

## Netzicherheit – Architekturen und Protokolle PKI: X.509



1. X.509-PKI
2. PKIX-Protokolle



## Zertifikatsstandard X.509

- X.509: bekanntester und verbreitetster Zertifikatsstandard
  - ursprünglich Authentifikationsmechanismus für Verzeichnis auf X.500-Basis
  - 5 Versionen, aktuell: X.509-2005 + Korrekturen
  - spezifiziert technisches Framework für eine PKI
    - ▶ Syntax eines Zertifikat in ASN.1
    - ▶ Syntax einer CRL in ASN.1
    - ▶ Validierung einer Zertifikatskette
  - Vielzahl optionaler Parameter und Erweiterungen
  - eine PKI selbst
    - ▶ entscheidet welche Teile von X.509 sie nutzt → *Profile*
    - ▶ kann Standard durch eigene Erweiterungen ergänzen

1



## ITU-T X.509

### Aktiv genutzt von

- SSL/TLS (Schutz von TCP-Verbindungen) (→ Vorlesungsthema)
- S/MIME (Schutz von eMails)
- IPsec/IKE (Schutz von IP-Paketen) (→ Vorlesungsthema)
- SET (Schutz von Kreditkarten-Transaktionen)
- (W)LAN-Sicherheit: 802.1x bzw. 802.11i (→ Vorlesungsthema)
- S-BGP/soBGP/psBGP (Routing-Sicherheit)

2



## Globales Namensschema: Distinguished Names

- Identität des Schlüsselbesitzers ist in X.509 zentral
  - Zertifikat enthält ID der CA
  - Zertifikat enthält ID des Schlüsselbesitzers
  - ID ist Grundlage für Aufbau der Zertifikatskette
- **Distinguished Name** ist hierarchisches Namensschema, das sich aus mehreren Attributen zusammensetzt
  - Land (country – c)
  - Bundesland (state – s)
  - Stadt (locality – l)
  - Name der Firma/Organisation (organisation – o)
  - Abteilung (organisational unit – ou)
  - Name (common name – cn), einziges nicht optionales Attribut
  - weitere...

3



-----BEGIN CERTIFICATE-----

```
MIIDbjCCAtgAwIBAgIQdP1CoYLH8473qHh4EEcR2zANBgkqhkiG9w0BAQQFADBD
MREwDwYDVQQKEWhWZXMjU2lnbjEuMCwGA1UECxMlVmVyaVNPZ24gQ2xhc3MgMiBP
blNpdGUGSW5kaXZpZHVhbnCBQDQTAeFw0wNzEwMzEwMDAwMDAwMDAwODEwMzU5
NTIwMTIwMRswGQYDVQQKDBJORUMGRXVyb3B1IEExpbW10ZWQxRjBEBGNVBAcMPXdx
dy52ZXJpc2lnbi5jb20vcnVwb3NpdG9yeS9DUFMgSW5jb3JwLiBieSBSZWYuLExJ
QUIuTFREKGMpOTkxNTAzBgNVBAsMLENvbXBhbnkgLSBORUMGTGFi3JhdG9yaWVz
IEV1cm9wZSBIZWlkZWxiZXJnMRkwFwYDVQQDDDBtYXJjdXMGc2Nob2VsbGVyMSww
KgYJKoZIhvcNAQkBFh1tYXJjdXMGc2Nob2VsbGVyMSw5Lm51Y2xhYi5ldTCBnzAN
BgkqhkiG9w0BAQEFAAOBjQAwGykCgYEAjWjxP/Nh/00qNWzQPFksoju0x1ltXUpt
HgPssWdZvgt7IfY0d+puDonEVy4qHPVlfnYcg+FE4B2q3D00OedmNO5Oz3smXkN
7B9P11/9XUVbWOI2yww66qETSDZ81B1NceCBhsjzCtz4PJ61pFTqXpopTzo9/Ihf
Z+vDKYBJ//sCAwEAaAOBvzCBvDAJBGNVHRMEAJAAMEQGA1UdIAQ9MDswOQYLYIZI
AYb4RQEHFwIwKjAoBggrBgEFBQcCARycaHR0cHM6Ly93d3cuZmVyaXNpZ24uY29t
L3JwYUd0bGVhbnQ8EBAMCBAAwEYQYJYIZIAYb4QgEBBAQDAgeAMEkGA1UdHwRCMEAw
PqA8oDqGOGH0dHA6Ly9vbnNpdGVjcmwudmVyaXNpZ24uY29tL09uU210ZVB1Ymxp
Yy9MYXRlc3RDUkwuY3J5MA0GCSqGSIb3DQEBAQUAA4GBABowBD+ywCKnnP38oaVa
afLnuo8Rrj+Z1fd95vUNC0yDjwUHQr2dheOUGoP7CZ1S+K38ndIWbGF+s1UK1ZF
a8aW6MOHeqcJv6NC9iaZQoIQ78P6PGYli2B8A6/CV1UmWm21kjin1lVhyNteWTTsZ
HnHKZp8JGAowbNM7NVhCGRwZ
-----END CERTIFICATE-----
```

4

**Version:** 3 (0x2)

**Serial Number:** 74 fd 42 a1 82 c7 f3 8e f7 a8 78 78 10 47 11 db

**Signature Algorithm:** md5RSA

**Issuer:** OU = „VeriSign Class 2 OnSite Individual CA“

O = „VeriSign“

**Validity**

**Not Before:** Wednesday, October 31, 2007 1:00:00 AM

**Not After :** Friday, October 31, 2008 12:59:59 AM

**Subject:** E = „marcus.schoeller@nw.neclab.eu“

CN = „marcus schoeller“

OU = „Company - NEC Laboratories Europe Heidelberg“

OU = „www.verisign.com/repository/CPS Incorp. ... “

O = „NEC Europe Limited“

**Subject Public Key Info:**

**RSA Public Key:** (2048 bit)

30 81 89 02 81 81 00 99 68 f1 ...

**Key Usage:** Digital Signature, Key Encipherment (a0)

**CRL Distribution Point:**

URL=http://onsitecrl.verisign.com/OnSitePublic/LatestCRL.crl

5

Ein ID-Zertifikat hat folgende Struktur

Feldname der ASN.1-Struktur	Beschreibung
version	Versionsnummer des Zertifikatformat
serialNumber	Seriennummer, zusammen mit issuer eindeutig
issuer	ID des Erzeugers des Zertifikates
signatureAlgorithm	Für Signatur genutzter Algorithmus
validity	Gültigkeitsdauer des Zertifikates
subject	X.500-ID des Zertifikatsbesitzers
subjectPublicKeyInfo	Öffentlicher Schlüssel
issuerUniqueIdentifier	Erweiterte ID des Zertifizierenden (v2)
subjectUniqueIdentifier	Erweiterte ID des Besitzers (v2)
extensions	Erweiterungen (v3)

6

- Erweiterungen sind im **extensions**-Feld enthalten und bestehen aus
  - Erweiterungs-ID
  - Critical-Flag: zeigt an ob die Erweiterung kritisch ist
  - Datenwert
- Kennt eine Implementierung eine Erweiterung **nicht**
  - **Nicht-kritisch:** Erweiterung wird ignoriert
  - **Kritisch:** Zertifikat ist ungültig
- Aktuell existierende Erweiterungen lassen sich gliedern in
  - **Informationen** über Schlüssel und Sicherheitsrichtlinien
  - **Attribute** von Zertifikatsbesitzer und signierender CA
  - **Einschränkungen** des Zertifikatspfades

7

- Problem: Schlüssel- bzw. Zertifikats-Identifizierung
  - Standardfelder enthalten nur Aussteller des Zertifikats
- Erweiterung: **Authority key identifier, Subject key identifier**
  - Identifizierung des Aussteller- bzw. Inhaber-Schlüssels
  - mögliche IDs:
    - ▶ Hashwert des Schlüssels
    - ▶ ID + Seriennummer des Zertifikats
  - erlaubt Verwendung mehrerer Schlüssel und Zertifikate
  - vereinfacht Pfad-Konstruktion

8

- Problem: Einschränkung der Schlüssel-Verwendung notwendig
  - z.B. Zertifizierung nicht mit jedem Schlüssel erlaubt
  - unterschiedliche Policies der Zertifizierung
- Erweiterung: **Key usage, Extended key usage**
  - Verwendungszweck des Schlüssels
    - ▶ Key usage: z.B. digitalSignature, keyCertSign, cRLSign, encipher-only
    - ▶ Extended Key usage: z.B. clientAuth, EmailProtection
  - Definition mittels OID (Object Identifier)
  - kann *Critical* sein

9

- Problem: X.500-Namen wenig verbreitet
  - Assoziation von Schlüssel an andere Namen notwendig
- Erweiterung: **Subject-Issuer Alternative Name**
  - alternative Namensformen mit dem Zertifikat assoziierbar
  - Beispiele: eMail-Adresse, IP-Adresse, Domain-Name, URI, AS-Nummer
  - jedes strukturiertes Namensschema möglich

10

- Problem
  - Kontrolle bei der Delegation des Zertifizierungsprivileges ist für komplexere Vertrauensmodelle notwendig!
- Erweiterung: **Basic constraints**
  - Markierung, ob Besitzer eine CA ist
  - maximale Pfadlänge spezifizierbar
- Erweiterung: **Name constraints**
  - nur für CAs
  - gibt den Namensraum an, in dem die CA Zertifikate ausstellen darf (permittedSubtree) bzw. nicht darf (excludedSubtree)
- Erweiterung: **Policy constraints**
  - requireExplicitPolicy: jedes Zertifikat im Pfad muss Zertifizierungsrichtlinien explizit enthalten
  - inhibitPolicyMapping: verbietet die Nutzung äquivalenter Richtlinien

11

## Fragen zum Zertifikatsformat von X.509?

12

## Praxisteil: openssl - Basics

- OpenSSL bietet u.a. die Möglichkeit, X.509v1 und v3 Zertifikate sowie v1- und v2- CRLs zu erzeugen
- Config-Datei `openssl.cnf` beeinflusst viele Operationen
- **Generieren eines Schlüsselpaares** (privater Schlüssel wird mit Hilfe eines Passwortes und 3DES verschlüsselt):  

```
openssl genrsa -des3 -out mykey.pem <Schlüssellänge>
```

 oder  

```
openssl dsaparam <Schlüssellänge> >dsaparam.pem
```

```
openssl gendsa -out mykey.pem -des3 dsaparam.pem
```
- Erstellen eines **Certificate Signing Requests** zur Signatur durch eine CA:  

```
openssl req -new -key mykey.pem -out mycsr.pem
```

```
openssl req -in mycsr.pem -text
```

13

## Praxisteil: openssl – CA

- Erstellen eines selbstsignierten Zertifikates (z.B. für eine CA)  

```
openssl req -new -x509 -key mykey.pem -out mycacert.pem
```

```
openssl x509 -in mycacert.pem -text
```
- Gekripteter Aufbau einer CA (mit openssl mitgeliefert)  

```
CA.sh -newca
```

 (erzeugt eine Verzeichnisstruktur unterhalb von `./demoCA` zur Ablage der Managementdaten und erzeugt ein selbstsigniertes Zertifikat, siehe oben)
- Signieren eines User-Zertifikates  

```
openssl ca -in mycsr.pem -out mycert.pem
```
- CA.sh vereinfacht die meisten Operationen, hier sollten jedoch die genauen Befehle direkt gezeigt werden

14

## X.509-Certificate Revocation Lists

- X.509 spezifiziert **keine Protokolle zur Online-Prüfung**
  - ob ein Zertifikat widerrufen wurde
  - jedoch das **Format** für Widerrufslisten
- Erweiterung: **CRL Distribution Point** im Zertifikat
- Format: Effizienz der Validierung? Skalierbarkeit der PKI?
  - Client-seitig unkritisch (geringen Anzahl von Prüfungen)
  - Server-seitig sehr kritisch (hohe Anzahl von Prüfungen)
- Inhaltlich unterscheidet X.509 zwei CRL-Typen
  - End-entity Public-key certificate Revocation List (EPRL)
  - Certification Authority Revocation List (CARL)
    - ▶ Erlaubt effiziente Prüfung von CA-Zertifikaten (warum?)

15



X.509-CRL hat in Formatsversion v2 folgende Struktur

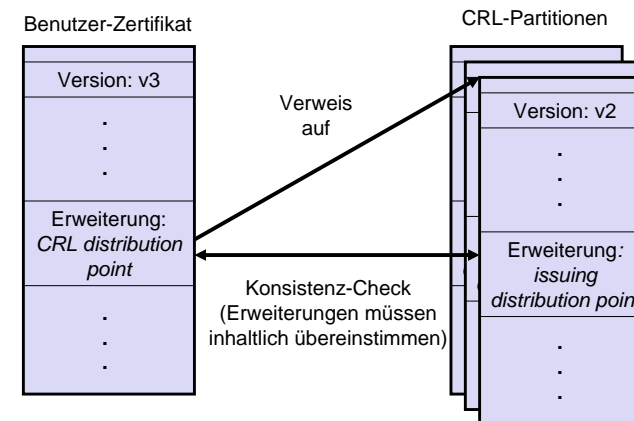
Feldname	Beschreibung
version	Versionsnummer des CRL-Formates (v2)
issuer	ID des Erzeugers der CRL
signatureAlgorithm	für Signatur genutzter Algorithmus
thisUpdate	Zeitpunkt der Ausstellung dieser CRL
nextUpdate	Zeitpunkt der Ausstellung der nächsten CRL
revokedCertificates	Liste widerrufenen Zertifikate
• serialNumber	• Seriennummer
• revocationDate	• Zeitpunkt des Widerrufs
• crlEntryExtensions	• Eintragsspezifische Erweiterungen (v2)
crlExtensions	Globale Erweiterungen (v2)

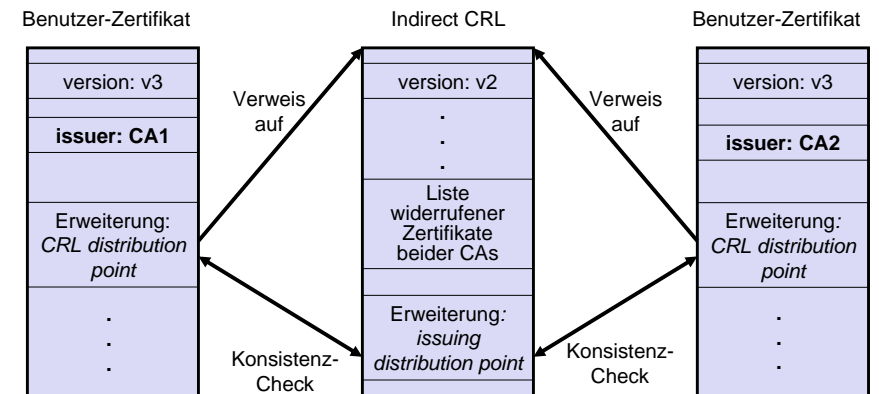
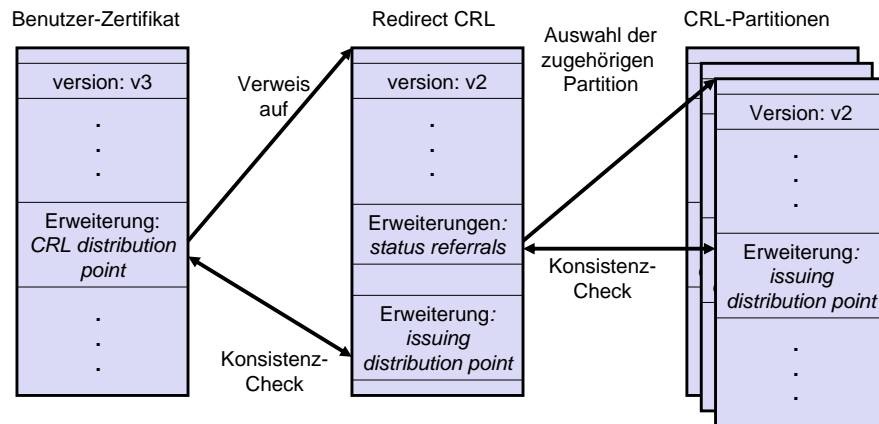
- Erweiterbarkeit der CRLs ab Format v2
  - CRL-weite Erweiterungen: **crlExtensions**
    - ▶ z.B. **CRL Number**, **CRL Scope**, **Issuing distribution point**
  - CRL-Eintrags-spezifische Erweiterungen: **crlEntryExtensions**
    - ▶ z.B. **Reason Code**, **Certificate Issuer**
- Abbruch der Prüfung, wenn
  - Erweiterung unbekannt und
  - Erweiterung kritisch

→ durch in den letzten Versionen von X.509 standardisierte CRL-Erweiterungen wird eine Vielzahl von CRL-Strukturen realisierbar

Folgende CRL-Strukturen lassen sich realisieren:

- Vollständige CRL
- **Partitioned CRLs**
- **Redirect CRL**
- **Delta-CRL (dCRL)**
- **Indirect-CRL (iCRL)**
- **Certificate Revocation Tree (CRT)**





### • Eingabe

- Zertifikatskette
- aktuelle Uhrzeit
- Menge von akzeptablen Sicherheitsrichtlinien bzw. deren OIDs
- Flags
  - ▶ ob Richtlinien explizit in den Zertifikaten enthalten sein müssen
  - ▶ ob Äquivalenzzuweisungen zwischen Richtlinien zulässig sind
  - ▶ ob die Richtlinie anyPolicy zulässig ist

### • Ausgabe

- Erfolg oder Misserfolg der Prüfung
- Fehlercode bei Misserfolg
- Richtlinien, die aufgrund von CA-Einschränkungen gültig sind
- Richtlinien, die aufgrund von Eingabe- und CA-Einschränkungen gültig sind
- Flag, ob die Sicherheitsrichtlinie explizit enthalten sein musste
- Äquivalenzzuweisungen während der Prüfung

### • Basis-Checks

- Korrektheit der Kettenbildung
- Korrektheit der Signaturen
- zeitliche Gültigkeit
- Prüfung jedes Zertifikates auf Widerruf

### • Prüfung von Einschränkungen

- sind Zwischenzertifikate CA-Zertifikate?
- wird die maximale Pfadlänge eingehalten? (vorgegeben durch CA-Zertifikate)
- werden Namenseinschränkungen eingehalten?
- werden Richtlinieneinschränkungen eingehalten?

### Vorteile und Nachteile der X.509-PKI

- 😊 flexibles Zertifikatsformat, viele Vertrauensmodelle realisierbar
- 😊 flexibles CRL-Format, viele CRL-Modelle realisierbar
- 😊 von vielen Anwendungen eingesetzt, praxisnah entwickelt
- ☹ Verwendung von ASN.1
  - ▶ sehr komplex und mächtig
  - ▶ in der Vergangenheit waren Implementierungen geprägt von Sicherheitslöchern (siehe SNMP)
- ☹ Komplexität des Standards

PKI allgemein und X.509 (PKI/PMI-Teil) wurden von vielen prominenten Kryptologen kritisiert:

- Bruce Schneier und Niels Ferguson im Buch [5.3]  
“Practical Cryptography”
- Bruce Schneier und Carl Ellison in *Computer Security Journal*: [5.4]  
“Ten Risks of PKI”
- Carl Ellison auf dem **1st Annual PKI Research Workshop**:  
“Improvements on Conventional PKI Wisdom” [5.5]
- Peter Gutmann in *IEEE Computer*: [5.6]  
“PKI: It’s not dead, just resting”
- Peter Gutmann in [5.7]  
“X.509 Style Guide”

- Widerrufen eines Zertifikates  
`openssl ca -revoke usercert.pem`
- Erstellen einer CRL  
`openssl ca -updatedb`  
`openssl ca -gencrl -crldays 60 -out mycrl.pem`

## Netzicherheit – Architekturen und Protokolle PKI: X.509

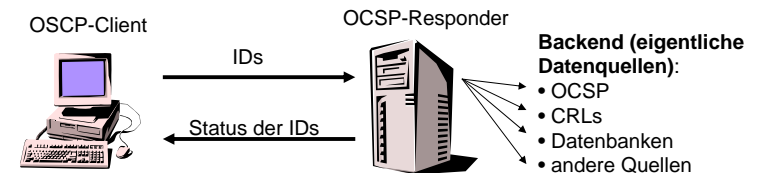


1. X.509-PKI
2. PKIX-Protokolle



- Aufgabe der Working Group ist
  - das Erstellen von Profilen für Zertifikate und CRLs von X.509 für die Nutzung im Internet
  - die Entwicklung von Managementprotokolle zur Online-Realisierung von PKI-Diensten
- Überblick über existierende Standards
  - Profile
  - Managementprotokolle
  - Protokolle zur Status-Prüfung
    - ▶ Online Certificate Status Protocol (OCSP, v1 RFC 2560)
    - ▶ Server-based Certificate Validation Protocol (SCVP, RFC5055)
  - Protokolle für erweiterte PKI-Dienste

- OCSP war der erste Ansatz eines Protokolls zur Online-Prüfung von Zertifikaten auf Widerruf
  - einfaches Frage-Antwort-Schema
  - Erweiterung im Zertifikat: **AuthorityInfoAccess**
    - ▶ u.a. Lokalisierung von OCSP-Respondern und das verwendete Protokoll (z.B. http, ldap, ...)
  - durch die **extendedKeyUsage**-Erweiterung (Wert: **OCSPSigning**) autorisiert die CA den Responder zur Signatur von Antworten



- Von der PKIX-Mailingliste:
- *The main [reference] mechanism(s) at, and shortly after, the time of writing OCSP IDs included:-*
  - (1) VeriSign, who used an **oracle database-based repository** to feed data to OCSP daemons acting in cached and interactive, direct-trust mode; CRLs were not involved. OCSP proxying/multiplexing interactive direct-trust mode was added, shortly after standardization, for a defense customer bridging multiple certification domains.
  - (2) ValiCert, who used **direct CRLs** to feed data to direct/indirect OCSP daemons. Indirect CRLs and CRLDPs support was added slightly after the architects had harmonized their work.

(Anmerkung des Kopierers: CRLDP = CRL Distribution Point)

- OCSP antwortet nur in Bezug auf Widerruf, prüft nicht
  - zeitliche Gültigkeit des Zertifikates
  - korrekter Verwendungszweck des Zertifikates
- On-line vs. Up-to-date
  - Unterschied?
- Signieren der Daten gefährdet Skalierbarkeit
  - CPU-Bedarf fällt synchron und bei jeder Anfrage an, nicht wie bei einer CRL einmalig
  - evtl. Vorbereiten von Antworten, wenn Aktualität ausreichend
- Nur geringe Verringerung der Komplexität der Validierung auf Client-Seite
  - Konstruktion der Zertifikatskette bleibt
  - Validierung der Zertifikatskette bis auf Widerrufprüfung bleibt

siehe  
SCVP



- SCVP soll Client ein **partiell bis vollständiges Auslagern der Zertifikat-Validierung** ermöglichen
- Fokus auf zwei Klassen von Benutzern
  - Auslagerung der Konstruktion, Validierung wird selbst gemacht
  - Vollständige Auslagerung, nur Ergebnis ist interessant
- Protokollseitige Abwicklung von Anfragen zur
  - Konstruktion einer Zertifikatskette (**Delegated Path Discovery**)
  - Validierung einer Zertifikatskette (**Delegated Path Validation**)

32

SCVP folgt wie OCSP dem einfachen Frage-Antwort-Modell und spezifiziert zwei Abläufe

1. Client befragt den Server nach unterstützten **Validierungsrichtlinien** (validation policy)
2. Client beauftragt Server mit der (teilweisen) Validierung von Zertifikaten

33

## Fragen zu PKIX-Protokollen?

34


- Ziel von X.509-ID-Zertifikaten
  - Vertrauen in Identität des Zertifikatsinhabers herstellen
- X.509-Standard kennt nur zwei Vertrauensstufen
  - Vertrauen besteht / besteht nicht
- Wunsch aus der Praxis: **höheres Maß an Vertrauen für bestimmte Anwendungen**
  - Online-Banking u.ä.
- Wie lässt sich das mit X.509 umsetzen?



35

- Definition einer Richtlinie mit technischen und organisatorischen Anforderungen
  - insbesondere an Durchführung der Identitätsprüfung durch die CA
- Zertifikate enthalten **certificatePolicies**-Erweiterung
  - im Wesentlichen enthalten
    - ▶ OID der Policy der CA für EV-Zertifikate
    - ▶ URL des **Certification Practice Statement**
- Prüfende Instanz kennt OIDs der EV-Policies vertrauter CAs
- Ergebnis: weitere Vertrauensstufe
  - sollte dem Nutzer deutlich angezeigt werden

36

→ Schwächste Glied der Kette bestimmt Gesamtsicherheit  [5.8]

- Verisign
  - stellte Code-Signing-Zertifikate für eine Firma Microsoft aus
  - Routine-Check stellte fest, dass die Zertifikate fälschlicher Weise ausgestellt wurden
  - Zertifikate wurden zurückgezogen, via CRL bekannt gegeben
- Microsoft-OSe erfuhren von dem Widerruf nichts
  - Windows enthält die CA-Zertifikate von Verisign
  - Verisign-Zertifikate enthalten keine Erweiterung CrlDistributionPoint (weil Verisign-PKI schon älter als X.509v3)
  - CRL ist unter bekannter und dokumentierter URL zu finden, die jedoch durch Windows nicht genutzt wurde

→ Ergebnis: Windows kann einige Verisign-Zertifikate nicht auf Widerruf prüfen

37

- [5.1] C. Kaufmann, R. Perlman, M. Speciner; Network Security – Private Communication in a public world; Prentice Hall; 2003
- [5.2] ITU-T Recommendation X.509, 2000.URL:  
<http://www.itu.int/ITU-T/asn1/database/itu-t/x/x509/2005/index.html>
- [5.3] N. Ferguson, B. Schneier: Practical Cryptography, Wiley, 2003.
- [5.4] B. Schneier, C. Ellison: Ten Risks of PKI: What You're not Being Told about Public Key Infrastructure, Computer Security Journal 16 (1), S. 1-7, 2000.
- [5.5] C. Ellison: Improvements on Conventional PKI Wisdom, Proceedings of the 1st Annual PKI Research Workshop, online verfügbar:  
<http://www.cs.dartmouth.edu/~pki02/>
- [5.6] P. Gutmann: PKI: it's not dead, just resting, IEEE Computer 35 (8), S. 41-49, August 2002.
- [5.7] P. Gutmann: X.509 Style Guide, 2000. URL  
<http://www.cs.auckland.ac.nz/~pgut001/pubs/x509guide.txt>
- [5.8] Microsoft warnt vor Cracker-Zertifikat, Meldung im Heise-Newsticker vom 24.3.2001, URL <http://www.heise.de/newsticker/meldung/16482>

38