfe des Adressbuches. Als erstes muss jedoch der LDAP-Server angegeben werden. Zu diesem Zweck muss ein Verzeichnisdienst Konto erstellt werden. Dies kann bei Outlook Express über den Dialog „Extras“ „Konten“ gemacht werden.



Wählen sie Hinzufügen->Verzeichnisdienst. Es erscheint dann der Assistent zum Einrichten eines Verzeichnisdienstes. Nach dem Einrichten muss noch zusätzlich die Suchbasis eingestellt werden. Wählen sie hierzu die Schaltfläche „Eigenschaften“ und dann das Register „Erweitert“. Nun geben sie die Suchbasis ein. Die Suchbasis ist normalerweise der `dn` (`distinguished name`) des Root-Eintrags.



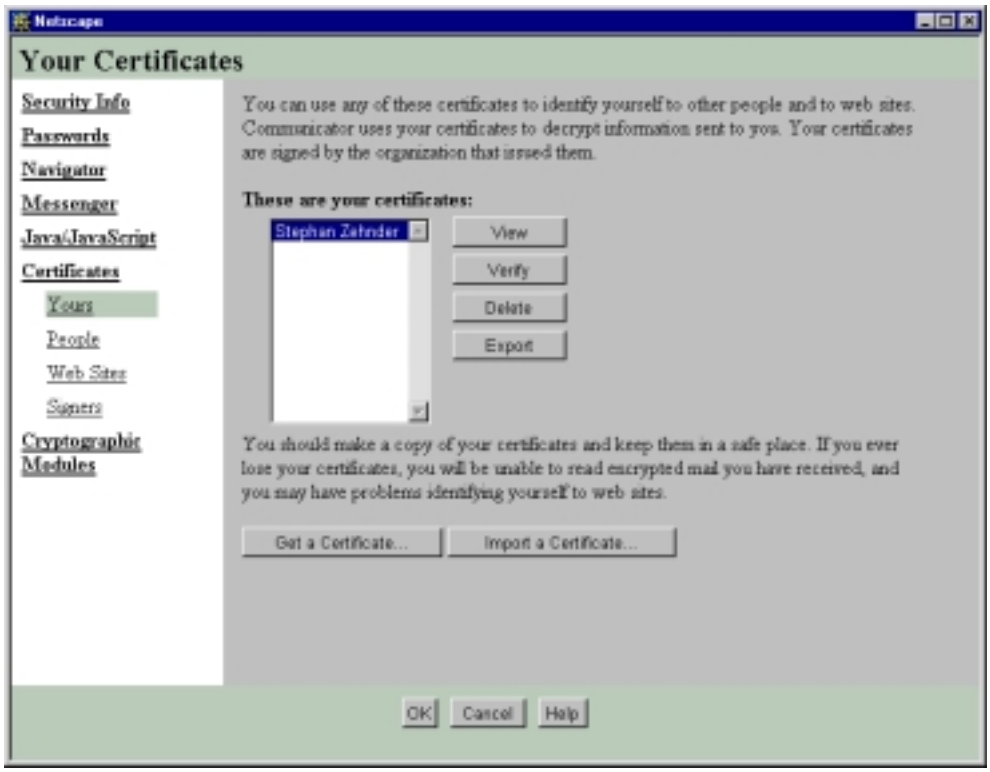
Nun können sie Personen suchen, indem sie im Adressbuch die Schaltfläche „Personen suchen“ betätigen. Wählen sie den zuvor eingerichteten LDAP-Server, geben sie E-Mail oder Name der Person ein und starten sie die Suche. Falls ein Eintrag gefunden wurde, wird dieser mit seinen Attributen angezeigt. Durch anwählen eines Eintrages wählen der Schaltfläche „Hinzufügen“, wird das Zertifikat übernommen.

6.5.2 Netscape

Folgende Beschreibungen beziehen sich auf den Netscape Communicator Version 4.x

6.5.2.1 Eigene Zertifikate

Netscape kennt ein eigenes Tag <keygen> für die Generierung von Schlüssel, beziehungsweise Zertifikate, über eine Webseite. Es ist also kein zusätzlicher Download dafür nötig. Ein Benutzer kann seine Schlüssel also selber lokal generieren, dies ist von der Sicherheit her sicher sinnvoll. Diese Schnittstelle wird zum Beispiel bei dem Softwarepaket OpenCA verwendet, welches in dieses Dokumentation vorgestellt wird.
Ein eigenes Zertifikat wird am einfachsten im PKSC#12 Format importiert. Diese Dateien tragen die Endung *.p12. Der Import einer solchen Datei geschieht in Netscape über den Dialog „Security", mit dem Symbol eines Schlosses.
Zum Importieren eines eigenen Zertifikats muss man in der linken Spalte „Yours" wählen und dann die Schaltfläche „Import a Certificate..." betätigen. Danach kann die *.p12 Datei angegeben werden und das Zertifikat wird importiert.



6.5.2.2 Fremde Zertifikate

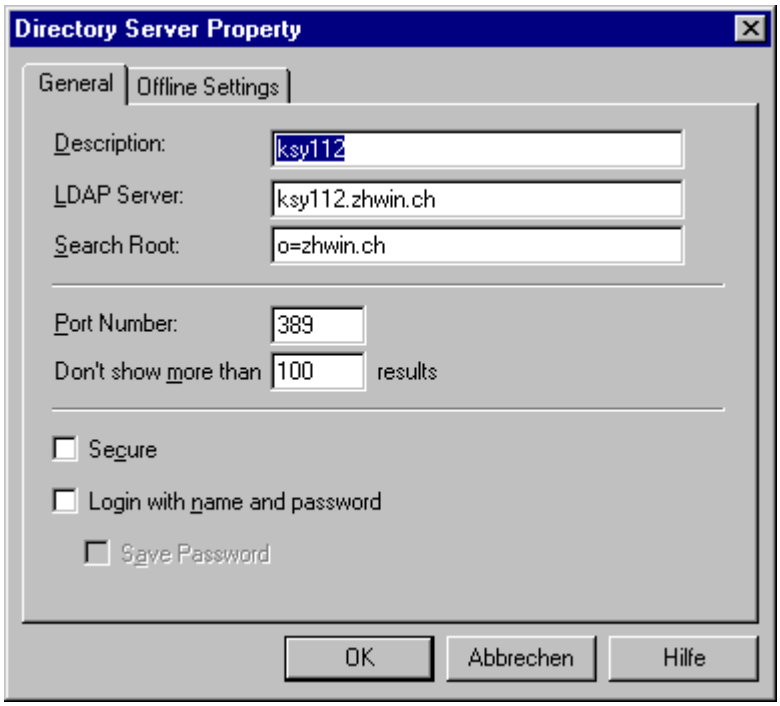
6.5.2.2.1 Import über Web

Damit der Netscape bei einer Anfrage für ein DER-codiertes Stamm-Zertifikat es als solches erkennt, muss der Mime-Typ „application/x-x509-ca-cert"zurückgeliefert werden. Bei einem User-Zertifikat braucht der Netscape den Mime-Type „application/x-x509-XXX-cert". Bei einer CRL wäre der Mime-Type „application/pkcs7-crl".

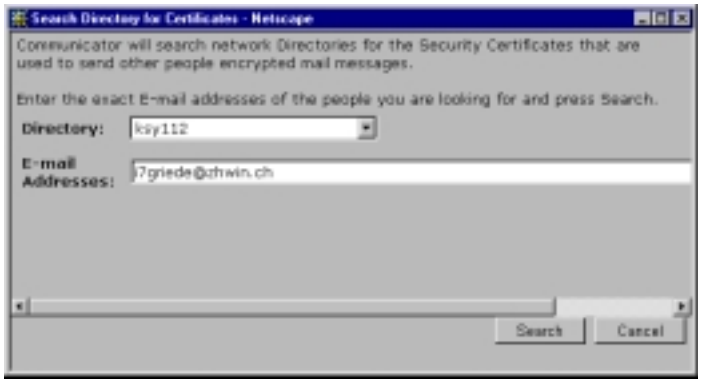
6.5.2.2.2 Import über LDAP

Das Importieren eines fremden Zertifikats geschieht bei Netscape über das Security Fenster. Bevor dies möglich ist, muss der LDAP-Server allerdings zuerst im Browser eingerichtet werden. Das Einrichten eines LDAP-Servers ist nicht über das Security Fenster möglich, sondern wird im Adressbuch vorgenommen. Öffnen sie also als erstes das Adressbuch. Klicken sie mit der rechten Maustaste auf die Liste mit den bereits vorhandenen Directories und wählen sie „New Directory...".

In diesem Fenster wird dann der Name und der Pfad des Servers, sowie der Search Root (Suchbasis) eingestellt. Die Suchbasis ist normalerweise der `dn` (`distinguished name`) des Root-Eintrags.



Nun können sie ein Zertifikat suchen, mittels des Security Fensters. Wählen sie in der linken Spalte „People“ und dann die Schaltfläche „Search Directory“.



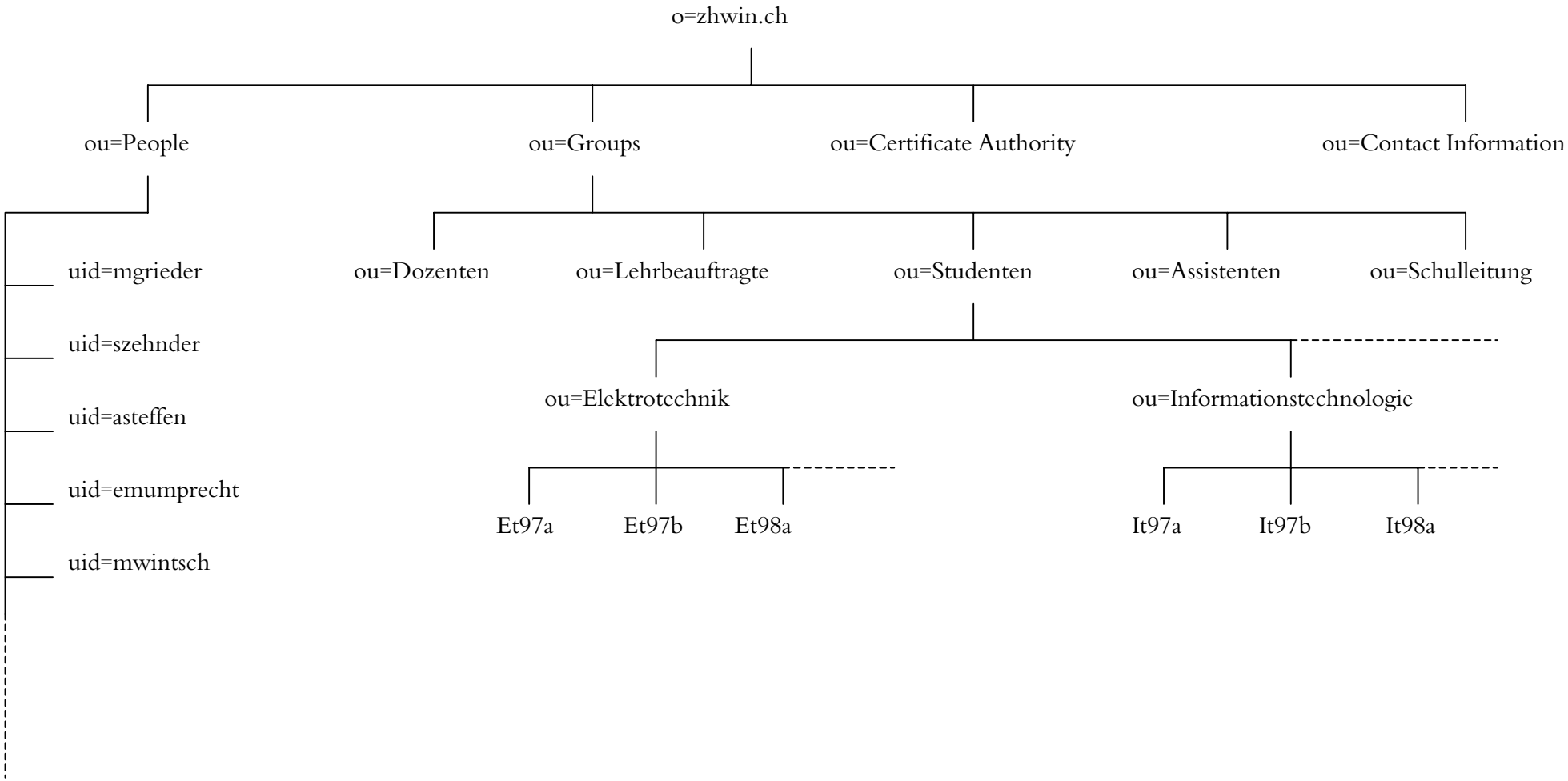
Wählen sie nun den von ihnen zuvor eingestellten LDAP-Server, geben sie die E-Mail-Adresse ein und betätigen sie „search“. Falls das Zertifikat vorhanden ist, erscheint eine Meldung und das Zertifikat wird importiert.

7 Design und Implementierung des LDAP-Schemas

7.1 Überlegungen zum Design des Schemas

Im Deployment Guide zum Netscape Directory Server wird empfohlen die Directory-Baumstruktur möglichst flach zu halten. Allerdings ist zu sagen, dass eine zu flache Baumstruktur nichts zur Übersichtlichkeit beiträgt. Man könnte natürlich sagen, dass zum Beispiel in unserem Schema die Verzweigung mit den Studiengängen überflüssig ist, da die Studienrichtung ja aus dem Kürzel der Klasse ersichtlich ist. Dies stimmt, jedoch könnte man dann nicht gezielt alle Studenten einer bestimmten Studienrichtung einfach abfragen, indem man nach dem Attribut ou=Studienrichtung sucht.
Wichtig ist, dass man keine Referenzen auf Strukturen oder Namen hat, die häufig ändern. So wäre es ungünstig das Klassenkürzel abhängig vom Studienjahr zu machen, da in jenem Fall jedes Jahr der Name der Klasse ändern würde.
Wichtig war für uns, dass man Änderungen nur an einem Ort vornehmen muss, damit nicht für die gleiche Person verschiedene Einträge vorhanden sind. Wir haben das folgendermassen erreicht: Alle Personen an der ZHW, also egal ob Student, Dozent, Direktor oder Assistent, sind im Ast People eingetragen. Die genauere Einteilung erfolgt durch sogenannte dynamische Gruppen. Jeder Dozent zum Beispiel hat in seinem Eintrag unter People den Eintrag ou=Dozenten (ou entspricht einem Ast, oder einer Verzweigung). Unter dem Ast Dozent sind nun nicht alle Dozenten aufgelistet, sondern eine Abfrage nach dem Attribut ou=Dozenten, auf diese Weise sind alle Dozenten dynamisch unter dem Ast Dozenten eingetragen. Wenn nun ein neuer Dozent angestellt wird, erhält dieser einen Eintrag im Ast People mit dem entsprechenden Attribut ou=Dozenten. Im Ast Dozent, muss dann effektiv nie etwas geändert werden, die Abfrage unter dem Ast Dozenten liefert immer die aktuellen Daten aller zur Zeit im Ast People eingetragenen Dozenten. Analog funktioniert es auch mit Assistenten und Studenten, oder anderen Gruppen von Organisationen.
Die Mitglieder der Gruppe Directory Administrators unter dem Ast Groups sind statisch eingetragen. Es können also nur die Directory-Administratoren Änderungen darin vornehmen. Dies ist wichtig, da sich sonst eine unberechtigte Person die Rechte verschaffen könnte, indem sie bei sich das Attribut ou=DirectoryAdministrators einfügt, um den Directory-Server zu verwalten.
Weiter im Baum enthalten sind noch die Äste (oder vielleicht besser Blätter), Certificate Authority, welche den öffentlichen Schlüssel der ZHW enthält, und Contact Information, welche Kontaktadressen und Telefonnummern enthält für alle Personen, die aus der ganzen Welt auf das Directory zugreifen.

7.2 Schema

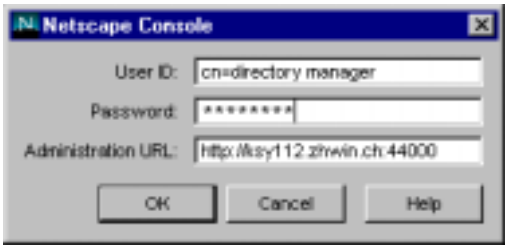

o=zhwin.ch
ou=People
ou=Groups
ou=Certificate Authority
ou=Contact Information
uid=mgrieder
uid=szehnder
uid=asteffen
uid=emumprecht
uid=mwintsch
ou=Dozenten
ou=Lehrbeauftragte
ou=Studenten
ou=Assistenten
ou=Schulleitung
ou=Elektrotechnik
ou=Informationstechnologie
Et97a
Et97b
Et98a
It97a
It97b
It98a

7.3 Schema implementieren

Die Implementierung des Schemas kann komfortabel mit der „Netscape Console“ erfolgen. Die Console besitzt eine schöne graphische Oberfläche, mit welcher man das Directory einfach aufbauen kann. Die Console gibt es sowohl für Windows als auch für Linux. Da die Version unter Windows einige Fehler aufweist, empfehlen wir die Version für Linux zu verwenden.

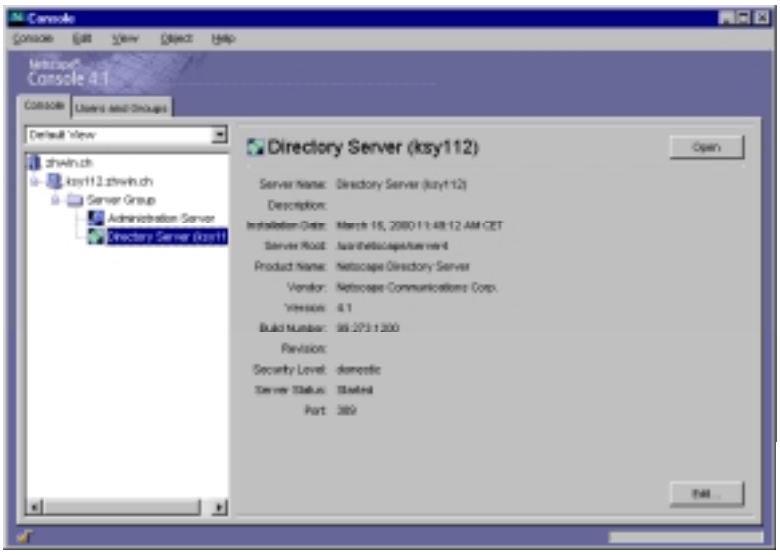
7.3.1 Einloggen

Am besten man loggt sich als Directory Manager ein, welcher noch mehr Rechte besitzt als der Administrator. Die Abkürzung cn steht für common name und ist notwendigerweise einzugeben. Bei der Administration URL muss noch der Port für den Administrations-Server stehen, in unserem Fall 44000.



7.3.2 Schema aufbauen

Nach dem Einloggen erscheint folgendes Fenster:



Es gibt nun zwei Möglichkeiten um weiter zu fahren. Entweder man klickt auf das Register Users and Groups und kreiert das Schema, oder man klickt auf den oben Dunkelblau markierten Eintrag Directory Server. Wir haben die zweite Möglichkeit gewählt und möchten deshalb diese Möglichkeit genauer beschreiben. Falls sie am richtigen Ort die Maustaste betätigt haben, erscheint in etwa das folgende Fenster:

Sie sehen, dass das Register „Directory“ angewählt wurde. Um einen neuen Eintrag zu erstellen wählen sie, wo im Baum der Eintrag plaziert werden soll und betätigen die rechte Maustaste. Auf dem erscheinenden DropDown-Menü wählen sie „New2 und dann den entsprechenden Eintrag.

7.3.3 Erstellen eines Personeneintrags

Wählen sie den Eintrag „User“ im DropDown-Menü des oben stehenden Bildes. Bitte beachten sie, dass User in unserem Fall nur unter „People“ einzutragen sind. Füllen sie die benötigten Attribute aus. Falls nicht alle benötigten Attribute angezeigt sind, wählen sie „Advanced“ und dann unter „View“, „show all Attributes“. Wichtig: Vergessen sie nicht die Einträge für die `ou`'s vorzunehmen. Für jede Organizational Unit und Group, welcher die Person angehört, muss ein Eintrag stehen. Ein Student der Klasse It97a hätte in unserem Schema folgende Einträge:

ou=People
ou=Groups
ou=Studenten
ou=Informatinostechnologie
ou=It97a.

Sind diese Einträge nicht richtig, kriegen sie Probleme beim Definieren der dynamischen Mitglieder einer Gruppe.

7.3.4 Group oder Organization Unit?

Jede Verzweigung ist als Organizational Unit zu realisieren. Enden können je nach Bedarf als Organizational Unit oder als Group realisiert werden. Hat ein Ende Mitglieder einer zusammengehörigen Organisationseinheit, z.B. einer Klasse, so sollte sie als Group realisiert werden. Beispiele für Organizational Units am Ende wären zum Beispiel das „Astende Contact Information“, es hat keine Mitglieder sondern enthält nur Addresse und Telefonnummern, und die „Organisation People“, hier sind alle Personen des gesamten Directorys eingetragen und in diesem Sinne keine Gruppe.

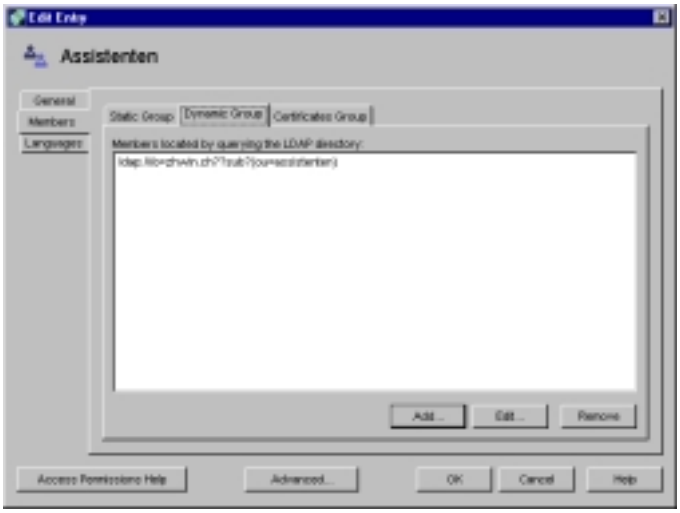
7.3.5 Erstellen einer Organizational Unit

Wählen sie den Eintrag Organizational Unit im DropDown-Menü des oben stehenden Bildes. Es erscheint ein Fenster in dem nur der Name der Organizational Unit einzutragen ist.

7.3.6 Erstellen einer Group und definieren der Member

Wählen sie den Eintrag Group im DropDown-Menü des oben stehenden Bildes. Es erscheint ein Fenster in dem nur der Name der Gruppe einzutragen ist.
Um die Mitglieder zu definieren, wählen sie das Register Members im aktuellen Fenster. Es gibt zwei Arten von Mitgliedern, statische und dynamische. Bei statischen Mitgliedern funktioniert die Zuordnung über die `uid`, bei dynamischen über eine LDAP-Abfrage welche die Mitglieder als

Ergebnis liefert. Der Vorteil bei einer dynamischen Zuweisung ist, dass bei Änderungen, zum Beispiel beim Löschen oder Hinzufügen von Personen unter People, die Mitglieder der Gruppe nicht angepasst werden müssen, falls die ou Einträge der Personen stimmen. Definieren sie deshalb nur die Mitglieder einer sicherheitsrelevanten Gruppe statisch, z.B. Mitglieder der Gruppe Directory Administrators. Würden die Mitglieder der Gruppe Directory Administrators statisch zugeordnet, d.h. mit der LDAP-Abfrage ldap://ksy112.zhwin.ch/o=zhwin.ch??sub?(ou=Directory Administrators), könnte eine Person sich die Administratorrechte besorgen, indem sie bei sich den Eintrag „ou=Directory Administrators“ hinzufügt und damit Mitglied der Gruppe ist.
Um die Mitglieder zu definieren wählen sie das Register Members. Es erscheint das folgende Fenster:



Nun wählen Sie Dynamic Group oder Static Group um die Mitglieder zu defninieren. Indem sie Add klicken können sie entweder einen Member bei statischen Gruppen oder eine LDAP-Abfrage bei dynamischen Gruppen hinzufügen.

7.4 Vergeben der Rechte

Nach dem Installieren des Netscape Directory Servers, sind viele Rechte bereits eingestellt. Es ist am Anfang relativ schwer den Überblick zu gewinnen. Das definieren der Rechte selbst ist mit der grafischen Oberfläche der Console möglich. Es können Regeln für alle Einträge gesetzt werden, also sowohl für einzelne User wie für ganze Gruppen oder Organizational Units.

7.4.1 Einige wichtige Hinweise zur Security Planung von Netscape

Wenn zwei verschiedene Definitionen für ein Objekt bestehen, gilt jene, welche den Zugriff verweigert. Es wird deshalb empfohlen, Definitionen für möglichst kleine Teile vorzunehmen. Man kann zum Beispiel grundsätzlich für alle Benutzer alle Rechte erteilen, aber weitere Definitionen vornehmen, welche schreiben auf Attribute für alle, ausser die Administratoren, verbietet. Um die Administration der ACL's zu vereinfachen, sollte man die Regeln zu so grossen Gruppen wie möglich zusammenfassen. Es ist besser die Regeln an der Wurzel des Baumes zu definieren, als an allen Ästen zu verteilen.
Um das LDAP-Verzeichnis möglichst aktuell zu halten, empfiehlt Netscape Attribute wie Telefonnummern und Adressen self-writable zu machen. Das bedeutet zum Beispiel, dass ein Student seine Telefonnummer und seine Adresse selber anpassen kann. Überprüft werden ob die Änderung durchgeführt wurde muss dann aber trotzdem.

7.4.2 Attribute die alle haben

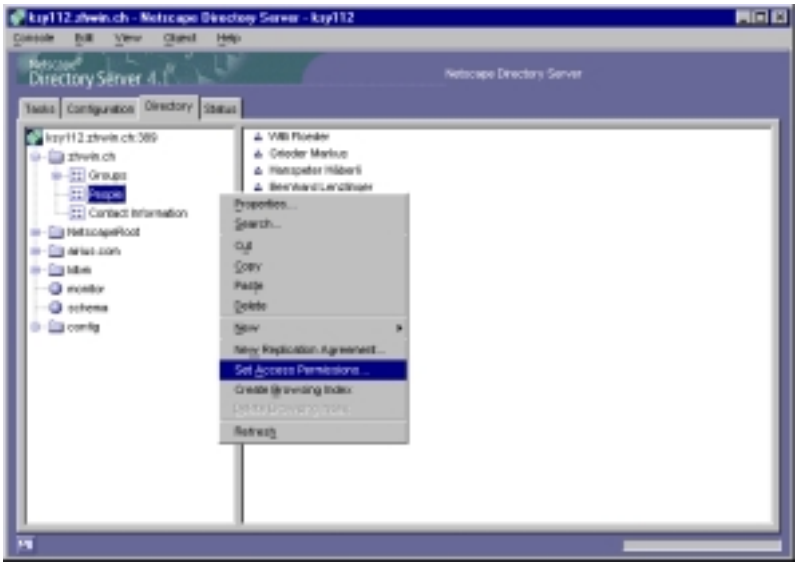
Eine Möglichkeit wäre, Username und Passwort auch zu speichern und dann voraussetzen, dass man einloggen muss, um auf Daten zugreifen zu können. Die Kontaktinformationen sollten für alle Lesezugriff erlauben.

verwendete Klassen: top, person, organizationalPerson, inetOrgPerson

Sinnvoll ist möglicherweise auch noch die Klasse ntUser, welche Entitäten im Zusammenhang mit einem WindowsNT-Netzwerk definiert. Sie enthält Attribute wie der Pfad des Home-Verzeichnisses oder das LoginScript.

ou=Groups

Datenname	**Klasse**	**Attribut**	**Self read/write**	**Global read**
Nachname	person	Sn	Read-only	Yes(anonymous)
Vorname	inetOrgPerson	GivenName	Read-only	Yes(anonymous)
Common Name	person	Cn	Read-only	Yes(anonymous)
Adresse	inetOrgPerson	HomePostalAddress	Read/Write	No
Telefonnummer	inetOrgPerson	HomePhone	Read/Write	No
E-Mail	inetOrgPerson	Mail	Read-only	Yes(anonymous)
Zertifikat	inetOrgPerson	UserCertificate	Read-only	Yes(anonymous)



7.4.3 zusätzliche Attribute für Studenten

ou=Studenten

Datenname	**Klasse**	**Attribut**	**Self read/write**	**Global read**
Studienrich-tung	organizationalPerson	Ou	Read-only	Yes(anonymous)
Klasse	organizationalPerson	Ou	Read-only	Yes(anonymous)

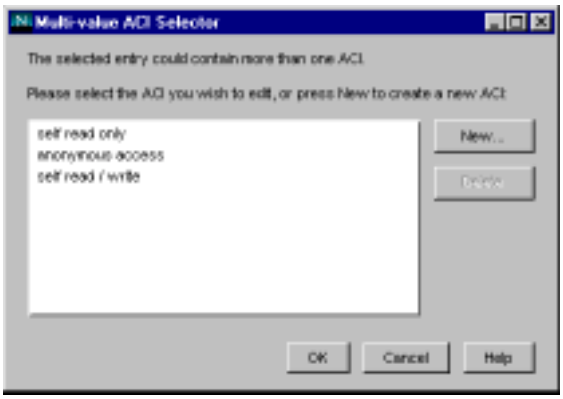
7.4.4 zusätzliche Attribute für Dozenten und Assistenten

ou=Dozenten bzw. ou=Assistenten

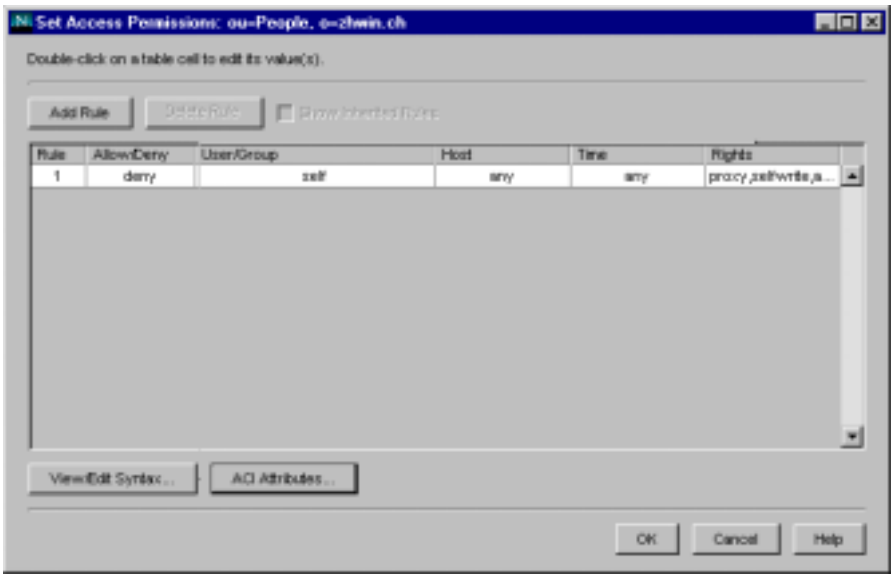
Datenname	**Klasse**	**Attribut**	**Self read/write**	**Global read**
Zimmernum-mer	inetOrgPerson	RoomNumber	Read-only	Yes(anonymous)
Telefon ZHW	person	TelephoneNumber	Read-only	Yes(anonymous)

7.5 Setzen der Rechte

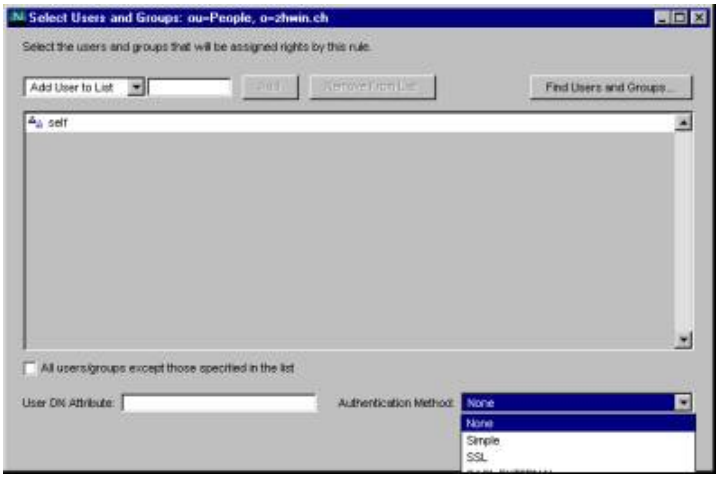
Um die Rechte für ein Objekt zu setzten, klickt man mit der rechten Maustaste darauf und wählt beim erscheinenden Drop-Down-Menü den Punkt „Set Access Permissions“. Danach erscheint das folgende Fenster:



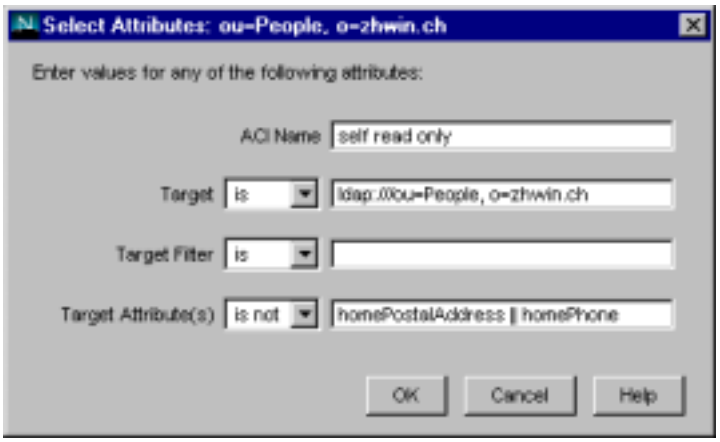
Hier sieht man die bereits für dieses Objekt gesetzten Rechte., das heisst die vorhandenen ACI's. Um eine Regel zu löschen wählt man „Delete“. Eine neue Regel für dieses Objekt zu setzten man mit „New“. Wählt man „New“, erscheint das folgende Fenster:



In diesem Fenster wird die Regel definiert. Man kann mehr als eine Regel pro ACI definieren, dies wird jedoch nicht empfohlen. Besser ist, wenn man für jede neue Regel eine neue ACI macht. Das heisst, man wählt für jede neue Regel im vorherigen Fenster „New“.
Durch anklicken mit der Maus unter Rights erhält man eine Liste mit verschiedenen Rechten. Diese können durch Mausklick ausgewählt werden. Danach wählt man durch Mausklick unter dem Punkt „Allow/Deny“, ob die angewählten Möglichkeiten erlaubt sind oder nicht. Nun definiert man noch, für wenn die Regel gilt. Dafür wählt man den Punkt „User/Group“, worauf das folgende Fenster erscheint:



Hier wird zuerst gewählt, ob die vorher definierte Regel für eine Gruppe oder einen Benutzer gilt. Danach gibt man im Feld daneben den Namen ein. Falls die „self read/write“ Rechte gesetzt werden, schreibt man „self“ ins Feld. Es können auch Benutzer oder Gruppen gesucht werden. Unten rechts kann noch die Authentifizierungsmethode gewählt werden. Unten Links kann ausserdem noch eingestellt werden, dass die Rechte für allem, ausser für die unter der Liste oben aufgeführten Benutzer, gelten sollen.
Ist die Gruppe oder der Benutzer dazugefügt worden, wählt man im vorhergehenden Fenster „ACI Attributes“, es erscheint folgendes Fenster:



In diesem Fenster sollte man unter „ACI Name“ einen sinnvollen Namen vergeben, der die Regel beschreibt. Unter „Target“ und „Target Filter“ muss man nichts ändern. Beim Punkt „Target Attribute(s)“ können die Attribute gewählt werden, für welche die Regel gelten soll. Zwischen den einzelnen Attributen stehen die Zeichen „||“ für eine Oder-Verknüpfung.

7.6 Austesten der gesetzten Rechte

Das Austesten der Rechte war leider nicht ausführlich genug möglich. Einfache Rechte, wie das Verbot zum Anschauen von Informationen einer Gruppe, können geprüft werden.
Für die Administration der Zugriffsrechte in der Praxis, wären zusätzliche Werkzeuge nötig. Man könnte zum Beispiel ein Skript schreiben, welches die Zugriffsrechte überprüft.

7.7 Offene Fragen

Offene Fragen haben wir in Bezug auf das Setzen der Rechte.
Die Netscape Console erlaubt das separate Setzen der Authentifizierungs-Methode für jede einzelne Gruppen. Ich kann zum Beispiel definieren, dass auf die Gruppe „people“ nur genau eine Person Zugriff haben soll und keine Authentifizierung wählen. Wie kann dann aber sichergestellt werden, dass nur diese Person zugreift? Die Definition ist also in sich ein Widerspruch.

8 Netscape Directory Server

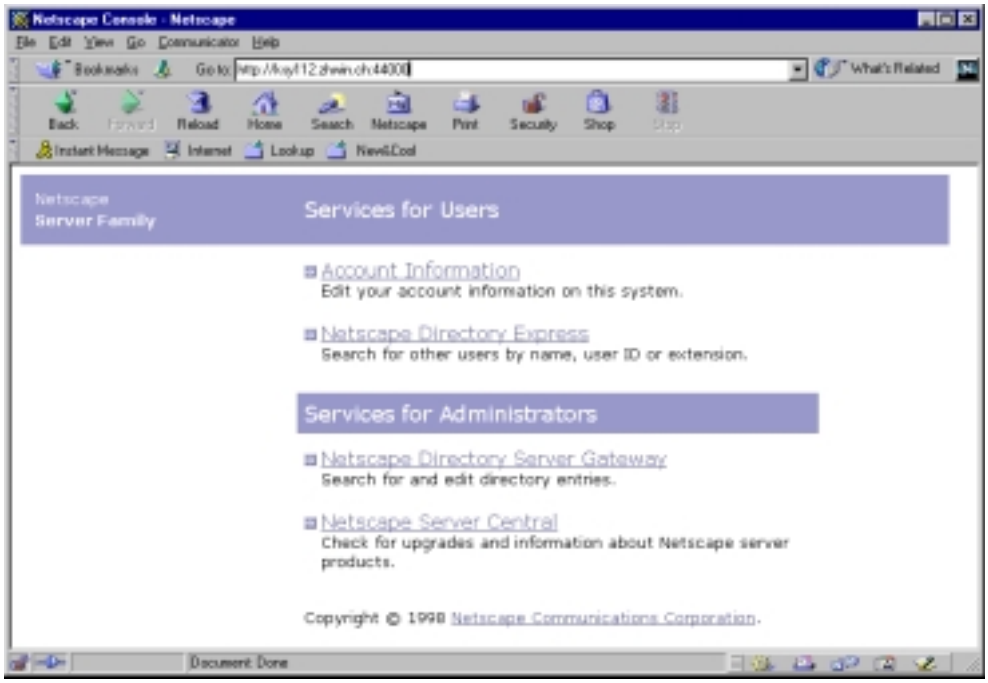
8.1 Installation Netscape Directory Server

Da die Installation des Netscape Directory Server recht gut in der bestehenden Dokumentation von Netscape beschrieben, gehen wir hier nicht genauer darauf ein. Bei unserer Installation traten keine nennenswerte Probleme auf.

8.2 Der Netscape Directory Server Gateway

Hier einige Erläuterungen zum Netscape Directory Server Gateway. Der Gateway bietet eine einigermassen komfortable Schnittstelle zum LDAP-Server. Um seine Möglichkeiten zu lernen braucht es jedoch ein bisschen Einarbeitungszeit.

8.2.1 Was ist der Netscape Directory Server Gateway?



Einerseits kann man mit dem Gateway Informationen suchen, andererseits kann ein Benutzer auch seine Daten anpassen.
Der Gateway besteht aus verschiedenen CGI-Programmen und HTML-Templates und lässt sich leicht erweitern. Siehe dazu das Dokument „Gateway Customization Guide“ von Netscape (gwgust.pdf).

8.2.2 Wie greift man auf den Gateway zu

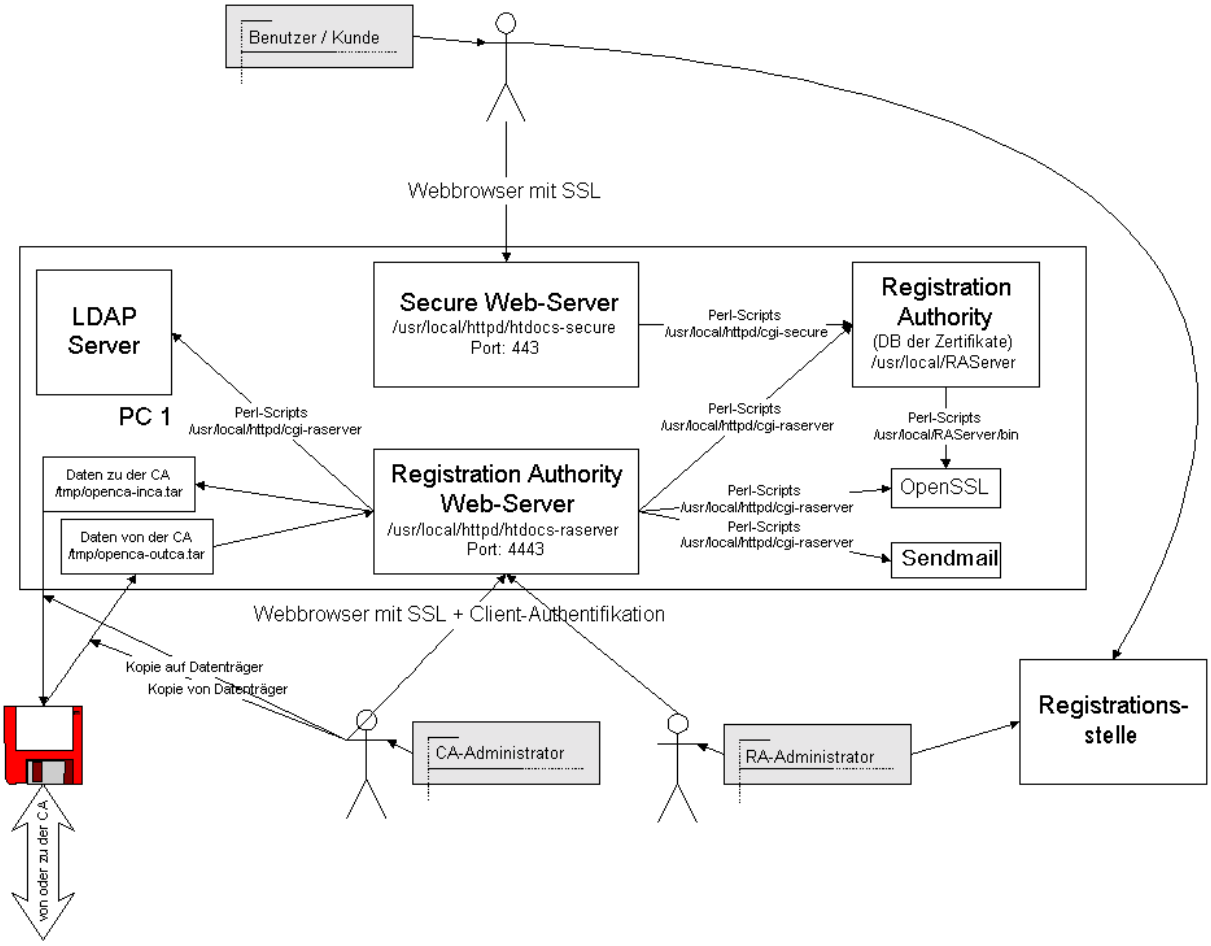
Der Zugriff auf den Gateway erfolgt über einen Browser. Dies hat den Vorteil, dass keine speziellen Programme installiert werden müssen, um den Benützern Zugriff auf ihre Daten zu gewähren.
Der Zugriff auf den Gateway ist einfach. Man startet einen Browser und gibt die URL des Gateways ein. Zu beachten ist, dass es keine LDAP-URL ist, sondern eine HTTP-URL.
Die URL für unseren Gateway lautet: „http://ksy112.zhwin.ch:44000“. Der Port ist derselbe wie der Administrations Server.
Um Änderungen vornehmen zu können, muss man sich mit „uid“ oder „cn“ und Passwort einlogen. Das heisst, es muss auch ein Passwort im LDAP-Verzeichnis eingetragen sein.

9 OpenCA

9.1 Einleitung

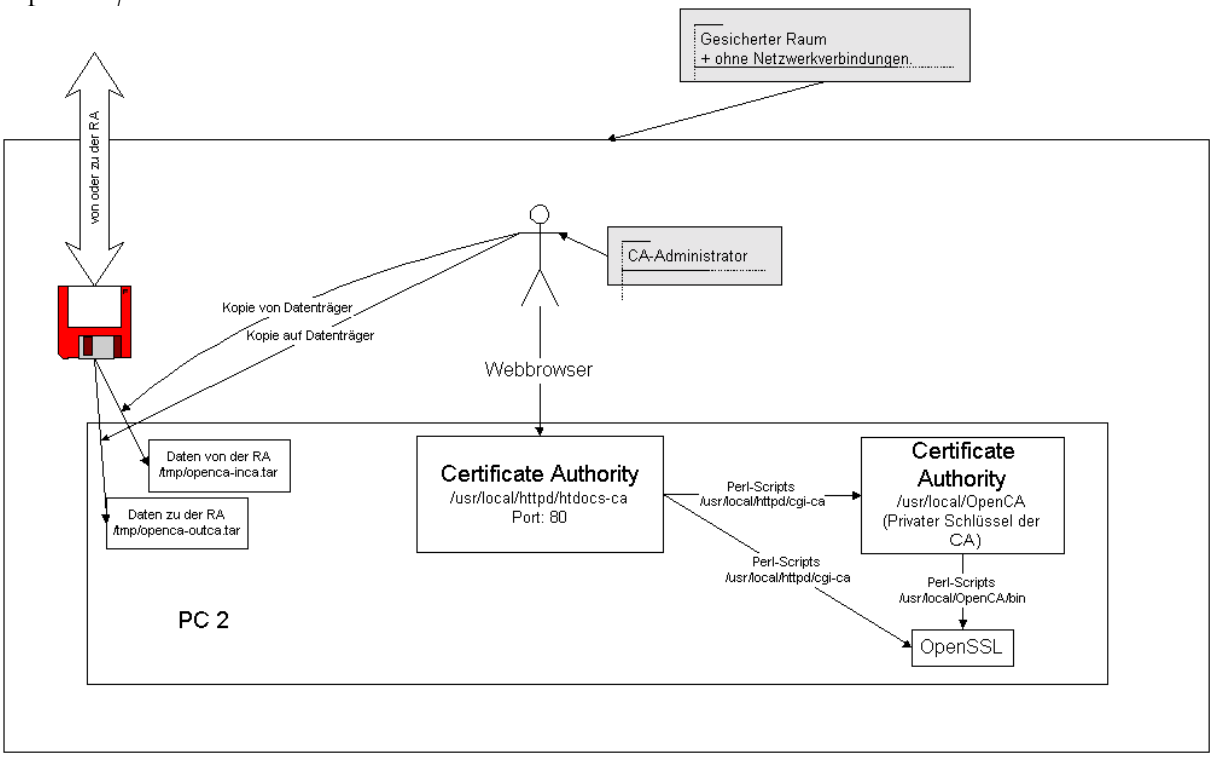
OpenCA ist ein Softwareprojekt für eine grafische Oberfläche, die das Verwalten von X.509 zur Verfügung stellt. Das Ziel des Projekts ist das Erstellen und Verwalten einer CA (Certificate Authority). Entwickelt wird es von freischaffenden Mitarbeitern, welche OpenCA unter einer Apache ähnlichen Lizenz (frei für den kommerziellen und nicht kommerziellen Gebrauch) vertreiben. OpenCA verwendet unter der Oberfläche diverse andere OpenSource-Produkte. Die Hauptarbeit im Hintergrund wird von OpenSSL geleistet, welches von OpenCA über Perl-Scripts gesteuert wird.
OpenCA besteht grundsätzlich aus drei Komponenten beziehungsweise Schnittstellen. Alle Schnittstellen bestehen aus Webseiten auf einem Webserver:

- „Registration Authority Server (RA)“: Über diese Webseiten kann ein CA-Administrator die Zertifikate verwalten.
- „Certification Authority Server (CA)”: Auf diesem Server liegt die CA und damit auch der „Secret Key“ der CA. Aus Sicherheitsgründen sollte darum dieser Server nicht an ein Netzwerk angeschlossen sein und auch physikalisch gesichert sein.
- „Secure Server”: Mit einem Netscape-Browser kann ein Benutzer über diese Webseiten einen Anfrage für ein neues Zertifikat erstellen. Ausserdem sind alle ausgestellten Zertifikate, das CA-Zertifikat und die CRL verfügbar.



Da diese drei Komponenten aus Webseiten und Perl-Scripts bestehen, ist es auch möglich zwei oder gerade alle Komponenten auf einem Server laufen zu lassen. Die CA sollte jedoch ausser bei Testzwecken immer auf einem eigenen PC in einem gesicherten Raum sein, in dem nur einzelne Personen Zugriff haben. Die RA und der „Secure Server“ können auf einzelnen Computern lau-

fen, es wird jedoch empfohlen diese auf dem gleichen Server laufen zu lassen, da es damit auch weniger Konfigurationsarbeit gibt.
Bei dem RA und der CA braucht es ausser den Webseiten noch OpenSSL welche die Schlüssel verwaltet.
Über die RA können die Zertifikate auf einem LDAP-Server gespeichert werden. Der LDAP-Server dient jedoch nur als zusätzlicher Speicherort der Zertifikate. Die Zertifikate, CRL und die Zertifikate-Request werden lokal gespeichert (normalerweise unter /usr/local/RAServer oder /usr/local/OpenCA). Zum Schluss lässt sich noch sagen, dass die Web-Schnittstellen zur Zeit alle speziell für den Netscape-Browser (Version 3.x oder 4.x) geschrieben sind. Das betrifft vor allem die Funktion für das Erstellen des Zertifikats, da dieses mit dem HTML-Tag <keygen> initiiert wird und der Internet Explorer dieses Tag (noch) nicht unterstützt. Der Administrator braucht für das Unterschreiben der Zertifikate und der Request ebenfalls den Netscape-Browser. Wie der Ablauf des Erstellens eines Zertifikats in der Praxis genau aussieht, ist im Kapitel „Bedienung OpenCA/Ablauf eines .....“ beschrieben.



9.2 Installation

Während der Praktikumsarbeit wurde die Version 0.2.0-3 von OpenCA vom 30.Dezember 1999 verwendet. Seit diesem Datum sind bis jetzt nur Snapshots, bestimmt für Entwickler erschienen. Die Dokumentation bezieht sich darum ausser im Kapital „Zukunft“ auf die Version 0.2.0-3. Ausserdem verwendeten wir OpenSSL 0.9.5 und als Webserver Apache Version 1.3.6 von der Suse Distribution. Für den LDAP-Server wurde Netscape Directory Server 4.11 verwendet.
Vor der Installation sollte sichergestellt sein, dass mindestens OpenSSL (ab Version 0.9.5) und ein Webserver mit Perl (5+ with DBM support) und SSL-Unterstützung, installiert ist.
Auf jedem Server müssen zuerst noch die benötigten Perl-Pakete installiert werden. Genaueres ist in Datei „Install“ von OpenCA im Abschnitt „Quick Install“ dokumentiert. Anschliessend kann es an die Installation von OpenCA gehen. Je nach Komponente wird eine der folgenden Kommandozeilen gewählt:
`make install-ca ; make install-ca-web` (siehe auch Kapitel 1)
`make install-raserver ; make install-raserver-web` (siehe auch Kapitel 6 in Install-Datei)
`make install-secure-web` (siehe auch Kapitel 7 in Install-Datei).
Das Meiste der Installation ist eigentlich in der „Install“-Datei gut beschrieben. Bei der CA kann bei der Installation auch gerade der CA-Schlüssel erstellt werden. Da der RA-Server und der Secure-Server über SSL laufen sollen, muss für diese wenn nötig noch ein Server-Zertifikat erstellt

werden. Dieser Vorgang ist in Kapitel 3 der Install-Datei beschieben. Damit der Administrator auf den RA-Server zugreifen kann, muss er ebenfalls noch ein Zertifikat haben, welches nach dem gleichen Vorgang wie für das Server-Zertifikat erstellt wird. Um dieses beim Administrator in den Browser einlesen zu können, kann nach Kapitel 4 der Install-Datei eine PKSC#12-Datei erstellt werden. Um auf die Webseiten der Schnittstellen zu zugreifen, müssen jedoch noch zuerst ein paar Konfigurationen erledigt werden. Diese sind in den nächsten zwei Kapiteln beschrieben.

9.2.1 Installation CA

Die Standardpfade stimmen im Normalfall. Als Webuser muss bei Suse „wwwrun.nogroup" angegeben werden. Die Frage, ob die Datei „/usr/local/OpenCA/stuff/openssl.cnf" editiert werden soll, kann mit nein geantwortet werden. Danach kann direkt ein CA Zertifikat erstellt werden.
Nach der Installation sollte noch die Konfigurationsdatei von OpenSSL angepasst werden.
Problem: Das CA Certificate ist standardmässig nur einen Monat lang gültig.

9.2.2 Installation CA-Web

Die Pfade der Webseiten sollten hier an die lokalen Gegebenheiten angepasst werden. Zum Beispiel bei Suse: „/usr/local/httpd/htdocs-ca" und „/usr/local/httpd/cgi-ca"

9.2.3 Installation RA-Server

Die Standardpfade stimmen im Normalfall. Als Webuser muss bei Suse „wwwrun.nogroup" angegeben werden.

9.2.4 Installation RA-Server-Web

Die Pfade der Webseiten sollten hier an die lokalen Gegebenheiten angepasst werden. Zum Beispiel bei Suse: „/usr/local/httpd/htdocs-raserver" und „/usr/local/httpd/cgi-raserver"
Für die Client-Authentifikation muss der Private-Key und der öffentliche Schlüssel des Servers in der Konfigurationsdatei angegeben werden.

9.3 Konfiguration Apache-Webserver

Im diesem Kapitel werden die nötigen Konfigurationsarbeiten für den Apache-Webserver beschrieben. Allgemein gehalten müssen folgende Tätigkeiten durchgeführt werden:

- Webserver für SSL konfigurieren (betrifft nur RA und Secure Server),
- Server-Zertifikat installieren (betrifft nur RA und Secure Server),
- Webserver für Client-Authentifikation einstellen. (betrifft nur RA Server),
- Konfiguration der Virtual Hosts mit den Pfadangaben zu den Webseiten und den Perl-Scripts.

OpenCA bietet bereits Vorgabe-Konfigurationsdateien für den Apache-Server, die auf die Distribution von Red Hat abgestimmt sind. Diese Dateien befinden sich im Verzeichnis von OpenCA unter „misc/apache". Erklärungen dafür stehen in der Datei „README" im gleichen Verzeichnis.
In den vorgegebenen Konfigurationsdateien müssen nur noch die Pfade zu allen Dateien an die lokalen Gegebenheiten angepasst werden. Ausserdem müssen natürlich noch die Hostnamen und IP-Adressen angepasst und die Server-Zertifikate für den RA und Secure Server in einem Verzeichnis bereitgestellt werden. Die Konfigurationen für SSL sind bereits in den Dateien enthalten. Für die Client-Authentifikation kann eine Datei mit allen öffentlichen Schlüsseln in BASE64-Codierung der Administratoren, welche auf die RA zugreifen dürfen, angelegt werden. Bei unserer Suse-Distribution mit Apache 1.3.6 mussten wir zudem noch den Konfigurationsdateien die ladbaren Module hinzufügen (Dynamic Shared Object Support), da er sonst nicht alle Befehle in der Konfigurationsdatei kannte. Wer noch keine grosse Erfahrung mit Apache-Konfiguration hat, vergleicht am besten die neue Konfiguration mit der alten (bei Suse: „/etc/httpd/httpd.conf").
Weiter auf die Konfiguration hier einzugehen, hätte keinen Sinn, da die Einstellungen je nach System verschieden sind und die Dokumentation oder ein Buch über Apache sicher besser und schneller helfen kann.

9.4 Konfiguration Perl-Scripts

Die Perl-Scripts welche in den Verzeichnissen „cgi-ca“, „cgi-raserver“ oder „cgi-secure“ (neu: „cgi-public“) liegen, müssen auch noch angepasst werden. Die Einstellungen in den Dateien, welche die Endung „.cnf“ haben, sind wieder hauptsächlich Änderungen der Pfadangaben an die lokalen Gegebenheiten. Für die HTML-Formulare können zum Teil auch noch Default-Werte eingestellt werden. Die Einstellungen sollten jedoch alle selbsterklärend sein, und keine Probleme verursachen. In dem Verzeichnis „cgi-raserver“ befindet sich eine Datei namens „certsMail.txt“, welches an Personen verschickt wird, wenn deren Zertifikat über den Secure-Server abholbereit ist.

9.5 Bedienung OpenCA

9.5.1 Secure Server

Nach dem Öffnen der Startseite mit „https://hostname:443/“ (je nach Konfiguration geht auch „https://hostname/“), erscheint eine Liste mit den für den normalen Benutzer verfügbaren Funktionen: („Get CA Certificate“, „Certificate Revokation Lists“, „Request a Certificate“, „Get Requested Certificate“ und „Issued Certificates List“).
Die Funktion „Request a Certificate“ ist der Startpunkt, wenn ein Besucher ein Zertifikat erstellen will. Bedingung dafür ist ein Netscape-Browser der Version 3.x oder 4.x. Genaueres ist im Kapitel „Ablauf beim Erstellen eines Zertifikats“ beschrieben.

9.5.2 Registrations Authority Server

Nach dem Öffnen der Startseite mit „https://hostname:4443/“, auf welche nur Administrator Zugriff haben, stehen folgende Funktionen zur Verfügung: „Export Requests“, „Pending Requests“, „Approved Requests“, „Remove Exported Requests“, „Import CA Certificate“, „Import New Certificates“, „Export Certs onto LDAP“, „Import CRL“, „Export Certificate Revokation Requests“, „Send e-mails to users for newly issued certs“, „Delete Temp Files (after importing certs)“
Bevor die Zertifikat-Anfragen zu der CA exportiert werden können, müssen diese von einem Administrator unter der Funktion „Pending Requests“ unterschrieben werden.

9.5.3 Certificate Authority Server

Nach dem Öffnen der Webseite mit „http://hostname/“ stehen folgende Funktion zur Verfügung: „CA Management (Initialization)“, „Import Request“, „Pending Request“, „Deleted Request“, „Remove Deleted Request“, „Issued Certificates“, „Export Certificates“ und „Export CRL“. Die erste Funktion ist in unserem Fall nicht nötig, da bereits bei der Installation der CA ein Stammzertifikat erstellt wurde. Die anderen Funktion sollten selbsterklärend sein. Bei unseres Version wurden die Funktion „Remove Deleted Request“ jedoch noch nicht unterstützt. Die wichtigste der Funktionen ist „Pending Request“, welche eine Liste aller anstehenden Zertifikat-Anfragen auflistet und dem Administrator erlaubt, jedes einzelne der Zertifikate zu unterschreiben.

9.5.4 Ablauf beim Erstellen eines Zertifikats

Anschliessend wird der Ablauf beschrieben, welcher von dem Zertifikate-Request eines Benutzers startet und mit dem Erhalten des Zertifikats endet. Vor jedem Ablaufschritt steht, wer für diesen Schritt verantwortlich ist.

1. (Benutzer) Der Benutzer füllt auf dem Secure-Server das Formular unter „Request a Certificate“ aus und schickt dieses ab. Nach dem Bestätigen der nächsten Seite erscheint ein Dialog vom Netscape-Browser für die Schlüsselerstellung. Nach dem Bestätigen des Dialogs ist der Schlüssel generiert und wird in der lokalen Netscape-Datenbank gespeichert. Der Request ist nun bei RA gespeichert.
2. (Benutzer) Der Benutzer muss nun bei einer Registrierungsstelle vorbei, um seine Identität zu bestätigen. Dabei muss er den Pincode dem Administrator mitteilen, um sicher zu gehen, dass er derjenige ist, welcher den Request aufgefüllt hat.
3. (RA-Administrator) Der Administrator, welcher die ID des Benutzers bestätigt hat, muss nun über den RA-Server mit der Funktion „Pending Request” den Zertifikat-Request

unterschreiben. Unter der Funktion „Pending Request“ werden alle anstehenden Requests aufgelistet. Bei der aktuellen Version muss beim Unterschreiben zweimal die Schaltfläche zum unterschreiben gedrückt werden.

4. (CA-Administrator) Dieser muss nun für die anstehenden unterschriebenen Requests die Zertifikate erstellen. Zuerst müssen auf dem RA-Server mit „Export Requests“ die unterschriebenen Requests exportiert werden. Normalerweise müsste der Administrator nun die Exportdatei (standardmässig „/tmp/openca-inca.tar“ oder „/tmp/openca-outca.tar“) auf einem Datenträger kopieren und bei der CA wieder einlesen. Bei unserem Testserver befinden sich alle Server auf dem gleichen Computer, womit das nicht nötig ist.
5. (CA-Administrator) Bei der CA müssen also nun die neuen Requests mit der Funktion „Import Request“ eingelesen werden. Unter „Pending Request“ werden die anstehenden Reqest aufgelistet und müssen einzeln unterschrieben werden. Danach sind die Zertifikate unter „Issued Certificates“ aufgelistet und können mit der Funktion „Export Certificates“ in die Exportdatei geschrieben werden. Diese muss wieder per Diskette auf den RA-Server gebracht werden und dort mit „Import New Certificates“ eingelesen werden.
6. (CA-Administrator) Nach dem Einlesen sind die Zertifikate unter „Issued Zertificates“ beim „Secure-Server“ sichtbar. Beim RA-Server kann der Administrator nun mit „Send e-mails to users...“ an alle Benutzer der neuen Zertifikate eine E-Mail schicken, dass ihr Zertifikat abholbereit ist. Zusätzlich kann der Administrator alle bestehenden Zertifikate auf den LDAP-Server schreiben. Die Zertifikate werden dabei unter dem Eintrag mit der gleichen E-Mail-Adresse, wie im Zertifikat abgelegt.
7. (Benutzer) Der Benutzer kann über die Funktion „Get Requested Certificate“ in einem Formular seine Serial-Number angeben, welche in der E-Mail steht. Nach dem Abschicken des Formular wird sein Zertifikat in den Browser eingelesen und ist ab sofort mit dem privaten Schlüssel für sicheren E-Mail-Verkehr benutzbar.

9.6 Zukunft

Zur Zeit unterstützt OpenCA leider noch nicht alle Funktionen, die für ein Management einer grossen Anzahl von Zertifikaten nötig wären, wie zum Beispiel die Unterstützung zum Revoken von Zertifikaten. Ebenfalls werden ist die Verbindung zum LDAP-Server nicht gross gebraucht. In der von uns benutzten Version, wurden neue Zertifikate auf dem LDAP-Server gespeichert, jedoch nicht verwaltet. Die Übersichtlichkeit von OpenCA könnte ebenfalls noch gesteigert werden. Da OpenCA auch laufend verbessert wird, ist es auch kaum für den produktiven Betrieb geeignet, jedoch ist nach meiner Meinung OpenCA auf einem guten Weg, als stabile CA produktiv einsetzbar zu werden. Das wird jedoch wahrscheinlich erst in ½ bis zwei Jahren der Fall sein. OpenCA kann zwar bereits zum Erstellen von Zertifikaten gebraucht werden, jedoch gibt es immer noch kleinere Fehler und Funktionen die noch nicht laufen. Vor allem ist die Dokumentation zu OpenCA noch recht klein, doch wenn OpenCA laufend verbessert und geändert wird, hat das auch noch keinen grossen Sinn.

Änderungen in der aktuellen Snapshot 4.5.2000.

- Installation wurde vereinfacht. Mögliche Installationswege werden bei Eingabe von „make“ angezeigt. Dokumentation zu der Installation stimmt jedoch nicht mehr.
- Installation des RAServer-Web hat noch Fehler. Wodurch die Installation ohne Fehlerkorrektur nicht möglich ist.
- Die Oberfläche wurde verbessert und einzelne Funktionen hinzugefügt.

9.7 Offene Fragen

Bei der Zugriffs-Beschränkung auf den RA-Server ist für uns noch eine Frage offen, welche aus Zeitgründen nicht mehr geklärt werden konnte. Die Webseiten der RA wurden so konfiguriert, dass alle, welche ein Zertifikat der eigenen CA haben, Zugriff auf den RA-Server haben, was wohl nicht so sein sollte. Wie diese Zugriffsbeschränkung richtig gemacht wird, ist also nicht klar, sollte jedoch lösbar sein.

10 LDAP-Server mit SSL

10.1 SSL

10.1.1 Einleitung

Der Netscape Directory Server unterstützt standardmässig den Aufbau einer sicheren Verbindung zu einem Client mit SSLv2 und SSLv3. Damit diese gebraucht werden kann, müssen jedoch zuerst ein paar Konfigurationen, wie die Installation des Server-Zertifikat gemacht werden.

10.1.2 Installation

Mit Hilfe von OpenCA (siehe Zertifikat manuell generieren) kann sehr leicht ein Zertifikat für den Server (der Name im Zertifikat sollte dabei natürlich dem DNS-Namen des Servers entsprechen) generiert werden. Das Zertifikat muss ausserdem ein Server-Zertifikat sein.
Wenn das Stamm-Zertifikat, das Zertifikat des Servers und der Schlüssel des Servers als PEM-Dateien bereitliegen kann gestartet werden.

1. Netscape Console starten und sich anmelden.
2. Management-Oberfläche des Directory-Servers starten und in Register „Tasks“ wechseln. Nun stehen im Menü „Console“ die Funktionen „Certificate Setup Wizard“ und „Manage Certificate“ zur Verfügung, die anschliessend gebraucht werden.
3. Installation des Stamm-Zertifikats
   „Certificate Setup Wizard“ starten und die zweite Frage bei dem Dialog Token Selection mit „Yes“ beantworten. Zwei Dialoge weiter wird der Punkt „Trusted Certificate Authority“ ausgewählt. Im nächsten Dialog muss entweder der Dateiname der Stammzertifikats (im PEM-Format) angegeben werden, oder das Zertifikat in das Textfeld eingefügt werden. Ein Dialog weiter wird das Zertifikat angezeigt womit eine Schlusskontrolle möglich ist. Wenn alles in Ordnung ist, kann das Stammzertifikat mit der Schaltfläche „Add“ hinzugefügt werden.
4. Kontrolle Stamm-Zertifikat
   Über das Menü „Console/Manage Certificate“ sollte nun ein Dialog erscheinen, in welchem das Stammzertifikat mit der Bemerkung „Trusted Client CA“ vorhanden ist. Im diesen Dialog kann das Zertifikat auch überprüft oder gelöscht werden.
5. Installation des Server-Zertifikat
   Wieder den „Certificate Setup Wizard“ starten und die zweite Frage beim Dialog mit „Yes“ beantworten. Zwei Dialoge weiter wird der Punkt „This Server“ ausgewählt und das Passwort, mit dem das Server-Zertifikat verschlüsselt wurde, angegeben. Wie vorhin kann nun wieder der Dateiname angegeben oder direkt der Dateiinhalt ins Textfeld eingefügt werden. Ein Dialog weiter wird wieder das Zertifikat angezeigt und kann das Server-Zertifikat mit der Schaltfläche „Add“ hinzufügt werden.
6. Anschliessend kann das Server-Zertifikat wieder unter „Console/Manage Certificate“ überprüft werden, wo dieses als „Own“ vermerkt sein sollte.
7. Jetzt muss SSL noch aktiviert werden. Das lässt sich im Register „Configuration/Encryption“ erledigen. Mindestens „Enable SSL“ und „RSA“ sollten aktiviert und als Zertifikat das eigene Zertifikate ausgewählt sein.

10.2 Client-Authentifikation

Mit den erstellten Zertifikaten ist es nun auch möglich, zwischen einem Server und dem Client nicht nur eine SSL Verbindung aufzubauen, sondern dabei auch noch den Client zu authentifizieren.
Beim Verbindungsaufbau schickt der Client dabei unter anderem auch sein eigenes Zertifikat zum Server. Dieser überprüft nun, ob er dem Zertifikat beziehungsweise der CA, welche dieses Zertifikat ausgestellt hat vertraut. Ist das der Fall, wird die Verbindung hergestellt, andernfalls abgebrochen. Bei einem LDAP-Server können nur Benutzer zugreifen, welche auch einen Eintrag im Verzeichnis haben. Mit dem DN dieses Eintrages ist der Client dann auch angemeldet. Um diesen Eintrag herauszufinden wird eine Suche im Verzeichnis gestartet, die genau einen Eintrag zurück-

liefern muss (die Person, welche das Zertifikat gehört), andernfalls wird die Verbindung abgebrochen. Wie diese Suche genau aussieht, kann eingestellt werden, normalerweise wird für die Suche eines oder mehrere Attribute des Zertifikats genommen. Zusätzlich besteht die Möglichkeit einer Zertifikatsüberprüfung, das heisst, es wird überprüft, ob das vom Client beim Verbindungsaufbau übergebene Zertifikat auch in dem Eintrag unter dem Attribut `userCertificate` gefunden werden kann. Ein Verbindungsabbruch ist die Folge, wenn das Zertifikat nicht gefunden wird.

10.2.1 Client-Authentifikation mit Netscape Directory Server

Um eine SSL-Verbindung mit Client-Authentication aufzubauen muss natürlich zuerst der LDAP-Server für eine normale SSL-Verbindung eingerichtet sein. Siehe dazu letztes Kapitel. Um Client-Authentication zu erlauben, muss der Directory-Server noch unter dem Register „Configuration/Encryption“ bei der Gruppe „Client Authentication“ auf „Allow client authentication“ konfiguriert werden. Nach einem Neustart des LDAP-Servers können bereits SSL-Verbindungen mit „Client-Authentication“ hergestellt werden.
Für eine korrekt Funktion muss jedoch noch zuerst die Suchabfrage konfiguriert werden, welche die Person im Verzeichnis zurückliefern sollte. Diese Einstellungen erfolgen in der Datei „shared/config/certmap.conf“, welche im Installationsverzeichnis des Netscape Directory Server liegt. Eine genaue Anleitung zu der Konfiguration dieser Datei ist in der Dokumentation des Servers „Managing Servers with Netscape Console“ im Kapitel „Using SSL/Using Client Certificates“ zu finden.
Bei unseren Tests sind jedoch kleinere Schwierigkeiten zum Vorschein gekommen. Da keines der Attribute in unseren Zertifikaten mit einem der `o/ou/c` – Attribute im LDAP-Server übereinstimmte wäre nach Dokumentation die Konfiguration der Einstellung „DNComps“ mit keinem Wert die einzige Möglichkeit. „DNComps“ bezeichnet normalerweise den DN des Startortes der Suche im LDAP-Verzeichnis. Bei einem leeren Wert sollte eigentlich der ganze LDAP-Baum durchsucht werden, was jedoch bei uns nicht der Fall war. Was der Fehler war konnte nicht festgestellt werden, klar feststellbar war nur, dass die Suchabfrage kein Resultat lieferte.
Um dieses Probleme schon von vornherein zu umgehen, sollte darauf geachtet werden, dass mindestens eines der Attribute o/ou/c im LDAP-Server und in den Zertifikaten das gleiche ist. Wenn das nicht der Fall ist, müsste für das Mapping zwischen Zertifikat und LDAP-Eintrag eine eigene Funktion geschrieben werden, die den Standard Mapping-Mechanismus von Netscape ersetzt. Weitere Informationen sind dazu in der Netscape-Dokumentation zu der „Netscape Console“ zu finden. Eine denkbare Lösung des Problem wäre eine Referenz im LDAP-Verzeichnis auf die richtigen Einträge. Die Referenz hätte dann den Namen, welcher in den Zertifikaten steht. Unsere Tests zeigten jedoch, dass die Standard-Mapping-Funktion diese Referenz nicht verfolgt und somit das Ganze keine Lösung ist.
In der Konfigurationsdatei kann ebenfalls das Überprüfen des Zertifikats ein und ausgeschaltet werden (standardmässig aus). Bei eingeschalteter Überprüfung, wird das Zertifikat, welches der Benutzer beim Verbindungsaufbau mitgibt, mit dem Zertifikat des Benutzereintrages auf dem LDAP-Verzeichnis verglichen. Wenn das Attribut `userCertificate` mehrere Zertifikate beinhaltet, werden alle mit dem Benutzerzertifikat verglichen. Diese Überprüfung hat den Vorteil das wenn ein Benutzer nicht mehr akzeptiert werden soll, oder das Zertifikat ungültig ist, mit dem Löschen des Zertifikat beim Benutzereintrag im LDAP-Server, der Benutzer keine Verbindung mehr aufbauen kann (natürlich nur über Client-Authentication).

10.2.2 Unterstützung Client-Autentication der Clients

Netscape Communicator 4.x oder der Microsoft Internet Explorer 5.01 unterstützen keine Client-Autentication zu einem LDAP-Server. Zu einem Web-Server hingegen schon.
Die einzige gefundene Unterstützung der Client-Authentifikation stellten die Tools des Netscape LDAP SDK bereit.

Zum Beispiel eine Abfrage mit ldapsearch:

```
ldapsearch –p 636 –P netscape/cert7.db –Z –W Passwort –N Keyname –h local-
host –K netscape/key3.db –b c=CH mail=i7griede@zhwin.ch
```

Das Zertifikate muss in einem Netscape-spezifischen Format vorliegen, welches auch der Netscape Communicator verwendet. Um diese Datei zu erhalten ist es darum am einfachsten, das private Zertifikat im Communicator über das Security-Menü einzulesen und danach diese Dateien aus dem eigenen Netscape-Konfigurationsverzeichnis zu kopieren. „Keyname“ ist die Bezeichnung des Zertifikats im Communicator und „Passwort“ natürlich das Passwort zu dem privaten Schlüssel.

11 Ausblick

In diesem Kapitel geht es darum aufzuzeigen, was man noch unternehmen muss, um unsere Projektarbeit umzusetzen.
Selbstverständlich muss ein definitives Schema, mit entsprechenden Rechten, aufgestellt und implementiert werden.
Ausserdem muss man sich überlegen, ob OpenCA in der Form wie es heute existiert genügt. An OpenCA wird noch laufend gearbeitet, es ist also möglich, dass neuere Versionen recht komfortabel sind. Bis jetzt ist OpenCA leider noch nicht sehr benutzerfreundlich und nicht kompatibel mit dem Internet Explorer. Wünschenswert wäre, wenn zukünftige Versionen mehr Gebrauch vom LDAP-Server machen würden. Es wäre natürlich möglich selber an OpenCA zu arbeiten und es den Bedürfnissen anzupassen. Eine weitere Möglichkeit ist das Netscape Certification Management System, oder andere kommerzielle Produkte.
Was zur Zeit auch noch fehlt, ist ein Werkzeug zum Überprüfen der gesetzten Zugriffsrechte. Es ist zwar theoretisch möglich, die Rechte manuell mit dem Gateway zu überprüfen, doch ist dies natürlich für eine grössere Datenbank nicht akzeptabel.
Ziel muss es sicher sein, dass es für den Benutzer einfach ist, ein eigenes Zertifikat zu erhalten. Aber es ist mindestens ebenso wichtig, nach einem Zertifikat einer anderen Person suchen zu können. Und wenn wir schon einen LDAP-Server haben, wäre es schade, wenn dieser nur für das Verteilen von X.509 Zertifikaten benützt würde. Man könnte sich also auch einen Link auf der Homepage der ZHW vorstellen, der auf eine Seite verweist, wo man Adressen und andere Daten einer Person der Schule suchen könnte. Hier wäre dann allerdings das genaue Definieren der Rechte eine wichtige Sache.

12 Schlusswort

Die Arbeit hat uns von Anfang an angesprochen. Uns hat besonders gefallen, dass es darum ging, etwas aufzustellen, was später auch tatsächlich verwendet werden könnte. Ausserdem sind Zertifikate und sichere Kommunikation ein sehr aktuelles Thema. Diese Projektarbeit gab uns die Gelegenheit, uns zu diesem wichtigen Thema vertieftes Wissen anzueignen.

Unsere Projektarbeit empfanden wir als angenehm. Ein Grund dafür ist, dass wir nicht zuerst wochenlang Unterlagen studieren mussten, um einen Weg zum Ziel sehen zu können. Schon sehr bald war es möglich, eine LDAP-Abfrage auf unseren eigenen LDAP-Server vorzunehmen, was natürlich motivierend war. Ebenfalls motivierend für uns war, dass wir nie an ein wirklich grosses Problem stiessen, an welchem wir Wochen lang hätten arbeiten müssen.

Wir glauben daran, dass wir mit unserer Projektarbeit eine Möglichkeit aufzeigen, wie man an der ZHW in Zukunft sichere und vertrauenswürdige E-Mail-Kommunikation ermöglichen könnte. Wir hoffen natürlich, dass dies auch geschehen wird.

Was wir sicher sagen können ist, dass diese Projektarbeit unser Wissen erweitert und unsere Erfahrung vergrössert hat. Wir haben anhand praktischer Arbeit viel gelernt.

Markus Grieder Stephan Zehnder

13 Quellen

Das OpenSSL Handbuch
„http://www.pca.dfn.de/dfnpca/certify/ssl/handbuch/ossl095/ossl095.html“

Netscape Directory Server 4.11 Documentation
„http://www.home.netscape.com/eng/server/directory/4.11/“

IBM Redbook Understanding LDAP
„http://www.redbooks.ibm.com/abstracts/sg244986.html“

Implementing LDAP, Mark Wilcox, Wrox press, ISBN 1-861002-21-1

IETF RFC 1777 Lightweight Directory Access Protocol (LDAP)

IETF RFC 1778 The String Representation of Standard Attribute Syntaxes

IETF RFC 1779 A String Representation of Distinguished Names

IETF RFC 1781 Using the OSI Directory to Achieve User Friendly Naming

IETF RFC 1804 Schema Publishing in X.500 Directory

IETF RFC 1823 The LDAP Application Program Interface

IETF RFC 1959 An LDAP URL Format

IETF RFC 1960 A String Representation of LDAP Search Filters

IETF RFC 2251 Lightweight Directory Access Protocol (v3)

IETF RFC 2252 Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions

IETF RFC 2253 Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names

IETF RFC 2254 The String Representation of LDAP Search Filters

IETF RFC 2255 The LDAP URL Format

IETF RFC 2256 A Summary of the X.500(96) User Schema for use with LDAPv3

IETF RFC 2559 Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2

Beispiele von LDAP URLs
„http://fbi.zhwin.ch/KSy/Block04/ldap_url.htm“

OpenSSL Home Page
„http://www.openssl.org„

C't 8/99, Heise Verlag

14 Anhang

14.1 Listing der Datei openssl.cnf

```
#
# OpenSSL example configuration file.
# This is mostly being used for generation of certificat requests.
#

# This definition stops the following lines choking if HOME isn't
# defined.
HOME            = .
RANDFILE        = /opt/ldap/ssl/.rand   # File für die Initialisierung des
                                        # Zufallsgenerators, gibt es #die-
                                        ses File noch nicht, kann #irgend
                                        ein grösseres File in den #ent-
                                        sprechenden Pfad kopiert und #um-
                                        benannt werden.

# Extra OBJECT IDENTIFIER info:
#oid_file        = $ENV::HOME/.oid
oid_section     = new_oids

# To use this configuration file with the „-extfile“ option of the
# „openssl x509“ utility, name here the section containing the
# X.509v3 extensions to use:
# extensions            =
# (Alternatively, use a configuration file that has only
# X.509v3 extensions in its main [= default] section.)

dir             = /opt/ldap/ssl         #Definition des pfades

[ new_oids ]

# We can add new OIDs in here for use by 'ca' and 'req'.
# Add a simple OID like this:
# testoid1=1.2.3.4
# Or use config file substitution like this:
# testoid2=${testoid1}.5.6

####################################################################
[ ca ]
default_ca  = User_CA         # The default ca section
```

```
##################################################################################
[ Root_CA ]

certs       = $dir/certs            # Where the issued certs are kept
crl_dir     = $dir/crl              # Where the issued crl are kept
database    = $dir/index.txt        # database index file.
new_certs_dir     = $dir/newcerts   # default place for new certs.

certificate = $dir/PCAcert.pem      # The CA certificate
serial      = $dir/serial           # The current serial number
crl         = $dir/crl.pem          # The current CRL
private_key = $dir/private/PCAkey.pem     # The private key
RANDFILE    = $dir/private/.rand    # private random number file

x509_extensions   = PCA_ext         # The extentions to add to the cert
                                    # weiter unten definiert

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crl_extensions  = crl_ext

default_days      = 730       # how long to certify for
default_crl_days= 30          # how long before next CRL
default_md  = md5             # which md to use.
preserve    = no              # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy            = policy_match

##################################################################################
[ User_CA ]                                 # Abschnitt fuer eine User CA

certs           = $dir/certs               # Where the issued certs are kept
crl_dir         = $dir/crl                 # Where the issued crl are kept
database        = $dir/index.txt           # database index file.
new_certs_dir   = $dir/newcerts            # default place for new certs.

certificate     = $dir/UCAcert.pem         # The CA certificate
serial          = $dir/serial              # The current serial number
crl             = $dir/crl.pem             # The current CRL
private_key     = $dir/private/UCAkey.pem  # The private key
RANDFILE        = $dir/private/.rand       # private random number file

x509_extensions = UCA_ext           # The extentions to add to the cert
                                          # weiter unten definiert


#crl_extensions = crl_ext                 # Extensions to add to CRL
default_days    = 365                     # how long to certify for
default_crl_days= 30                      # how long before next CRL
default_md      = md5                     # which md to use.
preserve        = no                      # keep passed DN ordering

policy          = policy_anything
```

```
#####################################################################

# For the CA policy

[ policy_match ]

countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.

[ policy_anything ]

countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

#####################################################################

[ req ]

default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions         = v3_ca     # The extentions to add to the self signed
cert

# Passwords for private keys if not present they will be prompted for
# input_password = secret
# output_password = secret

# This sets a mask for permitted string types. There are several options.
# default: PrintableString, T61String, BMPString.
# pkix       : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or UTF8Strings
# so use this option with caution!
string_mask = nombstr

# req_extensions = v3_req # The extensions to add to a certificate request
```

```
################################################################################

[ req_distinguished_name ]
countryName             = Country Name (2 letter code)
countryName_default             = CH
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Zuerich

localityName                    = Locality Name (eg, city)
localityName_default            = Winterthur

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = Zuercher Hochschule Winterthur

# we can do this but it is not needed normally :-)
#1.organizationName             = Second Organization Name (eg, company)
#1.organizationName_default     = World Wide Web Pty Ltd

organizationalUnitName          = Organizational Unit Name (eg, section)
organizationalUnitName_default      = Informationstechnologie

commonName              = Common Name (eg, YOUR name)
commonName_max                  = 64

emailAddress                    = Email Address
emailAddress_max        = 60

# SET-ex3               = SET extension number 3

################################################################################

[ req_attributes ]

# Das Challenge Password dient dazu, sich bei Verlust des geheimen
# Schluessels
# gegenueber der Herausgeber-CA fuer einen Zertifikatwiderruf auszuweisen.
# Wird bei Erstellung der Zertifikat-Anforderung erfragt.

challengePassword       = A challenge password
challengePassword_min           = 4
challengePassword_max           = 20

unstructuredName        = An optional company name
```

```
###############################################################################

[ PCA_ext ]                          # Extensions fuer ein Root Zertifikat

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.
basicConstraints         = critical, CA:TRUE

# Moeglich: digitalSignature, nonRepudiation, keyEncipherment,
#           dataEncipherment, keyAgreement, keyCertSign,
#           cRLSign, encipherOnly, decipherOnly
keyUsage                 = cRLSign, keyCertSign

# PKIX recommendations
subjectKeyIdentifier     = hash
authorityKeyIdentifier   = keyid,issuer:always

# Import the email address.
subjectAltName           = email:copy

# Copy subject details
issuerAltName            = issuer:copy

#crlDistributionPoints    = URI:http://mystic.pca.dfn.de/PCA.crl

# Moeglich: client, server, email, objsign, reserved, sslCA, emailCA, objCA
nsCertType               = sslCA, emailCA, objCA

###############################################################################

[ UCA_ext ]                          #Extensions fuer ein User Zertifikat

# basicConstraints       = critical, CA:FALSE
keyUsage                 = digitalSignature, keyEncipherment, keyAgreement
subjectKeyIdentifier     = hash
authorityKeyIdentifier   = keyid,issuer:always
subjectAltName           = email:copy
issuerAltName            = issuer:copy
#crlDistributionPoints    = URI:http://mystic.pca.dfn.de/UCA.crl
nsCertType               = client, email
#nsBaseUrl                = https://mystic.pca.dfn.de/
#nsCaPolicyUrl            =
http://www.pca.dfn.de/dfnpca/policy/wwwpolicy.html
#nsComment                = This certificate was issued by a User CA
#nsRevocationUrl          = cgi/non-CA-rev.cgi?
# nsRenewalUrl           = cgi/check-renw.cgi?

###############################################################################
```

```
[ v3_ca ]

basicConstraints         = critical, CA:TRUE
subjectKeyIdentifier     = hash
authorityKeyIdentifier   = keyid:always,issuer:always
keyUsage                 = cRLSign, keyCertSign
nsCertType               = sslCA, emailCA, objCA
subjectAltName           = email:copy
issuerAltName            = issuer:copy
#crlDistributionPoints    = URI:http://mystic.pca.dfn.de/PCA.crl
#nsBaseUrl                = https://mystic.pca.dfn.de/
#nsCaPolicyUrl            =
http://www.pca.dfn.de/dfnpca/policy/wwwpolicy.html
#nsComment                = This certificate is a Root CA Certificate


# RAW DER hex encoding of an extension: beware experts only!
# 1.2.3.5=RAW:02:03
# You can even override a supported extension:
# basicConstraints       = critical, RAW:30:03:01:01:FF

################################################################################

[ crl_ext ]

# CRL extensions.
# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.

issuerAltName            = issuer:copy
authorityKeyIdentifier   = keyid:always,issuer:always
```

14.2 Beispiel einer LDIF Datei

```
dn: uid=mgrieder, ou=People, o=zhwin.ch
cn: Grieder Markus
givenname: Markus
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
ou: Informationstechnologie
ou: People
ou: Studenten
ou: Informationstechnologie
ou: It97a
uid: mgrieder
sn: Grieder
userCertificate;binary:: MIIE8DCCA9igAwIBAgIBAjANBgkqhkiG9w0BAQQFADCBwTELM
 AkGA1UEBhMCQ0gxEDAOBgNVBAgTB1p1ZXJpY2gxEzARBgNVBAcTCldpbnRlcnRodXIxJzAlB
 gNVBAoTHlp1ZXJjaGVyIEhvY2hzY2h1bGUgV2ludGVydGh1cjEgMB4GA1UECxMXSW5mb3Jt
 YXRpb25zdGVjaG5vbG9naWUxFDASBgNVBAMTC1pIVyBSb290IENBMSowKAYJKoZIhvcNAQkBFh
 tzdGVwaGFuLnplaG5kZXJAYmlnZm9vdC5jb20wHhcNMDAwMzIxMTUwMTAxWhcNMDEwMzIxMTUw
 MTAxWjCBujELMAkGA1UEBhMCQ0gxEDAOBgNVBAgTB1p1ZXJpY2gxEzARBgNVBAcTCldpbnRlcn
 RodXIxJzAlBgNVBAoTHlp1ZXJjaGVyIEhvY2hzY2h1bGUgV2ludGVydGh1cjEgMB4GA1UECxMX
 SW5mb3JtYXRpb25zdGVjaG5vbG9naWUxFzAVBgNVBAMTDkdyaWVkZXIgTWFya3VzMSAwHgYJKo
 ZIhvcNAQkBFhFpN2dyaWVkZUB6aHdpbi5jaDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCg
 YEArMr47GrtYFfQTCbpXbZ1QAX1scnMlvDmIPDX1hB0UwqYHlkIeHiA2mnZj0zwW6Urz5lGkk
 JaXYZPfoSNqgmUVTTEhojhokp1+U/3hEN8SZAQT/5fF+xdy0juZTueMiA9hPSJkcdSaSEbH
 n3NrBzvTVwkrMKD1uYH8VI+TzWqR3MCAwEAAaOCAXowggF2MAsGA1UdDwQEAwIDqDAdBgNVHQ
 4EFgQUAYj8z3HAMvsjj8z7awLiFW8v2oQwge4GA1UdIwSB5jCB44AUJrYg3NF+W5Wn/zodWN3
 X32KGRTihgcekgcQwgcExCzAJBgNVBAYTAkNIMRAwDgYDVQQIEwdadWVyaWNoMRMwEQ
 YDVQQHEwpXaW50ZXJ0aHVyMScwJQYDVQQKEx5adWVyY2hlciBIb2Noc2NodWxlIFdpbnRl
 cnRodXIxIDAeBgNVBAsTF0luZm9ybWF0aW9uc3RlY2hub2xvZ2llMRQwEgYDVQQDEwtaSFcgU
 m9vdCBDQTEqMCgGCSqGSIb3DQEJARYbc3RlcGhhbi56ZWhuZGVyQGJpZ2Zvb3QuY29tggE
 AMBwGA1UdEQQVMBOBEWk3Z3JpZWRlQHpod2luLmNoMCYGA1UdEgQfMB2BG3N0ZXBoY
 W4uemVobmRlckBiaWdmb290LmNvbTARBglghkgBhvhCAQEEBAMCBaAwDQYJKoZIhvcNA
 QEEBQADggEBAJ5ekE14QJtoO51Z13g3c0w8kPsJxvfVzkgdNbdFtYNhx9e3XGPCqrSr0aymSG
 qC3m1RHkXd4UPY5H7KKp1SD+cMVX4WMxtNOv9ekuFVdCleNjvjvuJ6QbHj64LU2Pw195XJT2S
 Il7Iz7Q2R4gct318Ruc8HEDN+op5nz4w6zbzoh+vycshzse3kGARNU5mdSG70b7CpeANfPUZ17
 tzazndyn0t4H/kIfUPpUtld3YlVFk02NX2bK/wMF/I8Jt3ItBLomK0O4ea0rAW88sZqW54l85
 mWpe/ez4e+/eVIZct5Xtym0MrtCRnwhTQlUO5LXqOGXGBxGvOiRdE0ElNqbkQ=
mail: i7griede@zhwin.ch
```