

# **INDUSTRIAL SIGNATURE INTEROPERABILITY SPECIFICATION ISIS**

**Version 1.2**

**03.12.1999**

**Arbeitsgemeinschaft Trust-Center  
für digitale Signaturen**



# INHALTSVERZEICHNIS

<b>EINLEITUNG.....</b>	<b>4</b>
<b>ABSCHNITT I: ZERTIFIKATE .....</b>	<b>6</b>
<b>Z 1. EINFÜHRUNG ZUM TEIL ZERTIFIKATE .....</b>	<b>6</b>
<b>Z 2. SIGNATURSCHLÜSSEL-ZERTIFIKATE.....</b>	<b>12</b>
Z 2.1. SIGNATALGORITHMUS .....	14
Z 2.2. SIGNATUR EINES ZERTIFIKATES .....	16
Z 2.3. ZU SIGNIERENDE ZERTIFIKATSINFORMATIONEN.....	17
Z 2.3.1. Versionsnummer .....	19
Z 2.3.2. Seriennummer .....	20
Z 2.3.3. Signatur .....	21
Z 2.3.4. Technische Namen von Zertifizierungsstellen.....	22
Z 2.3.5. Gültigkeitsdauer.....	24
Z 2.3.6. Namen von Zertifikatsinhabern.....	26
Z 2.3.7. Öffentliche Schlüssel von Zertifikatsinhabern.....	30
Z 2.3.8. Eindeutige Bezeichner.....	34
Z 2.3.9. Erweiterungen.....	35
Z 2.3.9.1. Zertifizierungsstellen- und Endbenutzer-Zertifikate .....	38
Z 2.3.9.2. Verwendungszwecke des Schlüsselpaares .....	40
Z 2.3.9.3. Anwendungsabhängige Verwendungszwecke des Schlüsselpaares.....	43
Z 2.3.9.4. Zertifizierungsrichtlinien.....	46
Z 2.3.9.5. Alternative Namen von Zertifikatsinhabern.....	49
Z 2.3.9.6. Alternative Namen von Zertifizierungsstellen.....	53
Z 2.3.9.7. Identifizierung von Signaturschlüsseln von Zertifizierungsstellen.....	55
Z 2.3.9.8. Identifizierung von öffentlichen Teilnehmerschlüsseln.....	57
Z 2.3.9.9. Informationen zur Beschaffung von Sperrlisten.....	59
Z 2.3.9.10. Anerkennung von fremden Zertifizierungsrichtlinien.....	61
Z 2.3.9.11. Verzeichnisattributwerte für Zertifikatsinhaber .....	63
Z 2.3.9.12. Beschränkungen von Zertifizierungsrichtlinien.....	65
Z 2.3.9.13. Namensraum für Namen von Zertifikatsinhabern in Zertifikatketten.....	66
Z 2.3.9.14. Nutzungsdauer von privaten Schlüsseln.....	68
Z 2.3.9.15. Private Zertifikatserweiterungen.....	69
Z 2.3.9.15.1. Zugriff auf Informationen und Dienste durch Zertifizierungsstellen .....	69
Z 2.3.9.15.2. Kennzeichnung der Nutzungsbeschränkung des Signaturschlüssels .....	72
Z 2.3.9.15.3. Erstellungsdatum eines Zertifikates .....	73
Z 2.3.9.15.4. Vertretungsmacht .....	74
Z 2.3.9.15.5. Zulassung.....	76
Z 2.3.9.15.6. Monetäre Beschränkung .....	78
Z 2.3.9.15.7. Volljährigkeit.....	80
Z 2.3.9.15.8. Chipkarten-Seriennummer.....	81
Z 2.3.9.15.9. Chipkarten-Referenzierung öffentlicher Schlüssel.....	83
Z 2.3.9.15.10. Sonstige Einschränkungen.....	84

Z 2.3.9.15.11.	Vergabe weiterer privater Erweiterungen .....	85
<b>Z 3.</b>	<b>ANHANG <sup>2</sup> NAMENSKONVENTIONEN .....</b>	<b>86</b>
Z 3.1.	ANHANG <sup>2.1</sup> EINDEUTIGE IDENTIFIZIERUNG VON SIGNATURSCHLÜSSELLINHABERN.....	86
Z 3.2.	ANHANG <sup>2.2</sup> PSEUDONYMISIERUNG VON SIGNATURSCHLÜSSELINHABERN.....	87
Z 3.3.	ANHANG <sup>2.3</sup> BENUTZUNG VON DISTINGUISHED NAMES.....	87
Z 3.4.	ANHANG <sup>2.4</sup> BENUTZUNG VON E-MAIL-ADRESSEN .....	89
Z 3.5.	ANHANG <sup>2.5</sup> X.500 DIRECTORY DISTINGUISHED NAMES.....	89
<b>Z 4.</b>	<b>ANHANG <sup>2I</sup> ERMITTLUNG VON DIENSTADRESSEN VON ZERTIFIZIERUNGSSTELLEN.....</b>	<b>90</b>
<b>Z 5.</b>	<b>ANHANG <sup>2II</sup> OBJEKTBEZEICHNER.....</b>	<b>92</b>
<b>Z 6.</b>	<b>ANHANG IV ASN.1 DEFINITIONEN.....</b>	<b>95</b>
	<b>ABSCHNITT II: VERZEICHNISDIENSTE.....</b>	<b>105</b>
<b>V 1.</b>	<b>EINLEITUNG ZUM TEIL VERZEICHNISDIENSTE .....</b>	<b>105</b>
<b>V 2.</b>	<b>VERZEICHNISDIENST .....</b>	<b>109</b>
V 2.1.	ANFORDERUNGEN AUS DEM SIGNATURGESETZ UND DER SIGNATURVERORDNUNG .....	109
V 2.2.	ONLINE-VERZEICHNISDIENST OCSP .....	111
V 2.2.1.	Anfragen an den Verzeichnisdienst.....	112
V 2.2.2.	Antworten des Verzeichnisdienstes .....	118
V 2.3.	TRANSPORT VON VERZEICHNISDIENSTANFRAGEN ÜBER HTTP .....	129
V 2.3.1.	Definitionen für die Anfragen.....	129
V 2.3.2.	Definitionen für die Antworten.....	130
V 2.3.2.1.	Abspeichern von Antworten.....	131
V 2.3.3.	Verwendung von Proxies.....	131
V 2.3.3.1.	Verwendung von HTTP Proxy Servern.....	131
V 2.3.3.2.	Verwendung von SSL Proxy Servern.....	131
V 2.3.3.3.	Verwendung eines Verzeichnisdienst Proxies.....	131
V 2.3.3.4.	Dienstäquivalenz von Verzeichnisdiensten.....	132
V 2.4.	TRANSPORT VON VERZEICHNISDIENSTANFRAGEN ÜBER E-MAIL.....	132
<b>V 3.</b>	<b>SPERRLISTENMANAGEMENT .....</b>	<b>133</b>
V 3.1.	SPERRLISTENFORMATE .....	133
V 3.1.1.	Signaturalgorithmus .....	133
V 3.1.2.	Signatur einer Sperrliste.....	135
V 3.1.3.	Zu signierende Sperrlisteninformationen.....	136
V 3.1.3.1.	Versionsnummer.....	137
V 3.1.3.2.	Signatur.....	138
V 3.1.3.3.	Namen von Sperrlistenerstellern.....	139

---

V 3.1.3.4.	Datum und Zeitpunkt der Erstellung von Sperrlisten.....	141
V 3.1.3.5.	Datum und Zeitpunkt der Erstellung der nächsten Sperrliste.....	143
V 3.1.3.6.	Sperrlisteneinträge .....	144
V 3.1.3.6.1.	Erweiterung der Sperrlisten-Einträge.....	146
V 3.1.3.7.	Sperrlistenerweiterungen.....	152
V 3.1.3.7.1.	Identifizierung von Signaturschlüsseln von Zertifizierungsstellen.....	153
V 3.1.3.7.2.	Alternative Namen von Sperrlistenerstellern.....	156
V 3.1.3.7.3.	Sperrlistennummern .....	158
V 3.1.3.7.4.	Identifikation der Quellen von Sperrlisten.....	159
V 3.1.3.7.5.	Indikator von Sperrlistenänderungen.....	160
V 3.2.	VERWALTUNG UND BEREITSTELLUNG VON SPERRLISTEN .....	163
V 3.2.1.	CDP (Certificate Distribution Point).....	163
V 3.2.2.	OpenCDP (Open CRL Distribution Process).....	165
V 3.2.2.1.	CRL-Erweiterung cRLScope.....	165
V 3.2.2.2.	X.500-Attribut revocation information attribute .....	167
V 3.2.2.3.	X.500-Attribut CRL list attribute .....	168
V 3.2.2.4.	Erweiterung revocation issuer .....	168
V 3.2.3.	OCSP (Online Certificate Status Protocol) auf der Basis von CRLs .....	169
V 3.2.4.	Abfrage von Sperrlisten.....	170
<b>V 4.</b>	<b>ANHANG <sup>2</sup> OBJEKTBEZEICHNER .....</b>	<b>171</b>
<b>V 5.</b>	<b>ANHANG <sup>2 2</sup> ASN.1 DEFINITIONEN .....</b>	<b>173</b>
	<b>ABSCHNITT III: ALLGEMEINES.....</b>	<b>181</b>
<b>A 1.</b>	<b>ABKÜRZUNGEN UND BEGRIFFE.....</b>	<b>181</b>
<b>A 2.</b>	<b>LITERATUR .....</b>	<b>190</b>

## **EINLEITUNG**

In der Arbeitsgemeinschaft Trust-Center für digitale Signaturen, kurz AGTC, haben sich die Industrieunternehmen zusammengeschlossen, die Dienstleistungen als Zertifizierungsstellen im Sinne des Signaturgesetzes (SigG) anbieten bzw. anbieten werden.

Die AGTC legt hiermit die „Industrial Signature Interoperability Specification“ (ISIS) vor. Diese Spezifikation legt einheitliche Formate für Daten und Nachrichten fest, die bei Dienstleistungen im Sinne des SigG verwendet werden.

In der vorliegenden Spezifikation werden Formate für Zertifikate (siehe Abschnitt I) und für Verzeichnisdienste (siehe Abschnitt II) festgelegt. Weitere Erläuterungen zu diesen Teilen finden sich am Beginn des jeweiligen Abschnitts.

Es ist vorgesehen, das vorliegende Dokument sukzessive um Festlegungen für weitere Formate, etwa für Zeitstempeldienste, zu erweitern.

Die Erstellung der Spezifikation wurde mit Unterstützung des Bundesamtes für Sicherheit in der Informationstechnik durchgeführt.

Die Arbeitsgemeinschaft Trust-Center für digitale Signaturen besteht aus folgenden Firmen:

Bundesdruckerei

CCI Competence Center Informatik GmbH

debis Systemhaus Information Security Services GmbH

Deutsche Post AG

D-Trust GmbH

Gieseke + Devrient GmbH

TC Trust Center

TeleCash

TeleSec Deutsche Telekom AG

Die Erstellung der Spezifikation wurde weiterhin von folgenden Organisationen unterstützt:

Bundesamt für Sicherheit in der Informationstechnik (BSI)

# Abschnitt I: Zertifikate

## Z 1. EINFÜHRUNG ZUM TEIL ZERTIFIKATE

### BEDEUTUNG VON ZERTIFIKATEN

Digitale Signaturen werden in der elektronischen Welt verwendet um Sicherheitsziele wie Authentizität, Verbindlichkeit und Integrität erreichen zu können. Digitale Signaturen arbeiten mit zwei Schlüsseln, die gemeinsam erstellt und mathematisch voneinander abhängig sind. Einer dieser Schlüssel wird geheimgehalten und kann zur Erstellung einer digitalen Signatur verwendet werden. Der andere Schlüssel wird veröffentlicht und kann zur Verifikation einer geleisteten Signatur verwendet werden. Um digitale Signaturen Personen zuzuordnen, bedarf es einer Bindung des Namens einer Person an den entsprechenden öffentlichen Schlüssel. Diese Bindung erfolgt in der Form eines speziellen digitalen Dokumentes, welches von einer vertrauenswürdigen dritten Instanz ausgestellt wird. Diese Dokumente, üblicherweise als Zertifikate bezeichnet, können als "digitaler Ausweise" in Analogie zu beispielsweise einem Personalausweis angesehen werden.

Technisch gesehen sind Zertifikate Datenstrukturen, die Informationen enthalten, mit denen eine Bindung von öffentlichen Schlüsseln an Schlüsselinhaber gewährleistet wird. Die konkrete Bindung eines öffentlichen Schlüssels an einen bestimmten Schlüsselinhaber wird durch eine vertrauenswürdige und neutrale Zertifizierungsstelle (CA, certification authority) bewerkstelligt, die das zugehörige vollständige Zertifikat mit ihrer digitalen Signatur beglaubigt. Zertifikate haben nur eine begrenzte Gültigkeitsdauer, die ebenfalls als Bestandteil des Zertifikates von der Zertifizierungsstelle mitsigniert ist.

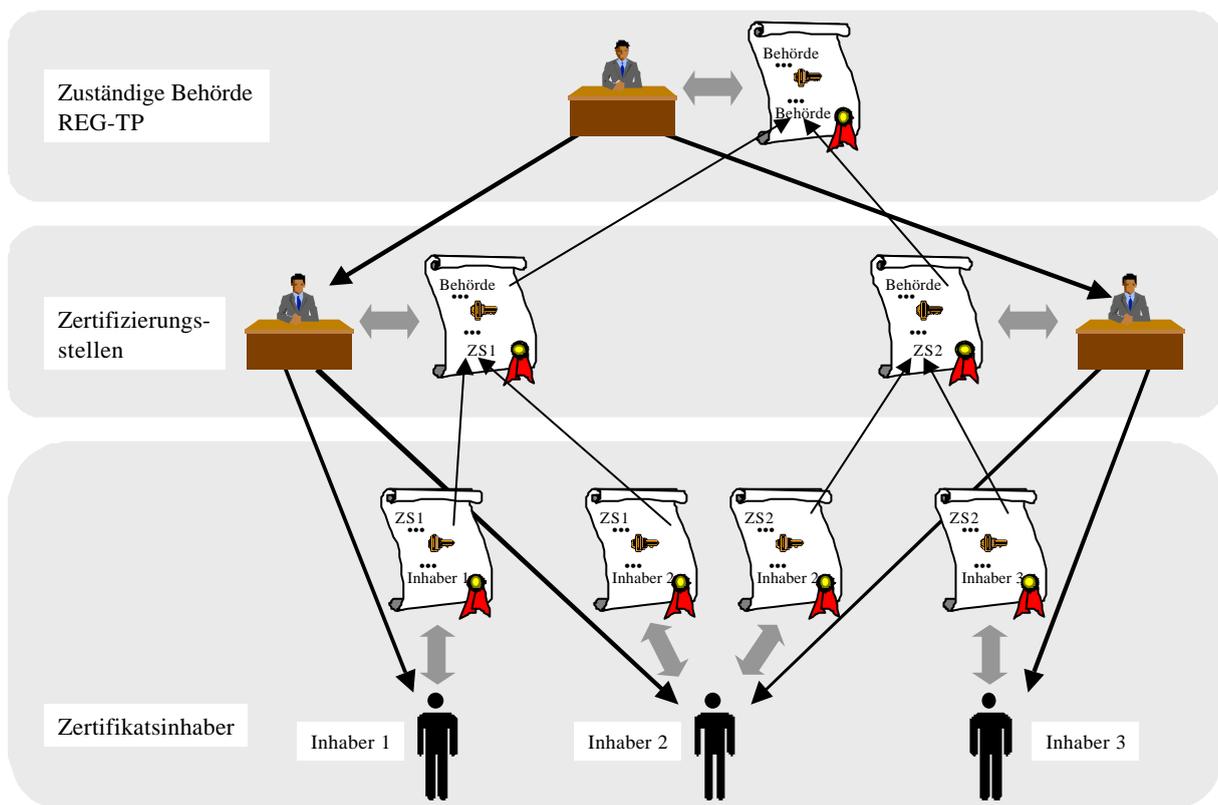
Die Zertifizierungsstelle übernimmt die Prüfung des Namens und bindet durch eine digitale Signatur (mit ihrem privaten Schlüssel) den Namen der Person an den öffentlichen Schlüssel dieser Person. Das Resultat der Zertifizierung eines öffentlichen Schlüssels ist ein Zertifikat. Als Zertifikatsstruktur wird der Standard X.509 benutzt. Solch ein Zertifikat enthält neben dem öffentlichen Schlüssel u. a. den Namen der ausstellenden Zertifizierungsstelle, einen Gültigkeitszeitraum, den Namen des Eigentümers und eine eindeutige Nummer der ausstellenden Zertifizierungsstelle. Hierbei wird vorausgesetzt, daß alle beteiligten Personen dem öffentlichen Schlüssel dieser Zertifizierungsstelle vertrauen. Zertifizierungsstellen besitzen getrennte Schlüsselpaare für das Signieren von Zertifikaten, Sperrlisten und Zeitstempeln sowie für die Abwicklung der Kommunikation mit anderen Kommunikationspartnern.

Eine einzelne Zertifizierungsstelle ist insbesondere bei sehr großen Teilnehmerzahlen aus praktischen und organisatorischen Gründen i.a. nicht dazu in der Lage, flächendeckende Zertifizierungsdienste zu erbringen. Dieses Problem läßt sich durch komplexere Systeme von Zertifizierungsstellen in Form von Baum- oder Netzstrukturen lösen, wobei einzelne Zertifizierungsstellen in unterschiedlichen Rollen agieren können. In einer solchen Hierarchie von Zertifizierungsstellen übernimmt eine spezielle Zertifizierungsstelle (root certification authority) die Rolle einer sogenannten Wurzelzertifizierungsstelle als höchste Instanz, der keine weiteren Zertifizierungsstellen übergeordnet sind und die Zertifikate für untergeordnete Zertifizierungsstellen ausstellen kann. Untergeordnete Zertifizierungsstellen können Zertifikate für

ihnen unterstellte Zertifizierungsstellen oder für Teilnehmer ausstellen. Durch das Signaturgesetz wird das Spektrum möglicher Strukturen von Zertifizierungsstellen dahingehend eingeschränkt, daß eine zweistufige Hierarchie von Zertifizierungsstellen festgelegt wird, in der die zuständige Behörde (RegTP, Regulierungsstelle für Telekommunikation und Post) die Rolle der höchsten Zertifizierungsstelle ausführt und es unter ihr nur eine Ebene von Zertifizierungsstellen gibt, die nur Zertifikate für Teilnehmer ausstellen können.

Das grundlegende Prinzip der Vertrauensbildung auf der Basis von Signaturschlüssel-Zertifikaten ist in der Abbildung 1 für signaturgesetzeskonforme Sicherheitsinfrastrukturen dargestellt. Hierbei symbolisieren Doppelpfeile die Bindung der öffentlichen Signaturschlüssel an die zugehörigen Personen bzw. Zertifizierungsstellen. Dicke Pfeile zeigen vom Ersteller auf den Inhaber eines Zertifikates und veranschaulichen den Prozeß der Zertifikaterstellung. Dünne Pfeile repräsentieren den Vertrauenspfad, der über die Namen der ausstellenden Zertifizierungsstellen, die u.a. in den Zertifikaten enthalten sind, nachvollziehbar konstruiert und überprüft werden kann.

**Abbildung 1: Zertifizierungshierarchie**



Zertifikate werden in einer Sicherheitsinfrastruktur von den unterschiedlichsten Anwendungen und Systemen benötigt und benutzt. In diesem Zusammenhang sind deshalb eine Reihe von Themen wie Namenskonventionen, Identifizierung von öffentlichen Schlüsseln von Zertifizierungsstellen und Zertifikatsinhabern, Zertifizierungsrichtlinien, Signaturverfahren, Attribute von Zertifikatsinhabern und die Möglichkeit der proprietären Zertifikaterweiterungen von großer Bedeutung.

Zertifikate sind elektronische Dokumente, die – beispielsweise zur Prüfung einer Signatur – vollautomatisch verarbeitet werden müssen. Deshalb müssen sie in ihrer Datenstruktur genau definiert sein. Alle zur Verarbeitung notwendigen Angaben müssen sich in ihnen speichern lassen.

Aus den genannten Themenbereichen ergeben sich zahlreiche Anforderungen an Zertifikatsstrukturen und Formate, die durch die internationale X.509-Zertifikatsnorm in ihrer neuesten Version berücksichtigt und gelöst sind. Auf die wichtigen Themen im Zusammenhang mit Zertifikaten wird in den folgenden Abschnitten dieses Dokumentes näher eingegangen. Das international standardisierte Zertifikatsformat nach X.509 in der Version 3 ist in der Lage, die gesamte Palette dieser Informationen in ihrer Komplexität abzubilden. Da der internationale Standard eine Vielzahl von wahlfreien Angaben zuläßt, muß der Standard für eine spezielle Anwendung – hier für das Signaturgesetz – konkretisiert werden. Diese Konkretisierung erfolgt im Rahmen eines sogenannten “Profils”. In diesem Profil werden Richtlinien festgelegt, wie bestimmte Wahlmöglichkeiten verwendet werden sollten.

## NORMEN UND RICHTLINIEN FÜR ZERTIFIKATE

### Normen für Zertifikate

Als generelles Format für Zertifikate wurde 1988 von der ITU-T (telecommunication standardization sector of the international telecommunication union) die Empfehlung X.509 als Bestandteil der X.500-Directory-Serie verabschiedet, die in der Zwischenzeit durch zwei weitere Versionen ergänzt wurde. Das ursprüngliche X.509-Standardformat wird heute als X.509 v1-Zertifikatsformat bezeichnet und diente als Grundlage für die Entwicklung des Internet-Reports für sichere elektronische Post (PEM, privacy enhanced mail) [RFC 1422 93].

In einer überarbeiteten zweiten Version, bezeichnet als X.509v2, wurden 1993 die neuen optionalen Felder *issuerUniqueIdentifier* (eindeutiger Bezeichner für Zertifizierungsstellen) und *subjectUniqueIdentifier* (eindeutiger Bezeichner für Zertifikatsinhaber) der Zertifikatsstruktur hinzugefügt. Bei dem Entwurf und der Realisierung von PEM traten weitere Schwächen und Mängel an X.509v1 und X.509v2 auf, die durch die dritte Version X.509 v3 [ITU-T X.509 97| ISO/IEC 9594-8 97] behoben wurden. In diesem neuen Format wurde das optionale Zertifikatserweiterungsfeld *extensions* hinzugefügt.

### Profile für Zertifikate in Sicherheitsinfrastrukturen

Der grundlegende Standard für Zertifikate ist derzeit X.509v3, der von der PKIX-Arbeitsgruppe der IETF (internet engineering task force) zur Entwicklung eines entsprechenden PKI-Profiles (PKI, public key infrastructure) [RFC 2459 99] benutzt wurde. In diesem Zusammenhang wurde von NIST (national institute of standards and technology), einem nationalen US-Institut, das für Normen und deren technische Umsetzung und Anwendungen zuständig ist, die Spezifikation “Minimum Interoperability Specification for PKI Components” [MISPC 97] entwickelt. Sie soll in den USA als Grundlage für die Zusammenarbeit zwischen PKI-Komponenten verschiedener Hersteller dienen und wird für die Realisierung einer NIST-Referenzimplementation und die Errichtung einer Wurzelzertifizierungsstelle der US-Bundes-PKI benutzt.

Der Zweck eines Profils besteht darin, relevante Normen und Empfehlungen für die praktische Entwicklung interoperabler Verfahren und Komponenten zu interpretieren und nutzbar zu machen, ohne dabei die Basisnormen zu verletzen. Bei dieser Vorgehensweise spielt die breite Anwendbarkeit von X.509-Zertifikaten in den unterschiedlichsten Anwendungen und Systemumgebungen eine entscheidende Rolle, für die letztendlich die *extensions*-Erweiterungen von Zertifikatsformaten eingeführt wurden.

Eine Sicherheitsinfrastruktur sollte deshalb ein möglichst hohes Maß an Flexibilität hinsichtlich der zugelassenen Formate und Verfahren bieten und unnötige Einschränkungen vermeiden. Profile bieten hierzu den geeigneten Mechanismus, um die extremen und teilweise konträren Anforderungen an Anwendungsbreite, Flexibilität, Interoperabilität und Realisierbarkeit zu erfüllen.

Prinzipiell wird durch das Signaturgesetz kein bestimmtes Zertifikatsformat festgeschrieben, sondern das ausschließliche Ziel verfolgt, Rahmenbedingungen für digitale Signaturen mit hohen Sicherheitsanforderungen zu schaffen. Aus diesem Grund sind neben dem X.509-Zertifikatsformat auch andere Zertifikatsformate wie z. B. EDIFACT-Formate als zulässige Formate zu betrachten, sofern sie die Anforderungen des Signaturgesetzes erfüllen.

An dieser Stelle sei darauf hingewiesen, daß von der PKIX-Arbeitsgruppe der IETF im Dezember 1998 ein neuer Arbeitsschwerpunkt (work item) mit dem Titel "Qualified Certificates" gestartet wurde, der die Erstellung von Zertifikaten unter Berücksichtigung rechtlicher Aspekte behandelt.

Neben den Signaturschlüssel-Zertifikaten, die nach X.509v3 kodiert werden, werden sogenannte "*card verifiable certificates*" (CV-Zertifikate) zur Authentisierung von Terminal und Chipkarte benötigt. In diesem Zusammenhang wird auf die *DIN Spezifikation der Schnittstelle zu Chipkarten mit Digitaler Signatur-Anwendung/Funktion nach SigG und SigV* [DIN SigG/V 98] hingewiesen. Diese Zertifikatstypen liegen außerhalb des Signaturgesetzes, da die Terminal- und Chipkarten-Authentisierungs-Zertifikate nicht für natürliche Personen ausgestellt werden.

Das Signaturgesetz legt ausschließlich die Rahmenbedingungen für die technische Realisierung von digitalen Signaturen fest und aus diesem Grund werden in diesem Teil der ISIS-Spezifikation auch nur Profilverfestlegungen für die Strukturen und Formate von Signaturschlüssel-Zertifikaten behandelt. Zertifikate für andere Schlüsselnutzungsarten wie beispielsweise zur Ver- und Entschlüsselung von Daten sind nicht Gegenstand diese Profils.

#### Anforderungen an Zertifikate, die sich aus dem Signaturgesetz und der Signaturverordnung ergeben

Das Signaturgesetz [SigG 97, §2] unterscheidet zwischen Signaturschlüssel-Zertifikaten und Attribut-Zertifikaten. Beide Zertifikatstypen müssen von einer Zertifizierungsstelle oder der zuständigen Behörde, die für die Akkreditierung von Zertifizierungsstellen zuständig ist, durch eine digitale Signatur beglaubigt werden. Desweiteren legt die Signaturverordnung [SigV, §8] fest, daß ausländische Zertifikate von der zuständigen Behörde durch deren digitale Signatur anzuerkennen sind.

Nach dem Signaturgesetz [SigG 97, §2] kann eine Zertifizierungsstelle eine natürliche oder juristische Person sein, während ein Zertifikatsinhaber nur eine natürliche Person sein kann. In einer öffentlichen Sicherheitsinfrastruktur werden aber auch Zertifikate für Systeme als Zertifikatsinhaber benötigt, wie z.B. Serverzertifikate. Das Signaturgesetz regelt nur den Bereich der digitalen Signatur von natürlichen Personen, Serverzertifikate werden bei dieser Regelung nicht betrachtet. Zertifizierungsstellen dürfen neben signaturgesetzkonformen Zertifikaten unter Beachtung bestimmter Randbedingungen auch Zertifikate ausstellen, die den Anforderungen des Signaturgesetzes nicht genügen. Um signaturgesetzkonforme von nicht signaturgesetzkonformen Zertifikaten unterscheiden zu können, müssen die jeweils verwendeten Sicherheitsmaßnahmen und Sicherheitsrichtlinien in den Zertifikaten kenntlich gemacht werden.

Nach dem Signaturgesetz [SigG 97, §7] muß ein Signaturschlüssel-Zertifikat die folgenden Angaben enthalten: den Name des Signaturschlüsselinhabers mit einem Zusatz bei Verwechslungsmöglichkeit oder ein unverwechselbares erkennbares Pseudonym, den öffentlichen Signaturschlüssel, die Bezeichnung der Algorithmen zur Benutzung der öffentlichen Schlüssel, die Seriennummer des Zertifikates, den Beginn und das Ende der Gültigkeit des Zertifikates, den Name der ausstellenden Zertifizierungsstelle und optionale Angaben zur Beschränkung der Schlüsselnutzung. Desweiteren können Angaben zur Vertretungsmacht für eine dritte Person sowie zur berufsrechtlichen oder sonstigen Zulassung in das Signaturschlüssel- oder Attribut-Zertifikat aufgenommen werden. Weitere Angaben darf das Signaturschlüssel-Zertifikat nur mit Einwilligung der Betroffenen enthalten.

Die Gültigkeit eines Zertifikates ist nach der Signaturverordnung [SigV, §7] auf maximal fünf Jahre begrenzt.

Eine Zertifizierungsstelle muß nach [SigG, §5] auf Verlangen eines Antragstellers Angaben über seine Vertretungsmacht für eine dritte Person sowie zur berufsrechtlichen oder sonstigen Zulassung in das Signaturschlüssel-Zertifikat oder ein Attribut-Zertifikat aufnehmen, soweit ihr die Einwilligung des Dritten zur Aufnahme dieser Vertretungsmacht oder die Zulassung zuverlässig nachgewiesen wird. Desweiteren muß eine Zertifizierungsstelle auf Verlangen eines Antragstellers im Zertifikat anstelle seines Namens ein Pseudonym aufführen. Für die Auflösung von Pseudonymen im Falle eines Disputes sind bei den Zertifizierungsstellen nach [SigG, §12] geeignete Maßnahmen zu ergreifen.

Durch das Signaturgesetz [SigG 97, §7] werden somit spezielle Angaben gefordert, die als Bestandteile in Signaturschlüssel-Zertifikaten und/oder in Attribut-Zertifikaten enthalten sein müssen. Diese Angaben müssen durch geeignete Zertifikatskomponenten realisiert werden. Außerdem werden im Rahmen dieses Profils (Signaturgesetz-Interoperabilität) noch weitere spezielle Angaben benötigt, wie z.B. das Erstellungsdatum eines Zertifikates oder eine Schlüsselnutzungsart für Verzeichnisdienste. Hierfür wurden ISIS-spezifische Objektbezeichner unter dem Objektbezeichnerzweig *id-isis* festgelegt, der seinerseits ein Zweig des TeleTrust-Vereins ist. Unter diesem Zweig wurden die Zweige *cp* (1) für Zertifizierungsrichtlinien, *kp* (2) für Schlüsselnutzungsarten sowie *at* (3) für Attribute und private Erweiterungen definiert.

In der folgenden Tabelle sind alle unmittelbaren Anforderungen des Signaturgesetzes und der Signaturverordnung zusammengestellt, die in diesem Dokument behandelt werden und durch Verweise auf die entsprechenden Abschnitte dieses Dokuments ergänzt.

**Tabelle 1: Anforderungen an Signaturschlüssel-Zertifikate**

#	ANFORDERUNG	REFERENZ
(1)	Name des Signaturschlüsselinhabers	2.3.6, 2.3.9.5, 3.3
(2)	Zusatz bei Verwechslungsmöglichkeit des Namens	2.3.6, 4.1
(3)	Pseudonym statt Name des Signaturschlüsselinhabers	2.3.9.5, 4.2
(4)	Unverwechselbarkeit eines Pseudonyms	2.3.9.5, 4.2
(5)	Erkennbarkeit eines Pseudonyms	2.3.9.5, 3.9.4, 4.2
(6)	Öffentlicher Signaturschlüssel	2.3.7
(7)	Bezeichnung der Algorithmen, mit denen der öffentliche Schlüssel des Signaturschlüsselinhabers sowie der öffentliche Schlüssel der Zertifizierungsstelle benutzt werden kann	2.1, 2.3.7, 2.3.9.4
(9)	Laufende Nummer des Zertifikates	2.3.2, 3.6
(10)	Beginn und Ende der Gültigkeit des Zertifikates	2.3.5, 3.7
(11)	Name der Zertifizierungsstelle	2.3.4, 2.3.9.6, 3.4
(12)	Weitere Angaben zum Signaturschlüsselinhaber oder zur Zertifizierungsstelle	2.3.9.5, 2.3.9.6
(13)	Kennung, ob eine Nutzungsbeschränkung vorliegt	2.3.9.15.2
(14)	Nutzungsbeschränkung nach Art und Umfang	2.3.9.2, 2.3.9.3, 2.3.9.15.6
(15)	Vertretungsmacht für dritte Person,	2.3.9.15.4, 3.9.1
(16)	Berufsrechtliche Zulassungsinformation	2.3.9.15.5, 3.9.2
(17)	Sonstige Zulassungsinformation	2.3.9.15.5, 2.3.9.15.7, 2.3.9.15.8, 3.9.2

### Ausblick

In Zukunft soll auch die Angabe einer Haftungsbeschränkung der Zertifizierungsstelle im ausgestellten Signaturschlüssel-Zertifikat ermöglicht werden (eine entsprechende Angabe wird beispielsweise für die Erzeugung qualifizierter Zertifikate nach der EU-Direktive erforderlich sein). In dieser Version von ISIS wird jedoch noch keine entsprechende Erweiterung definiert.

## Z 2. SIGNATURSCHLÜSSEL-ZERTIFIKATE

Als generelles Format für Zertifikate bietet sich das X.509v3 Zertifikatsformat an. In diesem Format wurden gegenüber dem X.509v1-Zertifikat die neuen optionalen Informationsfelder *issuerUniqueIdentifier* (X.509v2), *subjectUniqueIdentifier* (X.509v2) und *extensions* (X.509v3) hinzugefügt.

Zur Berechnung der Signatur eines Zertifikates werden Zertifikate nach den Vorschriften von ASN.1-DER [ITU-T X.690 94 | ISO/IEC 8825-1 94] kodiert. Die Abkürzung ASN.1 (abstract syntax notation one) bezeichnet eine genormte abstrakte Notation zur Beschreibung von Datentypen und Datenwerten. DER (distinguished encoding rules) ist eine spezielle Kodierungsvariante von ASN.1, die eine Einschränkung von deren Transfersyntax ermöglicht, die man zu einer eindeutigen Dekodierung empfangener Daten benötigt. An dieser Stelle sei darauf hingewiesen, daß im folgenden anstelle der in X.509v3 benutzten ASN.1-Syntax die von PKIX benutzte und implementierungsnähere ASN.1-Syntax [CCITT X.208 88] verwendet wird. Beide Syntaxformen sind hinsichtlich der Kodierung von Zertifikaten vollkommen äquivalent und produzieren die gleiche Transfersyntax.

Zertifikate werden durch den ASN.1-Typ *Certificate* als eine Folge von drei Feldern definiert, die zur Trennung der zu signierenden Daten *tbsCertificate*, des benutzten Signaturalgorithmus *signatureAlgorithm* und der eigentlichen Signatur *signature* dienen.

### ASN.1-Definitionen

```
Certificate ::= SEQUENCE {  
    tbsCertificate      TBSCertificate,  
    signatureAlgorithm  AlgorithmIdentifier,  
    signature           BIT STRING }
```

In den folgenden Abschnitten werden die einzelnen Zertifikatsfelder sowie deren unterstrukturierte Teilfelder beschrieben. Jeder Abschnitt ist dabei durch die Punkte “Zweck”, “ASN.1-Definitionen”, “Statische Semantik”, “Allgemeine Konformitätsanforderungen” und “ISIS-Konformitätsanforderungen” untergliedert, die folgende Beutung haben:

### Zweck

Unter diesem Unterpunkt wird die Bedeutung des betreffenden Zertifikatfeldes beschrieben.

### ASN.1-Definitionen

Dieser Unterpunkt enthält die ASN.1-Definitionen des Zertifikatfeldes gemäß der X.509v3-Empfehlung. Für ISIS-spezifische Zertifikatsfelder, die Objektbezeichner, private Erweiterungen oder Attribute betreffen, werden eigene ASN.1-Definitionen angegeben.

### Statische Semantik

Dieser optionale Unterpunkt enthält Einschränkungen von ASN.1-Definitionen, die nicht unmittelbar durch ASN.1 selbst ausgedrückt werden können.

Der Unterpunkt “Statische Semantik” ist integraler Bestandteil der ASN.1-Definitionen.

### Allgemeine Konformitätsanforderungen

An dieser Stelle wird eine Zusammenstellung von wesentlichen internationalen Konformitätsanforderungen gegeben. Konformitätsanforderungen sind Aussagen in Normen oder Empfehlungen, die festlegen, was in einem bestimmten Kontext zu tun ist, was getan werden darf oder was nicht getan werden darf. Aus Gründen der Interoperabilität sind deshalb von der vorliegenden Signatur-Interoperabilitätsspezifikation auch internationale oder nationale Festlegungen, wie sie beispielsweise in [RFC 2459 99], [DIN SigG/V 98] oder [MTRUST 96] getroffen wurden, zu beachten.

### ISIS-Konformitätsanforderungen

Dieser Unterabschnitt enthält Informationen über Einschränkungen und Anwendung der allgemeinen Konformitätsanforderungen hinsichtlich der durch X.509v3 möglichen Optionen. Im Rahmen der Interoperabilitätsspezifikation ISIS werden insbesondere weitere, durch die Normen und Empfehlungen zugelassene Strukturelemente wie spezielle Objektbezeichner, private Erweiterungen oder Attribute festgelegt oder die Benutzung bestimmter Elemente wie z.B. name *constraints* verboten.

Desweiteren enthält dieser Unterpunkt implementations-technische Informationen über einzelne Zertifikatsfelder und deren Unterstrukturen in einer tabellarischen Übersicht (Muster siehe Tabelle 2). Die erste Spalte enthält den ASN.1-Bezeichner des betreffenden Zertifikatsfeldes. Falls ein Zertifikatsfeld aus einer zusammengesetzten Struktur besteht, so werden auch die Bezeichner der Teilfelder aufgeführt. In der zweiten Spalte werden der zugelassene Wertebereich bzw. die zugelassenen Einzelwerte der Zertifikatsfelder dargestellt. Die dritte Spalte zeigt den Hexadezimalcode des Zertifikatsfeldes. Die vierte Spalte enthält die aktuelle Länge der Beispielfelder. Aus diesen Werten wird eine maximale Länge abgeleitet, die als Empfehlung für mindestens zu unterstützende Obergrenzen vorgegeben und in der Spalte durch Graudruck hervorgehoben wird. In den Spalten 5 bis 8 wird der Typ des Zertifikates angegeben, d.h. ob es sich um ein Zertifikat für Zertifizierungsstellen, den Zeitstempeldienst, den Verzeichnisdienst oder für Teilnehmer handelt. In den Spalten 9 bis 11 wird die Bedeutung eines Feldes entweder als obligatorisch, verboten oder optional gekennzeichnet. Die Spalten 12 bis 15 dienen zur Klassifikation von bestimmten Zertifikatsfeldern, die als Erweiterungen bezeichnet werden.

Zertifikatsformate sind nach der abstrakten ASN.1-Syntax definiert [ITU-T X.681 94] und konkrete Zertifikate werden nach den ASN.1-Transfersyntaxregeln [ITU-T X.690 94] kodiert, deren Kenntnis vorausgesetzt wird. Datentypen und Datenwerte werden nach ASN.1 durch das rekursive Schema “Typ-Länge-Wert” kodiert. Die Typ-Komponente (auch als sog. Tag-Feld bezeichnet) spezifiziert hierbei den Typ einer Zertifikatsstruktur, die Längenkomponente enthält die Länge des folgenden Zertifikatsfeldes in Bytes und die Wert-Komponente enthält das eigentliche Nutzdatenfeld, das seinerseits aus Unterstrukturen gemäß des Schemas “Typ-Länge-Wert” aufgebaut sein kann. Der Wertebereich der Wert-Komponente ist durch das Tag-Feld bestimmt. Prinzipiell besitzt nur die Typ-Komponente eine feste Kodierung und die beiden anderen Komponenten haben eine variable Länge. Aus diesem Grund haben die zweite

und die dritte Spalte der beschriebenen Tabelle überwiegend nur Beispielcharakter und dienen zur Illustration der Kodierung. Ebenso soll die in der vierten Spalte angegebene Längenangabe nur als minimale Länge verstanden werden, die ein System oder eine Anwendung unterstützen soll. In den angegebenen Beispielen sind die Tagfelder durch Fettschrift, die Längen durch Normalschrift und die Nutzdatenfelder durch Kursivschrift hervorgehoben.

**Tabelle 2: Implementations-technische Informationen**

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP						RELEVANZ				KLASSIFIKATION		
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung		
	(BEISPIELE)	(BEISPIELE)	[BYTES]													
		<b>Tag</b> Länge Wert														

## Z 2.1. Signaturalgorithmus

### Zweck

Das Signaturfeld *signatureAlgorithm* vom Typ *AlgorithmIdentifizier* enthält den Bezeichner des kryptographischen Algorithmus, der von der Zertifizierungsstelle zum Signieren des Zertifikates benutzt wird. Hierbei ist zu beachten, daß Signaturalgorithmen immer in Kombination mit Einweg-Hash-Funktionen und digitalen Signaturformaten (message formatting, padding) benutzt werden. Das Signaturfeld besteht syntaktisch aus einer Folge von Teilfeldern *algorithm* und *parameters*. Das Teilfeld *algorithm* ist ein Objektbezeichner, der zur Identifikation des Algorithmus dient. Der Inhalt des optionalen *parameters*-Teilfeldes ist abhängig vom angegebenen Algorithmus und dem Algorithmusbezeichner.

### ASN.1-Definitionen

```

Certificate ::= SEQUENCE {
    ...,
    signatureAlgorithm AlgorithmIdentifizier,
    ... }

AlgorithmIdentifizier ::= SEQUENCE {
    algorithm OBJECT IDENTIFIER,
    parameters ANY DEFINED BY algorithm OPTIONAL }

```

### Allgemeine Konformitätsanforderungen

Das Signaturfeld *signatureAlgorithm* muß denselben Algorithmusbezeichner wie das *signature*-Teilfeld der *tbsCertificate*-Struktur enthalten.

### ISIS-Konformitätsanforderungen

Das optionale *parameters*-Teilfeld darf nicht zur Übergabe von Parametern an den Algorithmus benutzt werden, da dieses Feld nicht durch die Signatur der Zertifizierungsstelle geschützt ist. Auch der innere Algorithmusbezeichner darf nicht mit Parametern versehen werden und dessen Komponente *parameters* ist mit dem Wert NULL zu belegen. Die Übergabe der Parameter erfolgt zusammen mit dem öffentlichen Schlüssel. Die maximale Länge des *signatureAlgorithm*-Feldes beträgt 20 Bytes.

Zum Signieren geeignete und zugelassene Algorithmen werden von der Regulierungsbehörde für Telekommunikation und Post im Bundesanzeiger veröffentlicht. Die in der Ausgabe vom 14. Februar 1998 aufgeführten und geeigneten Kryptoalgorithmen gelten für die kommenden sechs Jahre (14. Februar 1998 – 14. Februar 2004). Die Algorithmen und Parameter, mit denen eine Zertifizierungsstelle ein Zertifikat signiert, müssen mindestens für die Gültigkeitsdauer des Zertifikats als geeignet beurteilt sein.

ISIS-konforme Implementationen sollten RSA, DSA, SHA-1 und RIPEMD-160 unterstützen.

### Beispiele für Algorithmusbezeichner

```
rsaSignatureWithsha1 OBJECT IDENTIFIER ::= { 1 3 36 3 3 1 1 }
rsaSignatureWithripemd160 OBJECT IDENTIFIER ::= { 1 3 36 3 3 1 2 }
ecdsa-with-sha1 OBJECT IDENTIFIER ::= { 1 2 840 10045 1 }
```

Aus Gründen der Interoperabilität mit internationaler Software wird für RSA mit SHA-1 die Verwendung des Object Identifiers sha1withRSAEncryption (1 2 840 113549 1 1 5) aus [PKCS1 98] empfohlen.

Tabelle 3: Implementations-technische Informationen über *signatureAlgorithm*

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	ZERTIFI- KATSTYP						RELE- VANZ
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	
	(BEISPIELE)	(BEISPIELE)	20							
Signature- Algorithm	rsaSignatureWithsha1		11							
Algorithm	SEQUENCE {	30 0A								
Parameters	{1 3 36 3 3 1 1 }, NULL }	06 06 2B 24 03 03 01 05 00								
Signature- Algorithm	rsaSign.Withripemd160		11							
Algorithm	SEQUENCE {	30 0A								
Parameters	{1 3 36 3 3 1 2 }, NULL }	06 06 2B 24 03 03 01 02 05 00								
Signature- Algorithm	ecdsa-with-sha1		12							
Algorithm	SEQUENCE {	30 0A								
Parameters	{1 2 840 10045 1 }, NULL }	06 06 2A 86 48 CE 3C 01 05 00								

## Z 2.2. Signatur eines Zertifikates

### Zweck

Das Signaturfeld *signature* enthält eine digitale Signatur, die für das in ASN.1-DER kodierte Zertifikatsfeld *tbsCertificate* berechnet wird. Bei der Berechnung der Signatur wird das Zertifikatsfeld *tbsCertificate* als Eingabe in eine Einweg-Hash-Funktion benutzt. Auf den Ergebniswert der Hashfunktion wird der private Schlüssel der Zertifizierungsstelle angewandt und als ASN.1-Bitstring kodiert. Er liefert die konkrete digitale Signatur des Zertifikates im Signaturfeld *signature*. Durch den Signaturvorgang beglaubigt eine Zertifizierungsstelle die Gültigkeit der im Zertifikatsfeld *tbsCertificate* enthaltenen Informationen und gewährleistet insbesondere die Bindung zwischen dem öffentlichen Schlüssel und dem Zertifikatsinhaber.

### ASN.1-Definition

```
Certificate ::= SEQUENCE {
    ...
    signature BIT STRING }
```

Allgemeine Konformitätsanforderungen

Geeignete Signaturformate finden sich in den Spezifikationen [PKCS1 93] (Abschnitt 8.1) und [DIN SigG/V 98 (Anhang A)]. Üblicherweise wird das Ergebnis der Einweg-Hash-Funktion an die Signaturkomponente übergeben. Die Komponente ergänzt gegebenenfalls den ihr übergebenen Hashwert um zusätzliche Komponenten, bevor die eigentliche mathematische Signaturfunktion angewendet wird (siehe Teilspezifikation "A2 Signatur" von SigI).

ISIS-Konformitätsanforderungen

Bei der Erstellung digitaler Signaturen dürfen nur die in der Teilspezifikation "A2 Signatur" [A2 99] aufgeführten Signaturalgorithmen und Signaturformate benutzt werden.

**Tabelle 4: Implementations-technische Informationen über *signature***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	ZERTIFIKATSTYP						RELEVANZ
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	
	(BEISPIELE)	(BEISPIELE)	261							
signature	256 Byte-Schlüssellänge BITSTRING	03 82 01 01 ...	261							

**Z 2.3. Zu signierende Zertifikatsinformationen**Zweck

Die Zertifikatsinformationen werden durch den *TBSCertificate*-Typ repräsentiert, der seinerseits aus einer Folge von weiteren Teilfeldern besteht und in seiner Gesamtheit von einer Zertifizierungsstelle zu signieren ist.

ASN.1-Definitionen

```
Certificate ::= SEQUENCE {
    tbsCertificate TBSCertificate,
    ... }
```

```
TBSCertificate ::= SEQUENCE {
    version [0] EXPLICIT Version DEFAULT v1,
    serialNumber CertificateSerialNumber,
    signature AlgorithmIdentifier,
    issuer Name,
    validity Validity,
    subject Name,
    subjectPublicKeyInfo SubjectPublicKeyInfo,
    issuerUniqueID [1] IMPLICIT UniqueIdentifier OPTIONAL,
```

```

subjectUniqueID      [2]  IMPLICIT UniqueIdentifier OPTIONAL,
extensions           [3]  EXPLICIT Extensions Optional }

```

Das Zertifikatsfeld *tbsCertificate* besteht aus einer Folge von Informationen, die in unmittelbarem Zusammenhang mit dem Inhaber eines Zertifikates und der Zertifizierungsstelle stehen, die dieses Zertifikat ausgestellt hat. Jedes Zertifikat enthält die technischen Namen *subject* des Inhabers und *issuer* des Erstellers, den öffentlichen Schlüssel des Inhabers *subjectPublicKeyInfo*, den Gültigkeitszeitraum des Zertifikates *validity*, sowie die Versionsnummer *version* und die Seriennummer des Zertifikates *serialNumber*. Darüberhinaus können Zertifikate optionale Felder für Namensbezeichner *issuerUniqueID* und *subjectUniqueID* und Zertifikatserweiterungen *extensions* enthalten.

#### Allgemeine Konformitätsanforderungen

Technische Komponenten müssen die Felder *subjectUniqueID* und *issuerUniqueID* nicht unterstützen, sollen aber Zertifikate mit diesen Feldern zurückweisen, falls sie diese nicht verarbeiten können.

#### ISIS-Konformitätsanforderungen

Bei der Erzeugung von Zertifikaten ist die Verwendung der Felder *subjectUniqueID* und *issuerUniqueID* aus Gründen der internationalen Interoperabilität verboten. Außerdem erfolgt die Identifikation von Zertifikatsinhabern und Zertifikatsersteller über die Komponenten *subject*, *subjectAltName*, *issuer* und *issuerAltName*.

**Tabelle 5: Implementations-technische Informationen über *tbsCertificate***

FELD	ZERTIFIKATSTYP			RELEVANZ			FELD	ZERTIFIKATSTYP			RELEVANZ			
	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten		optional	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten
version							subject							
serialNumber							subjectPublicKeyInfo							
signature							issuerUniqueID							
issuer							subjectUniqueID							
validity							extensions							

## Z 2.3.1. VERSIONSNUMMER

### Zweck

Das Versionsfeld gibt die Version eines X.509-Zertifikates an. Die Voreinstellung für dieses Feld hat den Wert 0, der ein Zertifikat der Version 1 anzeigt. Hierfür hat sich auch die äquivalente Schreibweise "X.509v1-Zertifikat" eingebürgert.

### ASN.1-Definitionen

```
TBSCertificate ::= SEQUENCE {
  version          [0] EXPLICIT Version DEFAULT v1,
  ... }
```

```
Version ::= INTEGER { v1(0), v2(1), v3(2) }
```

### Allgemeine Konformitätsanforderungen

Zertifikate, die optionale Erweiterungen *extensions* enthalten, müssen die Version 3 verwenden. Zertifikate, die keine optionale Erweiterungen, dafür aber optionale Namensbezeichner enthalten, sollen die Version 2 verwenden. Zertifikate, die nur vorgeschriebene Felder und keine optionalen Felder enthalten, sollen die Version 1 verwenden, deren Wert aber nicht im Zertifikat zu kodieren ist, da er durch die Voreinstellung mit Hilfe des DEFAULT-Konstruktes bereits spezifiziert ist.

Implementationen sollten in der Lage sein, jede Zertifikatversion zu akzeptieren. An konforme Implementationen besteht die Minimalanforderung, daß sie Zertifikate der Version 3 erkennen können.

### ISIS-Konformitätsanforderungen

Bei der Erzeugung von Zertifikaten ist die Verwendung der Version X.509v3 obligatorisch. Die Erstellung von Zertifikaten der Version X.509v1 oder X.509v2 wird von diesem Profil nicht unterstützt.

**Tabelle 6: Implementations-technische Informationen über *version***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	ZERTIFIKATSTYP				RELEVANZ			
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer		obligatorisch	verboten	optional
	(BEISPIELE)	(BEISPIELE)	3								
version	2	02 01 02	3								

## Z 2.3.2. SERIENNUMMER

### Zweck

Die Seriennummer ist eine positive ganze Zahl, die von der Zertifizierungsstelle jedem Zertifikat zugewiesen wird und die dieses dadurch innerhalb der Zertifizierungsstelle eindeutig identifiziert. Zertifikate werden durch die Kombination aus der Seriennummer *serialNumber* und dem Namen der Zertifizierungsstelle *issuer* global eindeutig identifiziert. Diese Kombination wird beispielsweise auch zur eindeutigen Referenzierung von Zertifikaten innerhalb von Sperrlisten verwandt.

### ASN.1-Definitionen

```
TBSCertificate ::= SEQUENCE {
    ... ,
    serialNumber      CertificateSerialNumber,
    ... }
```

```
CertificateSerialNumber ::= INTEGER
```

### Allgemeine Konformitätsanforderungen

Die Kodierung der Seriennummer wird durch den ASN.1-Typ INTEGER festgelegt und unterliegt keiner expliziten Längenbegrenzung.

### ISIS-Konformitätsanforderungen

Bei der Erstellung von Zertifikaten ist die Seriennummer als eine 1- bis 15-Byte große ganze Zahl obligatorisch.

**Tabelle 7: Implementations-technische Informationen über *serialNumber***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	ZERTIFI- KATSTYP				RELE- VANZ	
				Zertifizierungsstellen	Zeitmeldedienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten
	(BEISPIELE)	(BEISPIELE)	17						
serialNumber	0	02 01 00	3						
	1	02 01 01	3						
	...	02 ... ..	3-17						
	$2^{8 \cdot 15 - 1} - 1$	02 0F 7F FF FF FF FF FF FF FF FF FF FF FF FF FF FF	17						

## Z 2.3.3. SIGNATUR

Zweck

Das Signaturfeld enthält den Bezeichner des Algorithmus, der von der Zertifizierungsstelle zum Signieren des Zertifikates benutzt wird. Zum Signieren geeignete Algorithmen werden von der Regulierungsbehörde für Telekommunikation und Post im Bundesanzeiger veröffentlicht. Die für die kommenden sechs Jahre (14. Februar 1998 – 14. Februar 2004) als geeignet beurteilten Kryptoalgorithmen sind in der Teilspezifikation "A2 Signatur" von SigI [A2 99] beschrieben.

ASN.1-Definitionen

```

TBSCertificate ::= SEQUENCE {
    ...,
    signature      AlgorithmIdentifier,
    ... }

AlgorithmIdentifier ::= SEQUENCE {
    algorithm      OBJECT IDENTIFIER,
    parameters    ANY DEFINED BY algorithm OPTIONAL }

```

Allgemeine Konformitätsanforderungen

Das Signaturfeld *signature* der *tbsCertificate*-Struktur muß denselben Algorithmusbezeichner wie im *signatureAlgorithm*-Feld der *Certificate*-Struktur enthalten.

ISIS-Konformitätsanforderungen

Das Signaturfeld *signature* der *tbsCertificate*-Struktur muß denselben Algorithmusbezeichner wie im *signatureAlgorithm*-Feld der *Certificate*-Struktur enthalten.

**Tabelle 8: Implementations-technische Informationen über *signature***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	ZERTIFI- KATSTYP	RELE- VANZ
	(BEISPIELE)	(BEISPIELE)	20	Zertifizierungsstellen Zeitstempeldienst Verzeichnisdienst Teilnehmer obligatorisch verboten optional	
Signature- Algorithm Algorithm Parameters	rsaSignatureWithsha1 SEQUENCE { {1 3 36 3 3 1 1 }, NULL }	30 0A 06 06 2B 24 03 03 01 05 00	11		
Signature- Algorithm Algorithm Parameters	RsaSig.Withripemd160 SEQUENCE { {1 3 36 3 3 1 2 }, NULL }	30 0A 06 06 2B 24 03 03 01 02 05 00	11		
Signature- Algorithm Algorithm Parameters	ecdsa-with-sha1 SEQUENCE { {1 2 840 10045 1 }, NULL }	30 0A 06 06 2A 86 48 CE 3C 01 05 00	12		

## Z 2.3.4. TECHNISCHE NAMEN VON ZERTIFIZIERUNGSSTELLEN

### Zweck

Das *issuer*-Namensfeld dient zur technischen Identifikation der Instanz bzw. Zertifizierungsstelle, die das betreffende Zertifikat erstellt und signiert hat.

Es sind bei der technischen Namensgebung von Zertifizierungsstellen nur Namen gemäß der X.500-Syntax [ITU-T X.500 97] für *distinguished name*-Typen zugelassen. Der *distinguished name* ist vom Typ *RDNSequence* und somit aus einer Folge von *AttributeType*- und *AttributeValue*-Paaren zusammengesetzt. *AttributeType* wird i.a. durch X.500 festgelegt, und für *AttributeValue* wird der Typ *DirectoryString* (für den unspezifischen Typ *ANY*) verwendet, der seinerseits ein Auswahltyp von *PrintableString*, *TeletexString*, *UniversalString* und *BMPString* ist. Eine Übersicht der möglichen Objektbezeichner für *AttributeType* ist in der folgenden Tabelle gegeben.

### ASN.1-Definitionen

```

TBSCertificate ::= SEQUENCE {
    ...,
    issuer          Name,
    ... }

Name ::= CHOICE { RDNSequence }

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName

RelativeDistinguishedName ::= SET OF AttributeTypeAndValue

AttributeTypeAndValue ::= SEQUENCE {
    type      AttributeType,
    value     AttributeValue }

AttributeType ::= OBJECT IDENTIFIER

AttributeValue ::= ANY DEFINED BY AttributeType

DirectoryString ::= CHOICE {
    printableString      PrintableString (SIZE (1..maxSize))
    teletexString        TeletexString (SIZE (1..maxSize))
    bmpString            BMPString (SIZE (1..maxSize))
    universalString      UniversalString (SIZE (1..maxSize)) }

```

### Allgemeine Konformitätsanforderungen

X.509v3-konforme Zertifizierungsstellen sollen bei der Konstruktion von *DirectoryString* stets die restriktivste Auswahl treffen und deshalb den minimalen Zeichensatz zur Repräsentation von *AttributeValue* wählen. Die Reihenfolge, in der die Zeichensätze auf ihre konkrete Anwendbarkeit hin geprüft werden sollen, lautet somit: *PrintableString*, *TeletexString*, *BMPString* und *UniversalString*.

Der Name einer Zertifizierungsstelle kann nach [ITU-T X.509 97] auch alternativ oder zusätzlich zum *issuer*-Feld im optionalen *extensions*-Feld unter *issuerAltName* (siehe Abschnitt Z 2.3.9.6) angegeben werden. Im ersten Fall kann das *issuer*-Feld als leere Folge kodiert werden und die *issuerAltName*-Erweiterung muß als *critical*, d.h. als "wichtige und zu berücksichtigende" Erweiterung gekennzeichnet werden.

ISIS-Konformitätsanforderungen

Bei der Erstellung von Zertifikaten ist die Benutzung des *issuer*-Feldes obligatorisch. Diese Anforderung ergibt sich aus der Notwendigkeit einer eindeutigen technischen Benennung der Zertifizierungsstelle. Das *issuer*-Feld soll mit dem technischen Namen der Zertifizierungsstelle belegt werden, damit die Konformität zu vielen Anwendungen im internationalen Kontext gewährleistet bleibt. Namen von Zertifizierungsstellen enthalten zumindest die obligatorischen Attribute *organization* und *countryName*. Alle anderen Attribute sind optional. Weitere Informationen über Namenskonventionen sind im Anhang I zu finden.

Beispiele für technische Namen von Zertifizierungsstellen

## (1) Technischer Name der RegTP

OU=Wurzelzertifizierungsstelle, O=RegTP, C=DE

## (2) Technischer Name der RegTP mit Acronym

CN=DEPCA, OU=Wurzelzertifizierungsstelle, O= RegTP, C=DE

## (3) Technischer Name einer untergeordneten Zertifizierungsstelle

CN=Name-der-ZS, O=Organisation-der-ZS, C=DE

Die Länge der *AttributeValue*-Stringtypen ist durch den Systemparameter *maxSize* festgelegt, dessen Wert für die einzelnen Attribute gemäß der folgenden Tabelle begrenzt ist. Hieraus ergeben sich die in der Längenspalte angegebenen maximalen Längen der Attribute inklusive der ASN.1-Kontrollinformationen, die eine Länge von 11 Bytes haben.

**Tabelle 9: Implementations-technische Informationen über Längen von Attributtypen**

OBJEKTBEZEICHNER		MAXSIZE	LÄNGE	OBJEKTBEZEICHNER		MAXSIZE	LÄNGE
NAME	NUMMER	[BYTES]	[BYTES]	NAME	NUMMER	[BYTES]	[BYTES]
commonName	{ 2 5 4 3 }	64	75	organizationName	{ 2 5 4 10 }	64	75
surName	{ 2 5 4 4 }	32	43	organizationalUnit	{ 2 5 4 11 }	64	75
serialNumber	{ 2 5 4 5 }	64	75	title	{ 2 5 4 12 }	10	21
countryName	{ 2 5 4 6 }	2	13	businessCategory	{ 2 5 4 15 }	32	43
localityName	{ 2 5 4 7 }	32	43	postalCode	{ 2 5 4 17 }	10	21
stateOrProvince	{ 2 5 4 8 }	32	43	givenName	{ 2 5 4 47 }	32	43

**Tabelle 10: Implementations-technische Informationen über *issuer***

BEZEICHNER	WERTEBEREICH EINZELWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	ZERTIFI- KATSTYP						RELE- VANZ
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	
	(BEISPIELE)	(BEISPIELE)	513							
issuer	SEQUENCE OF { SET OF SEQUENCE { {	30 53 31 0B 30 09	85 13							
countryName value	{ 2 5 4 6 }, "DE" } }	06 03 55 04 06 13 02 44 45								
organizationName value	SET OF SEQUENCE { { { 2 5 4 10 }, "regtp" } }	31 0E 30 0C 06 03 55 04 0A 13 05 72 65 67 74 70	16							
organizationalUnit value	SET OF SEQUENCE { { { 2 5 4 11 }, "Wurzelzertifizierungss- stelle" } }	31 24 30 22 06 03 55 04 0B 13 1B 57 75 72 7A 65 6C 7A 65 72 74 69 ...	38							
commonName value	SET OF SEQUENCE { { { 2 5 4 3 }, "DEPCA" } } }	31 0E 30 0C 06 03 55 04 03 13 05 44 45 50 43 41	16							

### Z 2.3.5. GÜLTIGKEITSDAUER

#### Zweck

Öffentliche Zertifikate besitzen wegen ihrer Verteil- und Kopierbarkeit prinzipiell eine beliebig große Lebensdauer. Die praktische Benutzbarkeit von Signaturschlüsseln wird jedoch von Zertifizierungsstellen bei der Erstellung von Zertifikaten durch eine Gültigkeitsdauer zeitlich begrenzt. Zertifizierungsstellen müssen jedoch über den Gültigkeitszeitraum hinaus Zustandsinformationen und weitere Informationen über das Zertifikat pflegen und anbieten. Die Gültigkeitsdauer eines Zertifikates ist ein Zeitintervall, das durch zwei Zeitpunkte definiert ist, die den Beginn und das Ende der Gültigkeit eines Zertifikates anzeigen, innerhalb dessen der Zertifikatsinhaber das Zertifikat zur Erzeugung von Signaturen verwenden darf.

Die Gültigkeitsdauer eines Zertifikates wird im Feld *validity* durch die zwei Zeitpunkte *notBefore* und *notAfter* angegeben. Beide Zeitpunkte können durch die Standard-ASN.1-Zeit-typen *UTCTime* (coordinated universal time, Weltzeit) oder *GeneralizedTime* (allgemeines Datums- und Zeitformat) repräsentiert werden, die Datums- und Zeitangaben bis auf Sekundengenauigkeit sowie die Angabe von Zeitverschiebungen der lokalen gegenüber der Weltzeit

gestatten. Die Hauptunterschiede zwischen beiden Formaten bestehen darin, daß mit dem verallgemeinerten Zeittyp kleinere Zeiteinheiten und vollständige Jahreszahlen angegeben werden können.

### ASN.1-Definitionen

```

TBSCertificate ::= SEQUENCE {
    .../
    validity      Validity,
    ... }

Validity ::= SEQUENCE {
    notBefore     Time,
    notAfter      Time }

Time ::= CHOICE {
    utcTime       UTCTime,
    generalizedTime GeneralizedTime }

```

### Allgemeine Konformitätsanforderungen

X.509v3-konforme Zertifizierungsstellen sollen zur Kodierung der Gültigkeitszeitpunkte bis zum Jahr 2049 als Zeittyp stets den Typ *UTCTime* und ab dem Jahr 2050 den Typ *GeneralizedTime* benutzen. Zertifizierungsstellen sollen bei der Verwendung eines dieser Typen die Werte von Zeitpunkten in Greenwich-Zeit (GMT, Greenwich Mean Time) bis auf Sekundengenauigkeit ausdrücken, wobei auch die Null-Sekunde zu kodieren ist. Bei der Kodierung der Datums- und Zeitangaben sind für *GeneralizedTime* das Format YYYYMMDDHHMMSSZ und für *UTCTime* das Format YYMMDDHHMMSSZ zu beachten. Die Bedeutung der einzelnen Felder der Datums- und Zeitformate ist in der folgenden Tabelle zusammengefaßt.

**Tabelle 11: Bedeutung der Felder in Datums- und Zeitformaten**

DATUMSANGABEN		ZEITANGABEN	
FELD	BEDEUTUNG	FELD	BEDEUTUNG
YYYY	vollständige Jahreszahl, nur bei <i>GeneralizedTime</i>	HH	Stunde 00, 01, ..., 23
YY	letzte zwei Ziffern der Jahreszahl, nur bei <i>UTCTime</i>	MM	Minute 00, 01, ..., 59
MM	Monat 01, 02, ..., 12	SS	Sekunde 00, 01, ..., 59
DD	Tag 01, 02, ..., 31	Z	GMT

X.509v3-konforme Systeme sollen bei der Benutzung des *UTCTime* Typs das 2-stellige Jahresfeld YY gemäß der folgenden Konvention (links) interpretieren.

Für das 2-stellige Jahresfeld YY gilt nach [MTRUST 96] die folgende Konvention (rechts), die jedoch nicht kompatibel zu [ITU-T X.509 97], [PKIX PRO 97] und [MISPC 97] ist.

$$\text{Jahr}(YY) = \begin{cases} 19YY & | YY \in [50,99] \\ 20YY & | YY \in [0,49] \end{cases}
 \qquad
 \text{Jahr}(YY) = \begin{cases} 19YY & | YY \in [65,99] \\ 20YY & | YY \in [0,64] \end{cases}$$

Die Inkompatibilität betrifft die Zeiträume zwischen 1950 und 1964, sowie zwischen 2050 und 2064. Der erste Zeitraum (1950 bis 1964) bereitet keine Probleme, da es hierfür noch keine Zertifikate gib. Zertifikate, deren Gültigkeitsdauern in den zweiten Jahreszeitraum (2050 bis 2064) fallen, sollten zur Kodierung ebenfalls im *GeneralizedTime*-Format erstellt werden.

### ISIS-Konformitätsanforderungen

Bei der Erzeugung von Zertifikaten wird für Zeitangaben bis zum Jahr 2049 das Format *UTCTime* in der Form YYMMDDHHMMSSZ empfohlen. Die Interpretation der zweistelligen Jahreszahl erfolgt dabei gemäß X509v3. Für Zeitangaben ab 2050 ist das Format *GeneralizedTime* zu verwenden, bei dessen Kodierung das Format YYYYMMDDHHMMSSZ genommen werden soll. Anwendungen müssen beide Formate interpretieren können.

Das *notBefore*-Feld soll den Zeitpunkt der Erstellung des Zertifikats enthalten. Ist es notwendig, in *notBefore* einen anderen Zeitpunkt aufzunehmen, muß das Erstellungsdatum in der in Kapitel Z 2.3.9.15.3 definierten Erweiterung *dateOfCertGen* eingetragen sein.

**Tabelle 12: Implementations-technische Informationen über *validity***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	ZERTIFI- KATSTYP				RELE- VANZ	
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten
	(BEISPIELE)	(BEISPIELE)	36						
validity	SEQUENCE {	30 22	36						
notBefore	GeneralizedTime "19980101000000Z",	18 0F 31 39 39 38 30 31 30 31 30 30 30 30 30 30 5A	17						
notAfter	GeneralizedTime "20030101000000Z" }	18 0F 32 30 30 33 30 31 30 31 30 30 30 30 30 30 5A	17						

## Z 2.3.6. NAMEN VON ZERTIFIKATINHABERN

### Zweck

Das *subject*-Namensfeld dient zur Identifikation des Inhabers eines Zertifikates, für den das Zertifikat ausgestellt wurde. Der Typname *subject* hat denselben Typ wie das *issuer*-Feld und muß wie dieses gemäß der X.500-Syntax ein *distinguished name* sein.

Es sind bei der technischen Namensgebung von Zertifikatsinhabern nur Namen gemäß der X.500-Syntax [ITU-T X.500 97] für *distinguished name*-Typen zugelassen. Der *distinguished*

*name* ist vom Typ *RDNSequence* und somit aus einer Folge von *AttributeType*- und *AttributeValue*-Paaren zusammengesetzt. *AttributeType* wird i.a. durch X.500 festgelegt, und für *AttributeValue* wird der Typ *DirectoryString* (für den unspezifischen Typ *ANY*) verwendet, der seinerseits ein Auswahltyp von *PrintableString*, *TeletexString*, *UniversalString* und *BMPString* ist. Eine Übersicht möglicher Objektbezeichner für *AttributeType* ist in der Tabelle 9 im Abschnitt Z 2.3.4 gegeben.

Die PKIX Working Group sieht in ihrem Internet-Draft zum Thema qualifizierte Zertifikate [PKIX QC 99] die Verwendung des Attributes *pseudonym* vor, wenn statt des wirklichen Namens einer Person ein Pseudonym angegeben werden soll. Die ISIS-Spezifikation läßt die zum Attribut *commonName* zusätzliche Verwendung von *pseudonym* zu.

Objektbezeichner, die von PKIX festgelegt werden, sind unter dem Objektbezeichnerzweig *id-pkix* angeordnet. Hierunter liegt u.a. der Objektbezeichnerzweig *id-pda*, der für „personal data attributes“ in [PKIX QC 99] definiert ist.

#### ASN.1-Definitionen

```

TBSCertificate ::= SEQUENCE {
    ...,
    subject      Name,
    ... }

Name ::= CHOICE { RDNSequence }

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName

RelativeDistinguishedName ::= SET OF AttributeTypeAndValue

AttributeTypeAndValue ::= SEQUENCE {
    type      AttributeType,
    value     AttributeValue }

AttributeType ::= OBJECT IDENTIFIER

AttributeValue ::= ANY DEFINED BY AttributeType

DirectoryString ::= CHOICE {
    printableString      PrintableString (SIZE (1..maxSize))
    teletexString        TeletexString (SIZE (1..maxSize))
    bmpString            BMPString (SIZE (1..maxSize))
    universalString      UniversalString (SIZE (1..maxSize)) }

-- Definitionen fuer das Attribut pseudonym
id-pkix      OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 }
id-pda       OBJECT IDENTIFIER ::= { id-pkix 9 }
id-pda-pseudonym AttributeType ::= { id-pda 3 }
Pseudonym    ::= DirectoryString

```

### Allgemeine Konformitätsanforderungen

X.509v3-konforme Zertifizierungsstellen sollen bei der Konstruktion von *DirectoryString* stets die restriktivste Auswahl treffen und deshalb den minimalen Zeichensatz zur Repräsentation von *AttributeValue* wählen. Die Reihenfolge, in der die Zeichensätze auf ihre konkrete Anwendbarkeit hin geprüft werden sollen, lautet somit: *PrintableString*, *TeletexString*, *BMPString* und *UniversalString*.

Der technische Name eines Zertifikatinhabers kann nach [ITU-T X.509 97] auch alternativ oder zusätzlich zum *subject*-Feld im optionalen *extensions*-Feld unter *subjectAltName* (siehe Abschnitt Z 2.3.9.5) angegeben werden. Im ersten Fall kann das *subject*-Feld als leere Folge kodiert werden und die *subjectAltName*-Erweiterung muß als *critical*, d.h. als "wichtige und zu berücksichtigende" Erweiterung gekennzeichnet werden.

### ISIS-Konformitätsanforderungen

Der von der ITU-T [ITU-T X.509 97] zugelassene Spielraum bei der Namensgebung (Weglassen des *subject*-Feldes) ist einzuschränken und bei der Erstellung von Zertifikaten ist stets das *subject*-Feld zu benutzen. Diese Anforderung ergibt sich aus der Notwendigkeit einer eindeutigen Benennung des Zertifikatsinhabers. Weitere Namen eines Zertifikatsinhabers können danach nur zusätzlich zum *subject*-Feld im *extensions*-Feld unter *subjectAltName* (siehe Abschnitt Z 2.3.9.5) enthalten sein.

Im *subject*-Feld muß mindestens der gesetzliche Name oder das Pseudonym des Zertifikatinhabers angegeben werden, um eine Person eindeutig zu identifizieren. Möglicherweise ist für einige Angaben im Zertifikat gemäß Signaturgesetz [SigG, §7 Abs.3] die Einwilligung der Betroffenen erforderlich. Das *subject*-Feld muß mindestens die Attribute

- *commonName*,
- *serialNumber* (wenn die Person nicht durch *commonName* eindeutig bestimmt ist)
- und *countryName* enthalten.

Pseudonyme sind durch den Zusatz ":PN" am Ende des *commonName* zu kennzeichnen. Gleichzeitig kann das Pseudonym auch im Attribut *pseudonym* angegeben werden.

Durch die Werte in diesen Attributen muß die betreffende Person eindeutig bestimmt sein.

Für Zertifikate von Zertifizierungsstellen, Verzeichnisdiensten oder Zeitstempeldiensten ist ebenfalls das *subject*-Feld zu benutzen, hierbei sind zumindest die obligatorischen Attribute *organization* und *countryName* zu benutzen. Um technische Namen mehrfach vergeben zu können, muß das Attribut *serialNumber* dem Namen hinzugefügt werden.

Alle anderen Attribute sind optional. Weitere Informationen über Namenskonventionen sind im Anhang I zu finden.

**Tabelle 13: Implementations-technische Informationen über *subject***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	ZERTIFI- KATSTYP						RELE- VANZ
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	
	(BEISPIELE)	(BEISPIELE)	513							
subject	SEQUENCE OF {	30 63	101							
countryName value	SET OF { SEQUENCE { { 2 5 4 6 }, "DE" } }	31 0B 30 09 06 03 55 04 06 13 02 44 45	13							
organization Name value	SET OF { SEQUENCE { { 2 5 4 10 }, " KV Hessen" } }	31 12 30 10 06 03 55 04 0A 13 09 4B 56 20 48 65 73 73 65 6E	20							
organizational Unit value	SET OF { SEQUENCE { { 2 5 4 11 }, "rca" } }	31 0C 30 0A 06 03 55 04 0B 13 03 72 63 61	14							
serialNumber	SET OF { SEQUENCE { { 2 5 4 5 }, "1" } }	31 0A 30 08 06 03 55 04 05 13 01 31	12							1)
title	SET OF { SEQUENCE { { 2 5 4 12 }, "Dr." } },	31 0C 30 0A 06 03 55 04 0C 13 03 44 72 2E	14							
commonName	SET OF { SEQUENCE { { 2 5 4 3 }, "Name-des-Arztes" } } }	31 18 30 16 06 03 55 04 03 13 0F 4E 61 6D 65 2D 64 65 73 2D ...	26							
pseudonym	2)	2)								

1) Ist die Person nicht durch common name eindeutig bestimmt, muß serialNumber vorkommen.

2) Für pseudonym liegt kein Beispiel vor.

## BEISPIELE FÜR NAMEN VON ZERTIFIKATSINHABERN:

- (1) Technischer Name einer Zertifizierungsstelle:  
CN=Name-der-ZS,O=Organisation-der-ZS, C=DE
- (2) Technischer Name einer Person für Verwendung in privaten Geschäftsbeziehungen über die postalische Adresse:  
CN=Vorname Name, ST=Straße, L=Postleitzahl Ort, C=DE
- (3) Technischer Name einer Person als Mitarbeiter einer Firma über den Firmennamen und gegebenenfalls die Personalnummer:  
CN=Vorname Name, SER=Personalnummer, OU=Organisationseinheit, O=Firma, C=DE
- (4) Technischer Name einer Person als Mitglied der Kassenärztlichen Vereinigung über Mitgliedsnummer und Namen der Vereinigung:  
CN=Vorname Name, T=Titel, SER=Mitgliedsnummer, O=KV Hessen, C=DE
- (5) Grundsätzlich können bei allen Beispielen noch weitere Attribute verwendet werden. Zur Unterscheidung von Vor- und Nachnamen kann beispielsweise zusätzlich das Attribut "S=Nachname" verwendet werden:  
CN=Vorname Name, S=Name, ST=Straße, L=Postleitzahl Ort, C=DE
- (6) Pseudonym als technischer Name einer Person:  
CN=Pseudonym, C=DE

### Z 2.3.7. ÖFFENTLICHE SCHLÜSSEL VON ZERTIFIKATSINHABERN

#### Zweck

Das *subjectPublicKeyInfo*-Feld enthält im Teilfeld *subjectPublicKey* den durch das Zertifikat zertifizierten öffentlichen Schlüssel des Zertifikatsinhabers. Das Teilfeld *algorithm* gibt an, mit welchem kryptographischen Algorithmus der Schlüssel zu verwenden ist.

#### ASN.1-Definitionen

```
TBSCertificate ::= SEQUENCE {
    ...,
    subjectPublicKeyInfo SubjectPublicKeyInfo,
    ... }

SubjectPublicKeyInfo ::= SEQUENCE {
    algorithm AlgorithmIdentifier,
    subjectPublicKey BIT STRING }

AlgorithmIdentifier ::= SEQUENCE {
    algorithm OBJECT IDENTIFIER,
    parameters ANY DEFINED BY algorithm OPTIONAL }
```

#### RSA-Schlüsselalgorithmen

Für RSA-Schlüsselalgorithmen besteht der öffentliche Schlüssel *subjectPublicKey* aus einer Folge von Integerwerten für den Modulus und den Exponenten, und für das Teilfeld *algorithm* sind die Varianten *rsa*, *rsaEncryption* und *rsaSignature* definiert.

Die rsa-Variante ist in [X.509] mit dem Objektbezeichner *rsa* und dem Parameter *KeySize* vom Typ INTEGER definiert, der die Länge des öffentlichen RSA-Schlüsselmodulus angibt. Für diese Variante sind keine Auffüllregeln (padding) festgelegt.

Die rsaEncryption-Variante ist in [PKCS1 93] mit dem Objektbezeichner *rsaEncryption* und leerem Parameterfeld definiert. Für diese Variante existieren zwei Typen von Block-auffüllregeln. In der [MTRUST 96]-Spezifikation wird nur der Blocktyp 1 zugelassen, der zusätzlichen Schutz gegen verschiedene Angriffsarten bieten.

Die rsaSignature-Variante ist in [ANS X9.31] mit dem Objektbezeichner *rsaSignature* und leerem Parameterfeld definiert. Dieser Algorithmus verwendet zusätzliche Redundanz bei der Konstruktion des Signaturblockes und verhindert dadurch, daß er sich als natürliche Potenz darstellen läßt, wodurch mögliche Verfälschungen der Signatur verhindert werden.

#### ASN.1-Definitionen

```

RSAPublicKey ::= SEQUENCE {
    modulus          INTEGER,
    publicExponent  INTEGER }
encryptionAlgorithm OBJECT IDENTIFIER ::= { 2 5 8 1 }
rsa ALGORITHM PARAMETER KeySize ::= { 2 5 8 1 1 }
KeySize ::= INTEGER
pkcs-1 OBJECT IDENTIFIER ::= { 1 2 840 113549 1 1 }
rsaEncryption ALGORITHM PARAMETER NULL ::= { pkcs-1 1 }
algorithm OBJECT IDENTIFIER ::= { 1 3 14 3 2 }
rsaSignature ALGORITHM PARAMETER NULL ::= { 1 3 14 3 2 11 }

```

#### DSA-Schlüsselalgorithmen

Für DSA-Schlüsselalgorithmen besteht der öffentliche Schlüssel *subjectPublicKey* aus einem Integerwert, und für das Teilfeld *algorithm* sind die Varianten *dsa* und *dsaCommon* definiert.

Die dsa-Variante ist in [ANS X9.30] mit dem Objektbezeichner *dsa* und dem Parameter *DSAParameters* als Folge der Teilfelder *prime1*, *prime2* und *base* vom Typ INTEGER definiert.

Die dsaCommon-Variante benutzt gemeinsame Parameter, die extern verteilt werden und enthält somit ein leeres Parameterfeld.

#### ASN.1-Definitionen

```

DSAPublicKey ::= INTEGER
DSAParameters ::= SEQUENCE {
    prime1          INTEGER,
    prime2          INTEGER,
    base            INTEGER }

```

**dsa** ALGORITHM PARAMETER DSAParameters ::= { 1 3 14 3 2 12 }  
**dsaCommon** ALGORITHM PARAMETER NULL ::= { 1 3 14 3 2 20 }

### ECDSA-Schlüsselalgorithmen

Für ECDSA-Schlüsselalgorithmen (elliptic curve digital signature algorithm, digitale Signaturalgorithmen basierend auf elliptischen Kurvenverfahren) [ANS X9.62, PKIX ECDSA 97] besteht der öffentliche Schlüssel *subjectPublicKey* aus einem Oktettstring, wobei die Abbildung von Oktett- auf Bitstring derart erfolgt, daß das MSB (most significant bit, höchstwertiges Bit) des Oktettstrings zum MSB des Bitstrings usw. und das LSB (least significant bit, niederwertiges Bit) des Oktettstrings zum LSB des Bitstrings wird.

Objektbezeichner für ECDSA-Algorithmen sind von ANSI (american national standards institute, US-Normungsgremium) unter dem Objektbezeichnerzweig *ansi-x9-62* definiert. Gegenwärtig existieren unter diesem Zweig nur die Objektbezeichner *id-ecPublicKey* und *ecdsa-with-sha1*, wobei letzterer nur zum Signieren von Zertifikaten, Sperrlisten oder PKI-Nachrichten mit leerem Parameterfeld benutzt werden soll. In diesem Fall werden durch den Algorithmus zwei Werte *r* und *s* erzeugt, die durch die ASN.1-Struktur *Ecdsa-SigValue* als Folge zweier INTEGER-Werte kodiert werden. Mit dem ersten Objektbezeichner *id-ecPublicKey* können Parameter im Zertifikat durch die Struktur *ECPParameters* explizit spezifiziert werden. Weitere Informationen zu der Parameterstruktur *ECPParameters* sind in der Spezialliteratur [ANS X9.62] zu finden.

In [MTRUST 97] wurde für das Schlüsselmanagement der Objektbezeichner *ecamvSign* festgelegt, der ein ECDSA-Signaturverfahren nach der Variante von "Agnew-Mullin-Vanstone" [ISO/IEC 14888] beinhaltet.

### ASN.1-Definitionen

```
ECDSAPublicKey ::= OCTET STRING  
Ecdsa-SigValue ::= SEQUENCE {  
    r          INTEGER,  
    s          INTEGER }  
ansi-x9-62 OBJECT IDENTIFIER ::= { 1 2 840 10045 }  
id-publicKeyType OBJECT IDENTIFIER ::= { 1 2 840 10045 2 }  
id-ecPublicKey OBJECT IDENTIFIER ::= { 1 2 840 10045 2 1 }  
ecsieSign OBJECT IDENTIFIER ::= { 1 3 36 3 3 2 }
```

An dieser Stelle sei darauf hingewiesen, daß ECDSA-Algorithmen zwar nach dem derzeitigen Kenntnisstand kryptographische Sicherheitsanforderungen erfüllen, aber zur Zeit kaum in Anwendungen eingesetzt werden und wenig Erfahrungen in deren praktischem Einsatz vorliegen.

Weitere Informationen zu Sicherheitsalgorithmen sind in [MTRUST 96, OIW 95, PKIX PRO 97] zusammengestellt. In diesen Spezifikationen wird u.a. definiert, wie die Komponenten der RSA-, DSA- und ECDSA-Algorithmen innerhalb des Bitstring-Teilfeldes *subjectPublicKey* nach DER zu kodieren sind.

ISIS-Konformitätsanforderungen

Zum Signieren geeignete und zugelassene Algorithmen werden von der Regulierungsbehörde für Telekommunikation und Post im Bundesanzeiger veröffentlicht. Die für die kommenden sechs Jahre (14. Februar 1998 - 14. Februar 2004) als geeignet beurteilten Kryptoalgorithmen sind in der Teilspezifikation "A2 Signatur" [A2 99, 6] aufgelistet. Um die zur Erstellung einer Signatur geeigneten und zugelassenen Hashalgorithmen im Zertifikat kenntlich zu machen, soll die Zertifizierungsrichtlinie, die in der Zertifikatserweiterung *certificate policies* identifiziert wird, die signaturgesetzkonformen Algorithmen beschreiben (siehe Abschnitt Z 2.3.9.4).

**Tabelle 14: Implementations-technische Informationen über *subjectPublicKeyInfo***

BEZEICHNER	WERTEBEREICH EINZELWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	ZERTIFIKATSTYP			RELEVANZ		
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten
	(BEISPIELE)	(BEISPIELE)	294						
SubjectPublicKeyInfo	SEQUENCE {	<b>30</b> 82 01 22	294						
AlgorithmIdentifier	SEQUENCE {	<b>30</b> 0D							
Algorithm	{ 1 2 840 113549 1 1 1 },	<b>06</b> 09 2A 86 48 86 F7							
Parameters	NULL },	<b>0D</b> 01 01 01							
SubjectPublicKey	BIT STRING	<b>05</b> 00							
Modulus	SEQUENCE {	<b>03</b> 82 01 0F 00							
PublicExponent	INTEGER, } }	<b>30</b> 82 01 0A							
		<b>02</b> 82 01 01 00 ...							
		<b>02</b> 03 01 00 01							

## Z 2.3.8. EINDEUTIGE BEZEICHNER

### Zweck

Die optionalen *issuerUniqueIdentifier*- und *subjectUniqueIdentifier*-Felder dienen zur eindeutigen Kennung bei Wiederverwendung von Namen von Zertifizierungsstellen und Zertifikatsinhabern.

### ASN.1-Definitionen

```
TBSCertificate ::= SEQUENCE {
    ...,
    issuerUniqueID ::= [1] IMPLICIT UniqueIdentifier OPTIONAL,
    subjectUniqueID ::= [2] IMPLICIT UniqueIdentifier OPTIONAL,
    ... }
```

```
UniqueIdentifier ::= BIT STRING
```

### Allgemeine Konformitätsanforderungen

Technische Komponenten sollten die Fähigkeit besitzen, *unique identifier* verarbeiten zu können, wenn sie auf Zertifikate mit diesen Erweiterungen treffen. Falls technische Komponenten diese Erweiterung jedoch nicht verarbeiten können, so sollten sie – in Analogie zur Behandlung von nicht bekannten und als *critical* gekennzeichneten Erweiterungen – Zertifikate zurückweisen, die diese Komponenten enthalten.

### ISIS-Konformitätsanforderungen

Die Benutzung der Felder *issuerUniqueIdentifier*- und *subjectUniqueIdentifier* ist bei der Generierung von Zertifikaten verboten. Zertifikatsinhaber und Zertifikatsersteller werden über die Komponenten *subject*, *subjectAltName*, *issuer* und *issuerAltName* identifiziert.

**Tabelle 15: Implementations-technische Informationen über *UniqueIdentifier***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	ZERTIFI- KATSTYP						RELE- VANZ
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	
	(BEISPIELE)	(BEISPIELE)								
issuerUniqueID										
SubjectUniqueID										

## Z 2.3.9. ERWEITERUNGEN

### Zweck

Zertifikatserweiterungen dienen zur Verknüpfung von zusätzlichen Attributen mit Benutzern oder öffentlichen Schlüsseln, die für die Verwaltung und den Betrieb der Zertifizierungshierarchie und der Sperrlisten benötigt werden. Zertifikate können eine beliebige Anzahl von Erweiterungen inklusive privat definierter Erweiterungen beinhalten.

Das *extensions*-Erweiterungsfeld besteht aus einer Folge von einzelnen Erweiterungen, die sich jeweils aus den Teilfeldern *extnId* als Objektbezeichner der betreffenden Erweiterung, *critical* als Flag zur Kennzeichnung der Wichtigkeit der Erweiterung und *extnValue* als konkreter Wert der Erweiterung zusammensetzen. Die Typdefinitionen der einzelnen Erweiterungen werden formal durch die *EXTENSION*-Klasse festgelegt. Danach enthält das Erweiterungsfeld *extnValue* die DER-Kodierung eines durch *&ExtnType* spezifizierten konkreten Typs für eine bestimmte Erweiterung. Das Erweiterungsfeld *extnId* enthält die DER-Kodierung des durch *&id* spezifizierten Objektbezeichners, durch den die neue Objektstruktur *&ExtnType* identifiziert wird. Objektbezeichner für Erweiterungen sind unter dem X.509-Objektbezeichnerzweig *id-ce* angeordnet.

### ASN.1-Definitionen

```

TBSCertificate ::= SEQUENCE {
    . . .
    extensions [3] EXPLICIT Extensions OPTIONAL }

Extensions ::= SEQUENCE (1..MAX) OF Extension

Extension ::= SEQUENCE {
    extnId OBJECT IDENTIFIER,
    critical BOOLEAN DEFAULT FALSE,
    extnValue OCTET STRING }

EXTENSION ::= CLASS {
    &id OBJECT IDENTIFIER UNIQUE,
    &ExtType }

WITH SYNTAX {
    SYNTAX &ExtnType
    IDENTIFIED BY &id }

certificateExtension OBJECT IDENTIFIER ::= { 2 5 29 }

id-ce OBJECT IDENTIFIER ::= certificateExtension
  
```

### Übersicht über die Zertifikatserweiterungen

In [ITU-T X.509 97] werden die folgenden Zertifikatserweiterungen definiert: *authority key identifier*, *subject key identifier*, *key usage*, *extended key usage*, *private key usage period*, *certificate policies*, *policy mapping*, *subject alternative name*, *issuer alternative name*, *subject directory attributes*, *basic constraints*, *name constraints*, *policy constraints* und *CRL distribution points*.

### Allgemeine Konformitätsanforderungen

Das optionale *extensions*-Erweiterungsfeld darf nur in Kombination mit der Version 3 verwendet werden. Anwendungen oder Systeme müssen Erweiterungen, die durch das *critical*-Feld als "wichtig" markiert wurden, immer auswerten und hierbei ggf. bei unbekannten Erweiterungen das Zertifikat als nicht verifizierbar erklären. Nicht-kritische Erweiterungen haben Informationscharakter und können bei der Gültigkeitsprüfung eines Zertifikates ignoriert werden. Innerhalb eines Zertifikates darf eine bestimmte Erweiterung nur einmal auftreten.

### ISIS-Konformitätsanforderungen

ISIS-konforme Zertifizierungsstellen müssen bei der Zertifikatserstellung stets die Erweiterungen *basic constraints*, *key usage*, *certificate policies* und *authority key identifier* verwenden. Zertifikate für den Zeitstempeldienst und den Verzeichnisdienst müssen darüber hinaus die Erweiterung *extended key usage* beinhalten. Desweiteren wird die optionale Unterstützung für die Erweiterungen *subject alternative name*, *issuer alternative name*, *subject key identifier*, *private key usage period*, *policy mapping*, *policy constraints*, *subject directory attributes*, *CRL distribution points* und *authority information access* empfohlen. Alle anderen Erweiterungen, hierzu gehört *name constraints*, dürfen nicht verwendet werden.

ISIS-konforme Zertifizierungsstellen können darüber hinaus weitere proprietäre Erweiterungen (*private extensions*) definieren und unterstützen, die aber bei einer Kennzeichnung als *critical* Interoperabilität verhindern können. Aus diesem Grund werden ISIS-spezifische proprietäre Erweiterungen stets als *non-critical* markiert. Im Rahmen dieses Profils werden ISIS-spezifische proprietäre Erweiterungen für die Anforderungen (13)-(17) der Tabelle 1 definiert.

ISIS-konforme Anwendungen und Systeme dürfen die ISIS-spezifischen proprietären und als *non-critical* markierten Erweiterungen bei der Gültigkeitsprüfung eines Zertifikates nicht ignorieren, wenn sie eine anwendungsbezogene Bedeutung haben, sondern ISIS-konforme Anwendungen und Systeme müssen diese auswerten und gegebenenfalls auch Zertifikate wegen einer proprietären nicht-kritischen Erweiterung zurückweisen.

Im folgenden werden die Erweiterungen gemäß ihrer Priorität für dieses Profils inklusive der ISIS-privaten Erweiterungen näher beschrieben.

**Tabelle 16: Implementations-technische Informationen über Erweiterungen**

ERWEITERUNG	ZERTIFI- KATSTYP			RELE- VANZ			KLASSIFI- KATION				
	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
basicConstraints											
keyUsage											
extKeyUsage											
certificatePolicies											
subjectAltName											
issuerAltName											
authorityKeyIdentifier											
subjectKeyIdentifier											
cRLDistributionPoints											
policyMappings											
policyConstraints											
subjectDirectoryAttributes											
nameConstraints											
privateKeyUsagePeriod											
authorityInfoAccess											
liabilityLimitationFlag											
dateOfCertGen											
procuration											
admission											
monetaryLimit											
declarationOfMajority											
iCCSN											
pKReference											
restriction											

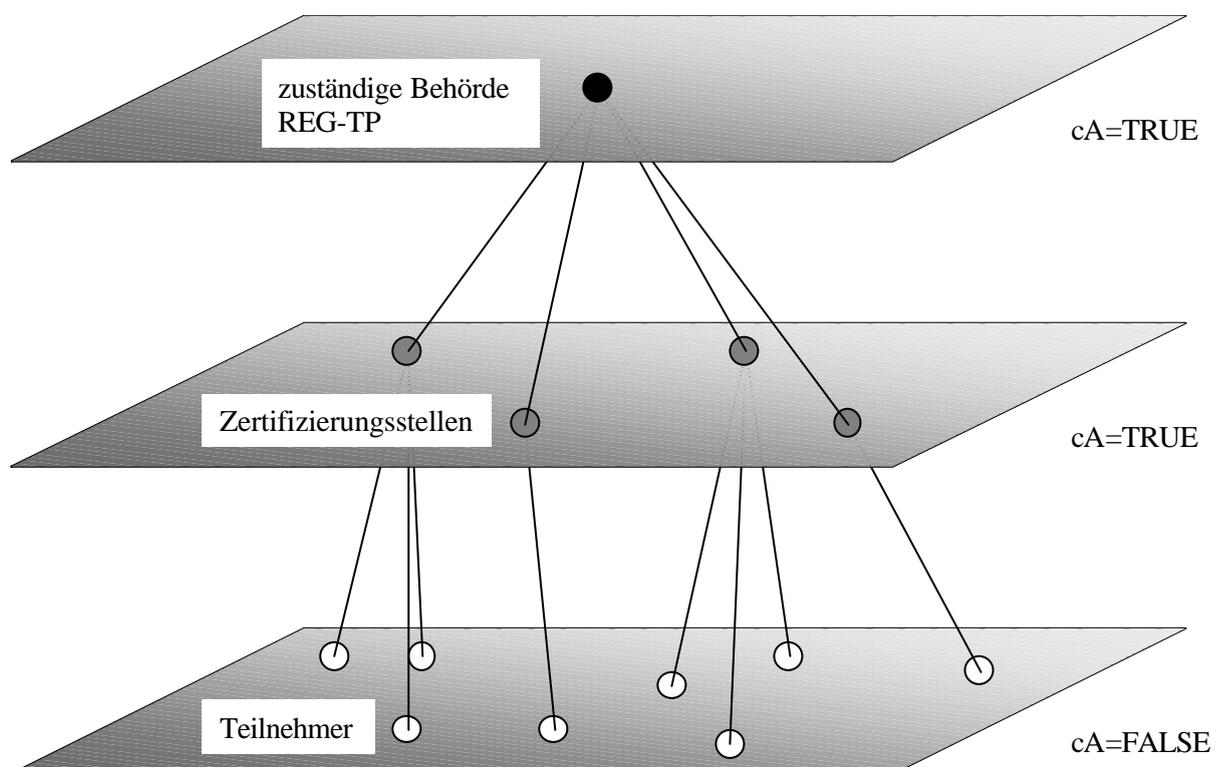
### Z 2.3.9.1. Zertifizierungsstellen- und Endbenutzer-Zertifikate

#### Zweck

Das Signaturgesetz [SigG, §2-4] und der Maßnahmenkatalog [MKAT 97] geben eine Sicherheitsinfrastruktur vor, bei der eine zweistufige Hierarchie von Zertifizierungsstellen etabliert wird. Die RegTP übernimmt dabei nach dem Signaturgesetz die Rolle der Wurzelinstanz und zertifiziert ausschließlich öffentliche Signaturschlüssel genehmigter Zertifizierungsstellen, Verzeichnis- und Zeitstempeldienste. Zertifizierungsstellen wiederum zertifizieren ausschließlich die öffentlichen Signaturschlüssel der Teilnehmer.

Durch die *basicConstraints*-Erweiterung wird mit Hilfe der *cA*-Komponente angezeigt, ob ein Zertifikatsinhaber in der Rolle als Zertifizierungsstelle auftreten kann, d.h. ob sein zertifizierter öffentlicher Schlüssel zur Verifikation von Zertifikatssignaturen benutzt werden kann. Falls dies der Fall ist, kann auch eine Beschränkung der Zertifizierungspfadlänge mittels der *pathLenConstraint*-Komponente angegeben werden. Sie liefert die maximale Anzahl von Zertifizierungsstellen-Zertifikaten, die einem Zertifikat in einem Zertifizierungspfad folgen können. Der Wert 0 zeigt an, daß nur noch Endanwenderzertifikate und kein Zertifikat einer Zertifizierungsstelle folgen kann. Ansonsten, d.h. bei fehlendem Feld, gibt es keine Beschränkung der Zertifizierungspfadlänge. Die Bedeutung der *basicConstraints*-Erweiterung ist in der Abbildung 2 veranschaulicht.

**Abbildung 2: Rolle von Zertifikatsinhabern**



ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

basicConstraints EXTENSION ::= {
    WITH SYNTAX {
        SYNTAX          BasicConstraintsSyntax
        IDENTIFIED BY   id-ce-basicConstraints }
    certificateExtension OBJECT IDENTIFIER ::= { 2 5 29 }
    id-ce OBJECT IDENTIFIER ::= certificateExtension
    id-ce-basicConstraints OBJECT IDENTIFIER ::= { 2 5 29 19 }
    BasicConstraintsSyntax ::= SEQUENCE {
        cA              BOOLEAN DEFAULT FALSE,
        pathLenConstraint INTEGER (0..MAX) OPTIONAL }

```

Allgemeine Konformitätsanforderungen

X.509v3-konforme Zertifizierungsstellen müssen die Erzeugung dieser Erweiterung in Zertifikaten unterstützen. Dies gilt auch für Endanwenderzertifikate, die allerdings hierfür nur einen leeren SEQUENCE-Wert einsetzen. Zertifikate für Zertifizierungsstellen sollen die *basicConstraints*-Erweiterung mit der *cA*-Komponente auf *TRUE* gesetzt und als *critical* markiert enthalten. Systeme müssen die *basicConstraints*-Erweiterung verarbeiten können.

ISIS-Konformitätsanforderungen

Bei der Erstellung von Zertifikaten ist die Benutzung der *pathLenConstraint*-Komponente optional. Die Beschränkung der Zertifizierungspfadlänge sollte sich aus der Sicherheitsrichtlinie der jeweiligen Zertifizierungsstelle ergeben. Die RegTP stellt nur Zertifikate für Zertifizierungsstellen, die Zeitstempeldienste und die Verzeichnisdienste aus. Die untergeordneten Zertifizierungsstellen stellen nur Zertifikate für Endbenutzer aus. Die Benutzung der *basicConstraints*-Erweiterung ist in allen Zertifikaten obligatorisch und als *critical* markiert.

Die *basicConstraints*-Erweiterung hat für Endanwender-, Zeitstempeldienst- und Verzeichnisdienstzertifikate die Länge 14 Bytes und für die RegTP- und Zertifizierungsstellenzertifikate die Länge 20 Bytes.

**Tabelle 17: Implementations-technische Informationen über *basicConstraints***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP							RELEVANZ			KLASSIFIKATION			
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung			
	(BEISPIELE)	(BEISPIELE)	[BYTES] 20														
basicConstraints extnId critical extnValue cA pathLenConstr.	SEQUENCE { { 2 5 29 19 }, TRUE OCTET STRING SEQUENCE { TRUE, 0 } }	30 12 06 03 55 1D 13 01 01 FF 04 08 30 06 01 01 FF 02 01 00	20														
basicConstraints extnId critical extnValue cA	SEQUENCE { { 2 5 29 19 }, TRUE OCTET STRING SEQUENCE { FALSE } }	30 0C 06 03 55 1D 13 01 01 FF 04 02 30 00	14														

### Z 2.3.9.2. Verwendungszwecke des Schlüsselpaares

#### Zweck

Das *keyUsage*-Erweiterungsfeld dient zur Anzeige der Verwendungszwecke des in einem Zertifikat enthaltenen Schlüssels, der z.B. zur Datenverschlüsselung oder zur Signaturerzeugung eingesetzt werden kann. Mit Hilfe dieser Erweiterung können die Verwendungszwecke eines Schlüssels eingeschränkt und nur für bestimmte Schlüsseloperationen zugelassen werden. Innerhalb der *keyUsage*-Struktur wird eine bestimmte Schlüsseloperation durch Setzen des entsprechenden Bits im *keyUsage*-Bitstring auf den Wert 1 definiert. Prinzipiell sind durch das PKIX-Profil beliebige Bitkombinationen zugelassen, von denen aber nur gewisse Teilmengen für eine konkrete Anwendung sinnvoll sind. Weitere anwendungsabhängige Nutzungsarten von zertifizierten Schlüsseln können durch das in Abschnitt Z 2.3.9.3 beschriebene *extKeyUsage*-Erweiterungsfeld definiert werden.

#### ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

```

```

keyUsage EXTENSION ::= {
  WITH SYNTAX {
    SYNTAX KeyUsage
    IDENTIFIED BY id-ce-keyUsage }
  id-ce OBJECT IDENTIFIER ::= { 2 5 29 }
  id-ce-keyUsage OBJECT IDENTIFIER ::= { 2 5 29 15 }
  KeyUsage ::= BIT STRING {
    digitalSignature (0),
    nonRepudiation (1),
    keyEncipherment (2),
    dataEncipherment (3),
    keyAgreement (4),
    keyCertSign (5),
    cRLSign (6),
    encipherOnly (7),
    decipherOnly (8) }

```

#### Allgemeine Konformitätsanforderungen

X.509v3-konforme Zertifizierungsstellen sollen die Generierung der *keyUsage*-Erweiterung unterstützen und stets als *critical* markieren.

X.509v3-konforme Systeme sollten das *keyUsage*-Erweiterungsfeld verarbeiten können. Die Bedeutung der einzelnen Bits zur Nutzungskennung ist in der folgenden Tabelle zusammengefasst.

**Tabelle 18: Nutzungsarten von öffentlichen Schlüsseln**

BIT#	BITNAME	VERWENDUNGSZWECK DER ZUGEHÖRIGEN SCHLÜSSELOPERATION
0	digitalSignature	allgemeine Prüfung digitaler Signaturen, die einen anderen als den durch die Positionen 1, 5 oder 6 angezeigten Zweck hat
1	nonRepudiation	Prüfung digitaler Signaturen zur Sicherung der Verbindlichkeit von Dokumenten und/oder Aktionen, die einen anderen als den durch die Position 5 oder 6 angezeigten Zweck hat
2	keyEncipherment	Schlüsseltransport, Schlüsselverwaltung
3	dataEncipherment	Verschlüsselung von Nutzdaten, die einen anderen als den durch die Position 2 angezeigten Zweck hat, d.h. die keine kryptographischen Schlüssel enthalten
4	keyAgreement	Schlüsselaustauschverfahren
5	keyCertSign	Prüfung der Zertifikatsignatur einer Zertifizierungsstelle
6	cRLSign	Prüfung der Sperrlistensignatur einer Zertifizierungsstelle
7	encipherOnly	Schlüsselaustauschverfahren zur alleinigen Verschlüsselung von Daten, falls auch das Bit 4 gesetzt ist, ansonsten ist der Verwendungszweck undefiniert
8	decipherOnly	Schlüsselaustauschverfahren zur alleinigen Entschlüsselung von Daten, falls auch das Bit 4 auch gesetzt ist, ansonsten ist der Verwendungszweck undefiniert

Die Bezeichnung des Bits *digitalSignature* beschreibt die Nutzungsart nur unzureichend. Digitale Signaturen sind Mechanismen, die Dienste wie Authentifizierung oder Sicherung der Verbindlichkeit ermöglichen. Ein

passenderer Name für das Bit *digitalSignature* wäre *authentication*, womit die eigentliche Nutzungsart beschrieben werden würde.

Die Verwendung der beiden Bits *digitalSignature* und *nonRepudiation* unterscheiden sich insofern, daß Authentifikations-Prozesse in der Regel automatisch und recht häufig ablaufen, wohingegen digitale Signaturen zur Sicherung der Verbindlichkeit bewußt und weniger häufig vom Zertifikatsinhaber ausgeführt werden.

### ISIS-Konformitätsanforderungen

Im Rahmen des ISIS-Profiles werden nur die in der folgenden Tabelle dargestellten Kombinationen von *keyUsage*-Bits berücksichtigt. Die Benutzung der *keyUsage*-Erweiterung ist in allen Zertifikaten obligatorisch und als *critical* zu markieren. Bei der Generierung von Benutzerzertifikaten darf nur das Bit *nonRepudiation* (Kombination 1) verwendet werden. Teilnehmerzertifikate sollen nicht für Authentifizierungszwecke benutzt werden. Desweiteren darf bei der Erzeugung von Zertifikaten für Zertifizierungsstellen und die RegTP nur das Bits *keyCertSign* (Kombination 2) benutzt werden. In Zertifikaten für den Verzeichnisdienst sind nur die Bits *cRLSign* und *nonRepudiation* (Kombination 3) zugelassen. In Zertifikaten für den Zeitstempeldienst ist nur das Bit *nonRepudiation* (Kombination 2) zugelassen.

**Tabelle 19: Benutzte Kombinationen von keyUsage-Bits**

SCHLÜSSEL-NUTZUNGSART	INSTANZEN	BIT#	KOMBINATION		
			1	2	3
Digital Signature		0			
Non Repudiation	Anwender, Zeitstempeldienst, Verzeichnisdienst	1			
Key Encipherment		2			
Data Encipherment		3			
Key Agreement		4			
Key CertSign	Zertifizierungsstellen	5			
CRL Sign	Verzeichnisdienst	6			
Encipher Only		7			
Decipher Only		8			

**Tabelle 20: Implementations-technische Informationen über *keyUsage***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFI- KATSTYP							RELE- VANZ			KLASSIFI- KATION	
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung	
	(BEISPIELE)	(BEISPIELE)	[BYTES] 16												
keyUsage extnId critical extnValue keyCertSign	SEQUENCE { { 2 5 29 15 }, TRUE, OCTET STRING BIT STRING }	30 0E 06 03 55 1D 0F 01 01 FF 04 04 03 02 01 04	16												
keyUsage extnId critical extnValue cRLSign+ nonRepubd.	SEQUENCE { 2 5 29 15 }, TRUE, OCTET STRING BIT STRING }	30 0E 06 03 55 1D 0F 01 01 FF 04 04 03 02 01 42	16												
keyUsage extnId critical extnValue nonRepudiation	SEQUENCE { 2 5 29 15 }, TRUE, OCTET STRING BIT STRING }	30 0E 06 03 55 1D 0F 01 01 FF 04 04 03 02 06 40	16												

### Z 2.3.9.3. Anwendungsabhängige Verwendungszwecke des Schlüsselpaars

#### Zweck

Das in [ITU-T X.509 97] enthaltene *extKeyUsage*-Erweiterungsfeld dient zur Definition von anwendungsabhängigen Nutzungsarten von zertifizierten Schlüsseln. Es kann zusätzlich oder alternativ zum *keyUsage*-Erweiterungsfeld benutzt werden.

Im PKIX-Profil [PKIX PRO 97] wurden für diese optionale Erweiterung eine Reihe von Schlüsselnutzungsarten, sowie die Definitionen von deren zugehörigen Objektbezeichner festgelegt. Prinzipiell können Schlüsselnutzungsarten von jeder Organisation definiert werden, die einen Bedarf hierfür hat. Objektbezeichner für Schlüsselnutzungsarten müssen unter Berücksichtigung von [ITU-T X.660 92 | ISO/IEC 9834-1 93] definiert werden.

Objektbezeichner, die von PKIX festgelegt werden, sind unter dem Objektbezeichnerzweig *id-pkix* angeordnet. Hierunter liegt u.a. der Objektbezeichnerzweig *id-kp*, der die anwendungsabhängigen Schlüsselnutzungsarten, wie beispielsweise Zeitstempeldienste, definiert.

ISIS-spezifische Objektbezeichner für Signaturgesetz-Interoperabilität sind unter dem Objektbezeichnerzweig *id-isis* festgelegt, der seinerseits ein Zweig von TeleTrust ist. Das Verfahren für die Vergabe von neuen Objektbezeichnern, zum Beispiel für Schlüsselnutzungsarten, ist in Kapitel Z.2.3.9.15.11 beschrieben.

### ASN.1-Definitionen

```
Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

extKeyUsage EXTENSION ::= {

WITH SYNTAX {
    SYNTAX          ExtKeyUsageSyntax
    IDENTIFIED BY   id-ce-extKeyUsage }

id-ce OBJECT IDENTIFIER ::= { 2 5 29 }
id-ce-extKeyUsage OBJECT IDENTIFIER ::= { 2 5 29 37 }
ExtKeyUsageSyntax ::= SEQUENCE SIZE (1..MAX) OF KeyPurposeId
KeyPurposeId ::= OBJECT IDENTIFIER

id-pkix OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 }
id-pkix-kp OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 3 }
id-pkix-kp-time-Stamping OBJECT IDENTIFIER
                               ::= { 1 3 6 1 5 5 7 3 8 }
id-isis OBJECT IDENTIFIER ::= { 1 3 36 8 }
id-isis-kp OBJECT IDENTIFIER ::= { 1 3 36 8 2 }
id-pkix-kp-OCSPSigning OBJECT IDENTIFIER
                               ::= { 1 3 6 1 5 5 7 3 9 }
id-isis-kp-directoryService OBJECT IDENTIFIER
                               ::= { 1 3 36 8 2 1 }
```

### Allgemeine Konformitätsanforderungen

X.509v3-konforme Systeme und Zertifizierungsstellen brauchen das *extKeyUsage*-Erweiterungsfeld nicht unterstützen, weil die Zusammenarbeitsfähigkeit von PKI-Komponenten mit dieser Erweiterung beeinträchtigt sein kann.

Die *extKeyUsage*-Erweiterung kann von der Zertifizierungsstelle als *critical* oder *non-critical* gekennzeichnet werden. Im ersten Fall soll ein Zertifikat ausschließlich für den angezeigten Zweck benutzt werden. Im anderen Fall dient es nur zur Anzeige eines oder mehrerer erwünschter Schlüsselnutzungsarten und kann zum Auffinden

eines entsprechenden Schlüsselzertifikates einer Entität benutzt werden, die mehrere Schlüsselzertifikate besitzt. Das Erweiterungsfeld wird als Hinweissfeld interpretiert und es setzt nicht voraus, daß die Schlüsselnutzungsart durch die Zertifizierungsstelle nur auf den angezeigten Zweck beschränkt ist. Anwendungen können jedoch die Anzeige einer bestimmten Nutzungsart erfordern, die sie zur Akzeptanz eines Zertifikates benötigen. Falls ein Zertifikat sowohl eine *keyUsage*- als auch eine *extKeyUsage*-Erweiterung enthält, die beide als *critical* markiert sind, so sind beide Felder unabhängig voneinander zu verarbeiten, und das Zertifikat ist nur für den Zweck einzusetzen, der mit beiden Feldern konsistent ist. Andernfalls soll das Zertifikat überhaupt nicht benutzt werden.

### ISIS-Konformitätsanforderungen

Die Benutzung des Objektbezeichners *id-kp-time-Stamping* ist bei der Generierung von Zertifikaten der Zeitstempeldienste obligatorisch. Die Benutzung eines der Objektbezeichner *id-pkix-kp-OCSPSigning* oder *id-isis-kp-directoryService* ist bei der Generierung von Zertifikaten der Verzeichnisdienste obligatorisch. Empfohlen wird wegen internationaler Interoperabilität die Benutzung von *id-pkix-kp-OCSPSigning* (das in [RFC 2560 99] eingeführt wird).

**Tabelle 21: Implementations-technische Informationen über *extKeyUsage***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP							RELEVANZ				KLASSIFIKATION				
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung					
	(BEISPIELE)	(BEISPIELE)	[BYTES] 24																
extKeyUsage extnId critical extnValue Verzeichnis- dienst	SEQUENCE { { 2 5 29 37 }, TRUE, OCTET STRING SEQUENCE { { 1 3 36 8 2 1 } } }	30 13 06 03 55 1D 25 01 01 FF 04 09 30 07 06 05 2B 24 08 02 01	21																
extKeyUsage extnId critical extnValue Zeitstempel- dienst	SEQUENCE { { 2 5 29 37 }, TRUE, OCTET STRING SEQUENCE { { 1 3 6 1 5 5 7 3 8 } } }	30 16 06 03 55 1D 25 01 01 FF 04 0C 30 0A 06 08 2B 06 01 05 05 07 03 08	24																

### Z 2.3.9.4. Zertifizierungsrichtlinien

#### Zweck

Das *certificatePolicies*-Erweiterungsfeld dient zur Anzeige der Verfahrensweisen bei der Erstellung eines Zertifikates durch die Zertifizierungsstelle und der Zwecke, die mit dem Zertifikat verbunden sind. Syntaktisch besteht die Erweiterungsstruktur aus einer Folge von *PolicyInformation*-Feldern, die jeweils Informationen über eine bestimmte angewandte Verfahrensweise enthalten. Hierzu gehören die Angabe eines Objektbezeichners der betreffenden Sicherheitsrichtlinien in der Teilkomponente *policyIdentifier* und die optionale Angabe sogenannter *policyQualifiers*-Merkmale. Jedes einzelne *policyQualifiers*-Merkmal wird durch einen eigenen *policyQualifierId*-Objektbezeichner und dessen zugehörige *qualifier*-Objektstruktur festgelegt.

#### ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

certificatePolicies EXTENSION ::= {
    WITH SYNTAX {
        SYNTAX          CertificatePoliciesSyntax
        IDENTIFIED BY   id-ce-certificatePolicies }
    id-ce OBJECT IDENTIFIER ::= { 2 5 29 }
    id-ce-certificatePolicies OBJECT IDENTIFIER ::= { 2 5 29 32 }
    CertificatePoliciesSyntax ::=
        SEQUENCE SIZE (1..MAX) OF PolicyInformation
    PolicyInformation ::= SEQUENCE {
        policyIdentifier CertPolicyId,
        policyQualifiers SEQUENCE SIZE (1..MAX) OF
            PolicyQualifierInfo OPTIONAL}
    CertPolicyId ::= OBJECT IDENTIFIER
    PolicyQualifierInfo ::= SEQUENCE {
        policyQualifierId PolicyQualifierId,
        qualifier          ANY DEFINED BY policyQualifierId }
    -- Internet-Definitionen
    id-qt ::= { 1 3 6 1 5 5 7 2 }
    id-qt-cps OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 2 1 }
    id-qt-unotice OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 2 2 }
    PolicyQualifierId ::= OBJECT IDENTIFIER
        ( id-qt-cps | id-qt-unotice )
    Qualifier ::= CHOICE
        { cpsUri CPSUri, userNotice UserNotice }

```

```

CPSuri ::= IA5String
UserNotice ::= SEQUENCE {
    NoticeRef NoticeReference OPTIONAL,
    explicitText DisplayText OPTIONAL}
NoticeReference ::= SEQUENCE {
    Organization IA5String,
    noticeNumbers SEQUENCE OF INTEGER }
DisplayText ::= CHOICE {
    VisibleString VisibleString,
    bmpString BMPString,
    utf8String UTF8String }
-- ISIS-Definitionen
id-isis OBJECT IDENTIFIER ::= { 1 3 36 8 }
id-isis-cp OBJECT IDENTIFIER ::= { 1 3 36 8 1 }
id-isis-cp-sigconform OBJECT IDENTIFIER ::= { 1 3 36 8 1 1 }

```

#### Allgemeine Konformitätsanforderungen

Zertifizierungsstellen sollen in der Lage sein, Zertifikate mit einem oder mehreren *policyIdentifier*-Feldern zu erzeugen. Die *certificatePolicies*-Erweiterung kann von Zertifizierungsstellen als *critical* oder *non-critical* gekennzeichnet werden. Jede Institution kann bei Bedarf weitere Typen für *policyIdentifier* und *policyQualifiers* selbst definieren. Konforme Zertifizierungsstellen müssen das optionale *policyQualifiers*-Teilfeld nicht erzeugen.

Die IETF-PKIX Arbeitsgruppe empfiehlt dringst die Verwendung eines einfachen Objektbezeichners im *certificatePolicies*-Erweiterungsfeld. Optionale *policyQualifiers*-Merkmale sollten nicht die Definition der betreffenden Verfahrensweise verändern, sondern nur Informationen darüber anbieten, wie die Zertifizierungsrichtlinien der Zertifizierungsstelle beschafft werden können. In [PKIX PRO 97] wurden die speziellen Objektbezeichner *id-qt-cps* und *id-qt-notice* für das *policyQualifierId*-Teilfeld und die speziellen zugehörigen Datenstrukturen *CPSuri* und *UserNotice* definiert, die von Zertifizierungsstellen zur Kennzeichnung ihrer Zertifizierungsprozedur und für Benutzermitteilungen verwendet werden können. *CPSuri* dient in diesem Zusammenhang als *URI* (universal resource identifier) für die *CPS* (certificate practise statement) der Zertifizierungsstelle, und *UserNotice* kann Textstrings enthalten.

X.509v3-konforme Systeme sollen in einem Zertifikat eventuell vorhandene *policyQualifiers*-Merkmale verarbeiten können. *PolicyQualifiers* können von Systemen wahlweise verarbeitet oder ignoriert werden.

Anwendungen können eine bestimmte *certificatePolicies*-Erweiterung im Zertifikat erwarten, die sie zur Verarbeitung eines Zertifikates benötigen. Sie sollen eine Liste der von ihnen akzeptierten Zertifizierungsrichtlinien besitzen, mit denen sie die *policyIdentifier*-Objektbezeichner eines Zertifikates vergleichen sollen. Der Zertifizierungspfad soll nur dann überprüft werden, wenn wenigstens einer der *policyIdentifier*-Objektbezeichner mit einem der Bezeichner in der Liste übereinstimmt.

Anwendungen, die keine speziellen Anforderungen an *certificatePolicies* Erweiterungen im Zertifikat haben, müssen auch keine Liste der akzeptierten Zertifizierungsrichtlinien führen und können jedes gültige Zertifikat

akzeptieren, unabhängig davon ob *certificatePolicies*-Erweiterungen enthalten sind oder ob das Feld als *critical* gekennzeichnet ist.

### ISIS-Konformitätsanforderungen

Die Verwendung der *certificatePolicies*-Erweiterung ist bei der Erstellung von Zertifikaten obligatorisch und dabei als *non-critical* zu kennzeichnen. Hierdurch wird die internationale Kompatibilität von Zertifikaten unterstützt. Darüberhinaus dürfen Zertifizierungsstellen auch nicht-signaturgesetzeskonforme Zertifikate ausstellen, wenn sie hierbei keine oder andere Objektbezeichner verwenden, aus denen dieser Sachverhalt eindeutig hervorgeht. Signaturgesetzeskonforme Anwendungen müssen auf jeden Fall die *certificatePolicies*-Erweiterung auswerten. Falls in einem Zertifikat mehrere *PolicyInformation*-Felder vorhanden sind, so muß nur eines davon ausgewertet werden. Bei der Erstellung von Zertifikaten, die als "konform zum Signaturgesetz" ausgezeichnet werden, muß der Objektbezeichners *id-isis-cp-sigconform* für das Feld *policyIdentifier* verwendet werden. Diese Zertifizierungsrichtlinie muß u. a. die zulässigen und geeigneten Hashalgorithmen benennen, die vom Zertifikatsinhaber verwendet werden dürfen. Zusätzlich können *policyQualifiers*-Merkmale verwendet werden, wie beispielsweise eine URI, die auf die zugrundeliegende CPS verweist.

Zertifizierungsstellen, die Signaturschlüssel-Zertifikate ausstellen, können mit externen Stellen wie Berufsverbänden, Kammern, Vereinigungen, Behörden, Firmen usw. bilaterale Verträge schließen oder Vereinbarungen treffen, um die Verantwortlichkeiten bei der Antragannahme, der Prüfung der Inhalte von Erweiterungen/Attributen und der Erstellung von Zertifikaten zu regeln. Für die Prüfung der Inhalte von Erweiterungen und Attributen können somit externe Stelle zuständig und verantwortlich sein. Nach erfolgreicher Prüfung der Daten übergeben diese Stellen die geprüften Informationen an die Zertifizierungsstellen. Durch die digitale Signatur einer Zertifizierungsstelle werden alle in dem ausgestellten Zertifikat enthaltenen Informationen durch die Zertifizierungsstelle beglaubigt. Damit bedeutet diese Signatur auch eine Beglaubigung der externen Stelle für die Erweiterungen- und Attributinhalt. Zertifizierungsrichtlinien der externen Stellen können im *certificatePolicies*-Erweiterungsfeld durch entsprechende Objektbezeichner angezeigt werden. Sie müssen kompatibel zu den Zertifizierungsrichtlinien der Zertifizierungsstelle sein und dürfen lediglich zusätzliche Einschränkungen darstellen. Das Verfahren für die Vergabe von neuen Objektbezeichnern, zum Beispiel für Zertifizierungsrichtlinien, ist in Kapitel Z2.3.9.15.11 beschrieben.

**Tabelle 22: Implementations-technische Informationen über *certificatePolicies***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP							RELEVANZ				KLASSIFIKATION					
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung						
	(BEISPIELE)	(BEISPIELE)	[BYTES] 60																	
Certific.Policies	SEQUENCE {	30 37	57																	
ExtnId	{ 2 5 29 32 },	06 03 55 1D 20																		
Critical	FALSE,																			
ExtnValue	OCTET STRING	04 31																		
PolicyIdentifier	SEQ. OF {SEQ. {	30 2E 30 2C																		
	{ 1 3 36 8 1 1 },	06 05 2B 24 08 01 01																		
PolicyQualifiers	SEQUENCE {	30 23																		
policyQualif.Id	{ 1 3 6 1 5 5 7 2 1 },	06 08 2B 06 01 05 05																		
	{ } }	07 02 01																		
CPSuri	"http://www.regtp.d e/Fachinfo/Digitalsi gn/neu/policy.htm" }	16 37 68 74 74 70 3A 2F 2F 77 77 77 ...																		

### Z 2.3.9.5. Alternative Namen von Zertifikatsinhabern

#### Zweck

Das *subjectAltName*-Erweiterungsfeld enthält einen oder mehrere alternative Namen für Zertifikatsinhaber, durch die zusätzliche identitätsgebundene Merkmale an den Zertifikatsinhaber gebunden werden.

Alternative Namen von Zertifikatsinhabern können in einem bestimmten Namensformat aus einem Spektrum von Formaten des Typs *GeneralName* angegeben werden, die in der folgenden Tabelle zusammengestellt sind. Die in [ITU-T 97] vordefinierten Optionen umfassen u.a. Namen für elektronische Post *rfc822*, Verzeichnisdiensteinträge *directoryName*, Internet-Protokolladressen *iPAddress*, URIs *registeredID* und lokale Definitionen.

Gemäß Signaturgesetz können Zertifikate nur für natürliche Personen ausgestellt werden, Rollen- oder Instanzenzertifikate sind nicht vorgesehen. Zertifikate von Zertifizierungsstellen, Verzeichnisdiensten und Zeitstempeldiensten sind somit auch an natürliche Personen gebunden, die aber auch unter einem Pseudonym auftreten können.

Das Signaturgesetz schließt nicht aus, daß ein Signaturschlüssel-Inhaber verschiedene technische Namen führen kann. Diese sind dann in der Regel abhängig von den verschiedenen Rollen des Signaturschlüssel-Inhabers. Eine Privatperson könnte ein Zertifikat für die Verwendung zur Kommunikation mit Behörden besitzen. Dieses würde die Person als Staatsbürger identifizieren. Dieselbe Person könnte ein weiteres Zertifikat besitzen, welches sie für private Geschäftszwecke verwendet. Dieses Zertifikat könnte beispielsweise die postalische Adresse beinhalten. Daneben sind ebenfalls Zertifikate denkbar, die eine Verbindung zu einem bestimmten Arbeitgeber bestätigen und für Erklärungen im Namen des Unternehmens verwendet wird. Ähnliches gilt für Zertifikate, die von Standesvereinigungen als Bestätigung der Mitgliedschaft ausgestellt werden. Anstatt mehrerer Signaturschlüssel-Zertifikate mit verschiedenen technischen Namen können auch zu einem Signaturschlüssel-Zertifikat mehrere Attributzertifikate ausgestellt und verwendet werden.

Adressat eines Namens ist die verifizierende Person, die in die Lage versetzt werden soll, über diesen Namen Vertrauen in die geleistete Unterschrift zu haben. Daraus folgt, daß dieser Name sprechend im Sinne einer Verwendung durch Personen sein sollte und daß dieser Name diejenigen Daten enthalten sollte, die im jeweiligen Anwendungskontext als üblich erachtet werden.

**Tabelle 23: Übersicht von *GeneralName* Formattypen**

GENERALNAME TYPNAMEN	BEDEUTUNG DER FORMATE UND BEISPIELE	RELEVANTE NORMEN
otherName	beliebiges Format, das als Instanz der OTHER-NAME Informationsobjektklasse definiert ist	[ITU-T X.681 94]
rfc822Name	Format für E-Mail-Adressen im Internet user@darmstadt.gmd.de	[RFC 822 82]
dNSName	Format für Domännennamen in Internet sonne.darmstadt.gmd.de	[RFC 1035 87]
x400Address	Format für O/R-Adressen S=user; P=darmstadt; A=gmd; C=de	[ITU-T X.411]
directoryName	Format für Verzeichnisdienstnamen CN=vorname name, L=darmstadt, O=gmd, C=DE	[ITU-T X.501 97]
ediPartyName	Format für elektronischen Dokumentenaustausch	
uniformResource- Identifizier	Format für universelle Betriebsmittelbezeichner (URI) im World-Wide-Web http://www.gmd.de oder ftp://... oder ldap://...	[RFC 1630 94]
iPAddress	Format für Internet-Protokoll-Adressen 141.12.63.6	[RFC 791 81]
registeredId	Format für Bezeichner von registrierten Objekten	[ITU-T X.660 92]

ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

subjectAltName EXTENSION ::= {
    WITH SYNTAX {
        SYNTAX          SubjectAltName
        IDENTIFIED BY   id-ce-subjectAltName }
    id-ce OBJECT IDENTIFIER ::= { 2 5 29 }
    id-ce-subjectAltName OBJECT IDENTIFIER ::= { 2 5 29 17 }
    SubjectAltName ::= GeneralNames
    GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName
    GeneralName ::= CHOICE {
        otherName          [0] OTHER-NAME,
        rfc822Name         [1] IA5String,
        dNSName            [2] IA5String,
        x400Address        [3] ORAddress,
        directoryName      [4] Name,
        ediPartyName       [5] EDIPartyName,
        uniformResourceIdentifier [6] IA5String,
        iPAddress          [7] OCTET STRING,
        registeredID       [8] OBJECT IDENTIFIER }
    OTHER-NAME ::= SEQUENCE {
        type-id           OBJECT IDENTIFIER,
        value             [0] EXPLICIT ANY DEFINED BY type-id }
    TYPE-IDENTIFIER ::= CLASS {
        &id               OBJECT IDENTIFIER UNIQUE,
        &Type }
    WITH SYNTAX {
        &Type
        IDENTIFIED BY   &id }
    EDIPartyName ::= SEQUENCE {
        nameAssigner      [0] DirectoryString OPTIONAL,
        partyName         [1] DirectoryString }

```

Allgemeine Konformitätsanforderungen

Die Erweiterung kann mehrere Namen unterschiedlichen oder gleichen Formats enthalten. Zertifizierungsstellen können die *subjectAltName*-Erweiterung als *critical* oder als *non-critical* kennzeichnen. Die *subjectAltName*-Erweiterung kann in einem Zertifikat immer dann benutzt werden, wenn weitere identitätsgebundene Merkmale an das Zertifikat gekoppelt werden sollen. Sie muß außerdem dann benutzt und als *critical* gekennzeichnet werden, wenn im betreffenden Zertifikat nur die alternative Form zur Identifikation des Zertifikatsinhabers benutzt wird. Außerdem muß in diesem Fall das *subject*-Zertifikatfeld als eine leere Folge kodiert werden.

Die Benutzung des String-Platzhaltersymbol "\*" in Namenstypen des *subjectAltName*-Erweiterungsfeldes ist verboten. Der URI-Name muß ein absoluter Pfadname sein, der einen Rechner bezeichnet. Der URI-Zugriff kann über FTP, HTTP, LDAP oder E-MAIL erfolgen. Die *subjectAltName*-Erweiterung muß, wenn sie in einem Zertifikat benutzt wird, mindestens einen Eintrag enthalten.

Für jedes Namensformat, das im *GeneralName*-Typ benutzt wird, muß es nach [ITU-T X.509 97] ein Namens-Registrierungssystem geben, das die eindeutige Identität von Entitäten für die Zertifizierungsstelle und die Zertifikatsbenutzer gewährleistet.

Die *subjectAltName*-Erweiterung sollte als *non-critical* markiert werden, falls das *subject*-Feld des Zertifikates einen Verzeichnisnamen enthält, der den Zertifikatsinhaber eindeutig identifiziert. Konforme Systeme, die diese Erweiterung unterstützen, müssen nicht alle Namensformate verarbeiten können. Es muß jedoch bei als *critical* markierten Erweiterungen zumindestens eines der in einem Zertifikat enthaltenen Formate erkannt und verarbeitet werden können, andernfalls ist das Zertifikat als nicht verifizierbar zu betrachten. Nicht erkannte oder nicht unterstützte Namensformate können ignoriert werden. Systeme müssen das alternative URI-Namensformat verarbeiten und die LDAP URL [RFC 1959 96] erkennen können. Andere URI-Formate müssen nicht erkannt werden.

### ISIS-Konformitätsanforderungen

Ob und wie dieses Feld verwendet wird, ist freigestellt.

#### BEISPIELE: BENUTZUNG DER *SUBJECTALTNAME*-ERWEITERUNG

(1) Mailadresse:

*rfc822Name*: Vorname.Name@Organisation.de

(2) X.500-Verzeichnisdienstname mit Angabe der E-Mail Adresse:

*directoryName*: CN=Zertifizierungsstelle,  
EMAIL= Zertifizierungsstelle@Organisation-der-ZS.de,  
O=Organisation-der-ZS,C=DE

### Z 2.3.9.6. Alternative Namen von Zertifizierungsstellen

#### Zweck

Das *issuerAltName*-Erweiterungsfeld enthält einen oder mehrere alternative Namen für den Ersteller eines Zertifikates, durch die zusätzliche Entitäten an die Zertifizierungsstelle gebunden werden.

Neben dem *distinguished name* der Zertifizierungsstelle können im alternativen Namensfeld des Ausstellers zusätzliche Adreßinformationen zur Erreichbarkeit im Internet abgelegt werden. Dazu gehören insbesondere die Angabe einer Internetadresse für elektronische Post, Angaben über den DNS-Namen der Zertifizierungsstelle (DNS, domain name system).

Die Adresse für elektronische Post (rfc822) sollte eine symbolische Mailadresse sein, die es einem Teilnehmer ermöglicht, Kontakt zur Zertifizierungsstelle aufzunehmen. Es sollen an dieser Stelle keine persönlichen Mailadressen von Mitarbeitern verwendet werden.

Der DNS-Name der Zertifizierungsstelle sollte der registrierte Domain-Name der Zertifizierungsstelle sein. Über diesen Namen können Anwendungen die Adressen zusätzlicher Dienste und Protokolle der Zertifizierungsstelle ermitteln. Beispiele wären die Adresse eines World Wide Web Servers, eines Verzeichnisdienstes oder eines Zeitstempeldienstes. Diese Vorgehensweise ist eine Alternative zur Verwendung des globalen X.500 Verzeichnisdienst. Mit etablierten Verfahren (beispielsweise [RFC 2052 96]) können die Adressen der gewünschten Dienste aus dem angegebenen DNS-Namen abgeleitet werden. Zu beachten ist, daß Informationen über den DNS-Namen einer Zertifizierungsstelle auch im *distinguished name* der Zertifizierungsstelle angegeben sein können. Dies geschieht durch die Definition des sog. *DC*-Bezeichners (DC, domain component, Teilname eines Domänennamens) für *distinguished names* [RFC 2247 98]).

#### ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

issuerAltName EXTENSION ::= {
    WITH SYNTAX {
        SYNTAX          IssuerAltName
        IDENTIFIED BY   id-ce-issuerAltName }
id-ce-issuerAltName OBJECT IDENTIFIER ::= { 2 5 29 18 }

IssuerAltName ::= GeneralNames

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
    otherName       [0] OTHER-NAME,

```

```
rfc822Name          [1] IA5String,
dNSName             [2] IA5String,
x400Address         [3] ORAddress,
directoryName       [4] Name,
ediPartyName        [5] EDIPartyName,
uniformResourceIdentifier[6] IA5String,
iPAddress           [7] OCTET STRING,
registeredID        [8] OBJECT IDENTIFIER }

OTHER-NAME ::= SEQUENCE {
  type-id          OBJECT IDENTIFIER,
  value            [0] EXPLICIT ANY DEFINED BY type-id }

EDIPartyName ::= SEQUENCE {
  nameAssigner    [0] DirectoryString OPTIONAL,
  partyName       [1] DirectoryString }
```

#### Allgemeine Konformitätsanforderungen

Es gelten die gleichen Aussagen wie für *subjectAltName*.

#### ISIS-Konformitätsanforderungen

Die optionale *issuerAltName*-Erweiterung kann bei der Erstellung von Zertifikaten benutzt werden und ist dabei als *non-critical* zu kennzeichnen. Diese Erweiterung besitzt jedoch keine Bedeutung für die technische Identifikation des Zertifikatsinhabers, sondern sie bindet lediglich weitere Merkmale (z.B. E-Mail-Adressen) an ihn. Weitere Informationen über Namenskonventionen sind im Anhang I zu finden.

#### BEISPIELE FÜR DIE BENUTZUNG DER *ISSUERALTNAME*-ERWEITERUNG:

- (1) Mailadresse:

```
rfc822Name: rootca@regtp.de
```

- (2) X.500-Verzeichnisdienstname mit Angabe der E-Mail Adresse der Zertifizierungsstelle:

```
directoryName: CN=Verzeichnisdienst, EMAIL=ca@cert.de, O=ZS1, C=DE
```

**Tabelle 25: Implementations-technische Informationen über *issuerAltName***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP							RELEVANZ	KLASSIFIKATION			
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional		Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 500												
issuerAltName	SEQUENCE {	30 15	23												
extnId	{ 2 5 29 18 },	06 03 55 1D 12													
critical	FALSE,														
extnValue	OCTET STRING	04 0E													
rfc822Name	SEQUENCE OF { [1] "ca@cert.de" }	30 0C 81 0A 63 61 40 63 65 72 74 2E 64 65													

### Z 2.3.9.7. Identifizierung von Signaturschlüsseln von Zertifizierungsstellen

#### Zweck

Das *authorityKeyIdentifier*-Erweiterungsfeld dient zur Identifizierung eines bestimmten öffentlichen Schlüssels und/oder eines bestimmten Zertifikates einer Zertifizierungsstelle zum Signieren eines Zertifikates. Die Erweiterung wird dann verwendet, wenn eine Zertifizierungsstelle mehrere Signaturschlüssel – sei es als gleichzeitig aktive Schlüssel oder zum Schlüsselwechsel – besitzt. Die Identifizierung kann entweder durch den Schlüsselnamen im *keyIdentifier*-Teilfeld oder durch den Namen der Zertifizierungsstelle im *authorityCertIssuer*-Teilfeld und die Seriennummer im *authorityCertSerialNumber*-Teilfeld erfolgen.

Die Kombination *authorityCertIssuer* und *authorityCertSerialNumber* identifiziert eindeutig ein bestimmtes Zertifikat einer Zertifizierungsstelle. Der *keyIdentifier* kennzeichnet den öffentlichen Schlüssel der Zertifizierungsstelle und somit nicht nur eine, sondern gegebenenfalls eine Reihe von Zertifikaten. Damit erlaubt die Verwendung des *keyIdentifiers* eine größere Flexibilität zum Auffinden eines Zertifizierungspfades. Außerdem müssen die Zertifikate, die den *keyIdentifier* im *authorityKeyIdentifier*-Erweiterungsfeld benutzen, nicht zurückgezogen werden, wenn die Zertifizierungsstelle sich bei gleichbleibendem Schlüssel ein neues Zertifikat ausstellen läßt.

Andererseits ist die Flexibilität zum Auffinden eines Zertifizierungspfades nicht immer gewünscht. Verfügt eine Zertifizierungsstelle über mehrere Zertifikate für den gleichen Schlüssel, die aber beispielsweise verschiedene Haftungsgrenzen beinhalten, so ist es erforderlich,

nicht nur den öffentlichen Schlüssel sondern genau dasjenige Zertifikat der Zertifizierungsstelle zu referenzieren, das für den jeweiligen Teilnehmer gültig ist.

### ASN.1-Definitionen

```
Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

authorityKeyIdentifier EXTENSION ::= {
    WITH SYNTAX {
        SYNTAX          AuthorityKeyIdentifier
        IDENTIFIED BY   id-ce-authorityKeyIdentifier }
    id-ce-authorityKeyIdentifier OBJECT IDENTIFIER ::= { 2 5 29 35}
    AuthorityKeyIdentifier ::= SEQUENCE {
        keyIdentifier      [0] KeyIdentifier OPTIONAL,
        authorityCertIssuer [1] GeneralNames OPTIONAL,
        authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL }
    KeyIdentifier ::= OCTET STRING
```

### Allgemeine Konformitätsanforderungen

X.509v3-konforme Zertifizierungsstellen, die diese Erweiterung generieren, sollen entweder die beiden *authorityCertIssuer*- und *authorityCertSerialNumber*-Teilfelder in Zertifikate integrieren oder beide weglassen. Im zweiten Fall muß stattdessen das *keyIdentifier*-Teilfeld eingebaut werden.

Falls beide Identifizierungsmethoden benutzt werden, sollte die Zertifizierungsstelle deren Konsistenz sicherstellen. Ein Schlüsselbezeichner soll bezüglich aller Schlüsselbezeichner, die eine Zertifizierungsstelle für einen Zertifikatsinhaber benutzt, eindeutig sein.

Systeme sollten die Fähigkeit besitzen, Zertifizierungspfade finden und validieren zu können, wenn die ausstellende Zertifizierungsstelle mehrere Signaturschlüssel besitzt. Sie sollten eine der beiden Identifikationsmethoden zum Auffinden von Zertifizierungspfaden unterstützen.

### ISIS-Konformitätsanforderungen

Die Benutzung dieser Erweiterung ist in allen Zertifikaten obligatorisch und sie muß als *non-critical* gekennzeichnet werden. Außerdem muß als Schlüsselidentifizierungsmethode die Verwendung der beiden *authorityCertIssuer*- und *authorityCertSerialNumber*-Teilfelder unterstützt werden, die ein bestimmtes Zertifikat der Zertifizierungsstelle eindeutig identifiziert. Dabei muß im *authorityCertIssuer*-Teilfeld zumindest der *issuer*-Name des Zertifikaterstellers vom Typ *directoryName* angegeben werden.

**Tabelle 26: Implementations-technische Informationen über *authorityKeyIdentifier***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFI- KATSTYP				RELE- VANZ			KLASSIFI- KATION			
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 280											
authorityKeyId extnId critical extnValue authCertIssuer directoryName countryName value	SEQUENCE { { 2 5 29 35 }, FALSE, OCTET STRING SEQUENCE { [1] SEQ. OF { [4] SEQ. OF { SET OF SEQ. { { 2 5 4 6 }, "DE" }	30 68 06 03 55 1D 23  04 60 30 5E 80 57 84 55 30 53 31 0B 30 09 06 03 55 04 06 13 02 44 45	106											
organization Name value organizational Unit value commonName value authCertSerNum	SET OF SEQ. { { 2 5 4 10 }, "RegTP" } SET OF SEQ. { { 2 5 4 11 }, "Wurzelzertifizie rungsstelle" } } SET OF SEQ. { { 2 5 4 3 }, "DEPCA" } } }, [2] 1 }	31 0E 30 0C 06 03 55 04 0A 13 05 52 45 47 54 50 31 24 30 22 06 03 55 04 0B 13 1B 57 75 72 7A 65 6C 7A 65 72 74 69 66 69 7A 69 65 ... 31 0E 30 0C 06 03 55 04 03 13 05 44 45 50 43 41 82 03 02 01 01												

### Z 2.3.9.8. Identifizierung von öffentlichen Teilnehmerschlüsseln

#### Zweck

Das *subjectKeyIdentifier*-Erweiterungsfeld dient zur Identifizierung eines bestimmten öffentlichen Schlüssels eines Zertifikatinhabers. Hat ein Zertifikatsinhaber seinen öffentlichen Schlüssel von mehreren Zertifizierungsstellen zertifizieren lassen, so ermöglicht diese Erweiterung das schnelle Auffinden aller Zertifikate, die denselben öffentlichen Schlüssel beinhalten.

ASN.1-Definitionen

```
Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

subjectKeyIdentifier EXTENSION ::= {
    WITH SYNTAX {
        SYNTAX          SubjectKeyIdentifier
        IDENTIFIED BY   id-ce-subjectKeyIdentifier }
    id-ce-subjectKeyIdentifier OBJECT IDENTIFIER ::= { 2 5 29 14}
    SubjectKeyIdentifier ::= KeyIdentifier
    KeyIdentifier ::= OCTET STRING
```

Allgemeine Konformitätsanforderungen

Ein Schlüsselbezeichner soll bezüglich aller Schlüsselbezeichner, die ein Zertifikatsinhaber benutzt, eindeutig sein. Falls ein Zertifikat das *subjectKeyIdentifier*-Erweiterungsfeld nicht enthält und eine Referenz auf eine Schlüsselbezeichner benötigt wird, sollte entweder der 160 Bit SHA-1 Hashwert des öffentlichen Teilnehmerschlüssels aus dem *subjectPublicKeyInfo*-Feld des Zertifikates hierzu benutzt werden, oder es kann eine verkürzte Form verwendet werden, die aus den niederwertigen 60 Bit des 160 Bit SHA-1 Hashwerts plus 4 führenden Bits als Kennung mit dem Wert '0100'B, d.h. insgesamt aus 64 Bit besteht. Der Hashwert soll dabei nur über das zugehörige Inhaltsfeld und nicht über das vorangehende Tag- und Längenfeld berechnet werden. Die Erweiterung muß stets als *non-critical* gekennzeichnet werden. Sofern eine Beschränkung hinsichtlich der Größe von Zertifikaten eine Rolle spielt, kann als *subjectKeyIdentifier* eine fortlaufende Nummer verwendet werden.

ISIS-Konformitätsanforderungen

Das optionale *subjectKeyIdentifier*-Erweiterungsfeld kann bei der Erstellung von Zertifikaten unterstützt werden. Es ist stets als *non-critical* zu kennzeichnen.

**Tabelle 27: Implementations-technische Informationen über *subjectKeyIdentifier***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFI- KATSTYP				RELE- VANZ			KLASSIFI- KATION			
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 31											
subjectKeyId	SEQUENCE {	30 1D	31											
extnId	{ 2 5 29 14 },	06 03 55 1D 0E												
critical	FALSE,													
extnValue	OCTET STRING	04 16												
keyIdentifier	OCTET STRING }	04 14												
SHA-1 Hashwert		0D 95 ED 1B B3 6A 94 EF 2A 83 30 37 24 33 9D C9 3E 52 9A 9F												

### Z 2.3.9.9. Informationen zur Beschaffung von Sperrlisten

#### Zweck

Die *cRLDistributionPoints*-Erweiterung enthält Informationen, die zur Beschaffung von Sperrlisten dienen.

An dieser Stelle sei darauf hingewiesen, daß zu dem Thema “*cRLDistributionPoints*” das US-Patent 5,699,431 von Entrust Technologies Inc. existiert, das aber weltweit und gebührenfrei benutzt werden darf.

#### ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue      OCTET STRING }

cRLDistributionPoints EXTENSION ::= {

WITH SYNTAX {
    SYNTAX          CRLDistPointsSyntax
    IDENTIFIED BY  id-ce-cRLDistributionPoints }

id-ce-cRLDistributionPoints OBJECT IDENTIFIER ::= { 2 5 29 31 }

CRLDistPointsSyntax ::= SEQUENCE SIZE (1..MAX) OF
    DistributionPoint

DistributionPoint ::= SEQUENCE {

```

```
    distributionPoint      [0]   DistributionPointName OPTIONAL,
    reasons                [1]   ReasonFlags OPTIONAL,
    cRLIssuer              [2]   GeneralNames OPTIONAL }

DistributionPointName ::= CHOICE {
    fullName                [0]   GeneralNames,
    nameRelativeToCRLIssuer [1]   RelativeDistinguishedName }

ReasonFlags ::= BIT STRING {
    unused(0),
    keyCompromise(1),
    cACompromise(2),
    affiliationChanged(3),
    superseded(4),
    cessationOfOperation (5),
    certificateHold(6) }
```

### Allgemeine Konformitätsanforderungen

Falls der *DistributionPointName*-Name einer *CRL*-Verteilungsstelle im *URI*-Format angegeben wird, so ist die *URI* als ein Pointer auf die aktuelle Sperrliste anzusehen, deren zugehörigen Sperrgründe durch das Feld *reasons* und deren Ersteller durch das Feld *cRLIssuer* gekennzeichnet werden können. Die Werte im *URI*-Format (http, https, ldap, ftp) unterliegen denselben Einschränkungen wie für *subjectAltName*-Erweiterungen. Falls das optionale *reasons*-Teilfeld in der Erweiterung nicht verwendet wird, so soll die Sperrliste gesperrte Zertifikate für alle Sperrgründe enthalten. Falls das optionale *cRLIssuer*-Teilfeld nicht benutzt wird, so soll die Sperrliste von derjenigen Zertifizierungsstelle erstellt werden, die das Zertifikat erzeugt hat.

Prinzipiell können *CRLs* (certificate revocation list, Sperrliste von Zertifikaten) mittels geeigneter Segmentierkriterien wie beispielsweise Seriennummernbereiche in disjunkte Teil-*CRLs* aufgeteilt werden. Hierbei ist zu beachten, daß die *CRL* über Information über das Segmentierungskriterium enthalten muß. Verwendet eine Zertifizierungsstelle mehrere verschiedene *URIs*, so müssen diese alle auf die gleiche Information zeigen.

### ISIS-Konformitätsanforderungen

Die optionale *cRLDistributionPoints*-Erweiterung muß als *non-critical* markiert werden, so daß statt der Benutzung von Sperrlisten auch andere Mechanismen wie z.B. On-line Prüf-dienste, die zwingend vorgeschrieben sind, zur Verifikation herangezogen werden können. Diese Option soll durch Zertifizierungsstellen und Anwendungen unterstützt werden. Die Benutzung des *reasons*-Teilfeld ist bei der Generierung von Zertifikaten verboten. Es soll keine Segmentierung der Sperrliste vorgenommen werden.

Es wird empfohlen, diese Erweiterung zu verwenden, und zwar um die Angabe über den Ort, an dem eine passende *CRL* zu finden ist, in das Benutzerzertifikat einzubinden.

Wenn der *CRL*-Herausgeber nicht die Zertifizierungsstelle ist, die dieses Zertifikat ausgestellt hat, so muß der Name des *CRL*-Herausgebers im *cRLIssuer*-Teilfeld angegeben werden.

**Tabelle 28: Implementations-technische Informationen über *cRLDistributionPoint***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFI- KATSTYP				RELE- VANZ		KLASSIFI- KATION		
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 100									
CRLDist.Point	SEQUENCE {	30 29	43									
extnId	{ 2 5 29 31 },	06 03 55 1D 1F										
critical	FALSE,											
extnValue	OCTET STRING	04 22										
distributionP.	SEQ. OF{ SEQ. {	30 20 30 1E										
	[0] SEQ. OF {	A0 1C 30 1A										
fullName	[6]	86 18										
uRI	"http://www.regtp. de/crls" } } }	68 74 74 70 3A 2F 2F 77 77 77 2E ...										

### Z 2.3.9.10. Anerkennung von fremden Zertifizierungsrichtlinien

#### Zweck

Nach dem Signaturgesetz [SigG 97, §15] und der Signaturverordnung [SigV, §8] erfolgt die Anerkennung ausländischer Zertifikate und der damit verbundenen fremden Sicherheitsrichtlinien ausschließlich durch die digitale Signatur der zuständigen Behörde. Die *policy-Mappings*-Erweiterung kann nur in Zertifikaten für Zertifizierungsstellen verwendet werden und enthält eine Folge von Objektbezeichnerpaaren, die jeweils aus einem *issuerDomainPolicy*- und einem *subjectDomainPolicy*-Teilfeld bestehen. Durch die gepaarte Struktur zeigt eine ausstellende Zertifizierungsstelle die Äquivalenz ihrer Zertifizierungsrichtlinien mit denen des Zertifikatsinhabers an, der ebenfalls eine Zertifizierungsstelle ist. Dieser Sachverhalt wird in der Abbildung 3 veranschaulicht.

#### ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

```

```

policyMappings EXTENSION ::= {
    WITH SYNTAX {
        SYNTAX          PolicyMappingsSyntax
        IDENTIFIED BY   id-ce-policyMappings }
}

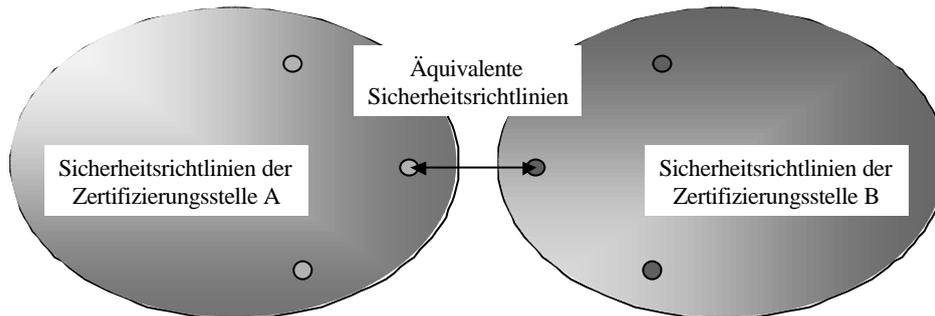
```

```

id-ce-policyMappings OBJECT IDENTIFIER ::= { 2 5 29 33 }
PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF
SEQUENCE {
    issuerDomainPolicy      CertPolicyId,
    subjectDomainPolicy    CertPolicyId }
CertPolicyId ::= OBJECT IDENTIFIER

```

**Abbildung 3: Anerkennung fremder Sicherheitsrichtlinien**



### ISIS-Konformitätsanforderungen

Diese optionale Erweiterung kann im Zusammenhang mit der Anerkennung ausländischer Zertifizierungsinfrastrukturen [SigG 97, §15] wichtig werden. Ausschließlich die RegTP soll die *policyMappings*-Erweiterung erzeugen können und Systeme sollen die *policyMappings*-Erweiterung verarbeiten können. Sie sollte hierbei stets als *non-critical* gekennzeichnet werden.

**Tabelle 29: Implementations-technische Informationen über *policyMappings***

BEZEICHNER	WERTEBEREICH EINZELWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP							RELEVANZ			KLASSIFIKATION		
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung		
	(BEISPIELE)	(BEISPIELE)	[BYTES] 30													
policyMapp extnId critical extnValue issuerDom.Pol. subjectDom.Pol.	{ { 2 5 29 33 }, FALSE, OCTET STRING SEQ. OF SEQ. { { { 1 3 36 8 1 1 }, -- fremder Root- CA OID } } }	30 ... 06 03 55 1D 21 04 22 30 ... 30 ... 06 05 2B 24 08 01 01 06 ...	>25													

### Z 2.3.9.11. Verzeichnisattributwerte für Zertifikatsinhaber

#### Zweck

Das stets als *non-critical* zu verwendende *subjectDirectoryAttributes*-Erweiterungsfeld dient zur Bereitstellung von Verzeichnis-Attributwerten für den Zertifikatsinhaber. Die Typdefinitionen der einzelnen Attribute werden formal durch die *ATTRIBUTE*-Klasse festgelegt. Danach enthält das Attributfeld *values* die DER-Kodierung eines durch *&Type* spezifizierten konkreten Typs für ein bestimmtes Attribut, das durch den Objektbezeichner *&id* identifiziert wird. Das Attributfeld *type* enthält die DER-Kodierung des durch *&id* spezifizierten konkreten Objektbezeichners für dieses Attribut. Hierunter fallen alle in [ITU-T X.520 95] vordefinierten Attribute wie beispielsweise *commonName*, sowie zusätzliche Attribute, die durch die Festlegung neuer Objektbezeichner für *&id* und der zugehörigen Objektstrukturen für *&Type* definiert werden können.

#### ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

subjectDirectoryAttributes EXTENSION ::= {
    WITH SYNTAX {
        SYNTAX          AttributesSyntax
        IDENTIFIED BY   id-ce-subjectDirectoryAttributes }
    id-ce-subjectDirectoryAttributes OBJECT IDENTIFIER
        ::= { 2 5 29 9}
    AttributesSyntax ::= SEQUENCE SIZE (1..MAX) OF Attribute
    Attribute ::= SEQUENCE {
        type           AttributeType,
        values         SET OF AttributeValue
    }
    AttributeType ::= ATTRIBUTE.&id
    AttributeValue ::= ATTRIBUTE.&Type
    ATTRIBUTE ::= CLASS {
        &id            OBJECT IDENTIFIER UNIQUE,
        &Type         }
    WITH SYNTAX {
        SYNTAX          &Type
        IDENTIFIED BY   &id }

```

#### Allgemeine Konformitätsanforderungen

Attribute können auch gemäß X.500 im *attributes*-Feld von Attributzertifikaten angegeben werden.

ISIS-Konformitätsanforderungen

ISIS-konforme Zertifizierungsstellen können die optionale Erweiterung *subjectDirectoryAttributes* bei der Erstellung von Zertifikaten unterstützen.

Beschränkungen, Zulassungen oder andere Zusatzinformationen können entweder als Attribute oder als private Erweiterungen in Zertifikaten enthalten sein. Attribute können sowohl in dem *subjectDirectoryAttributes*-Erweiterungsfeld als auch in einem Attributzertifikat gespeichert werden. Grundsätzlich gilt, Beschränkungen und Informationen, die für Zugriffsregelungen benötigt werden, müssen direkt im Zertifikat erkennbar sein. Wenn im Signaturzertifikat eine Beschränkung angezeigt wird, die nicht im Signaturschlüssel-Zertifikat selbst, sondern in einem Attributzertifikat enthalten ist, so muß das betreffende Attributzertifikat Teil des signierten Dokuments sein.

**Tabelle 30: Implementations-technische Informationen über *subjectDirectoryAttributes***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP								RELEVANZ	KLASSIFIKATION							
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung		private Erweiterung	critical-Markierung	non-critical-Markierung					
	(BEISPIELE)	(BEISPIELE)	[BYTES] 36																	
subjectDirAtt	SEQUENCE {	30 22	36																	
extnId	{ 2 5 29 9 },	06 03 55 1D 09																		
critical	FALSE,																			
extnValue	OCTET STRING	04 1A																		
atMonetaryLimit	SEQ. OF { SEQ. {	30 18 30 16																		
	{ 1 3 36 8 3 4 },	06 05 2B 24 08 03 04																		
currency	SET OF { SEQ {	31 0D 30 0B																		
amount	"DEM",	13 03 44 45 4D																		
exponent	1,	02 01 01																		
	4 } } } }	02 01 04																		

### Z 2.3.9.12. Beschränkungen von Zertifizierungsrichtlinien

#### Zweck

Die *policyConstraints*-Erweiterung dient zur Spezifikation von Beschränkungen, die zusätzlich bei der Überprüfung von Zertifizierungspfaden zu beachten sind. Durch eine solche Beschränkung kann eine Zertifizierungsstelle verhindern, daß nachfolgende Zertifizierungsstellen fremde Zertifizierungsrichtlinien anerkennen können. Darüberhinaus kann eine Zertifizierungsstelle mit dieser Erweiterung bewirken, daß alle nachfolgenden Zertifikate eine akzeptierte Zertifizierungsrichtlinie beinhalten müssen. Das optionale *inhibitPolicyMapping*-Teilfeld enthält die Anzahl von weiteren Zertifikaten, die im Zertifizierungspfad folgen können, ehe eine Anerkennung fremder Zertifizierungsrichtlinien verboten ist. Das optionale *requireExplicitPolicy*-Teilfeld enthält die Anzahl von weiteren Zertifikaten, die im Zertifizierungspfad folgen können, ehe eine akzeptierte Sicherheitsrichtlinie im Zertifikat enthalten sein muß. Wenn die *requireExplicitPolicy* Einschränkung Verwendung finden soll, müssen die von der zuständigen Behörde ausgestellten Zertifikate den Wert 0 an dieser Stelle enthalten. Durch die relativ flache Hierarchie, die vom Signaturgesetz festgeschrieben ist, kann nur die zuständige Behörde die *policyConstraints*-Erweiterung in Zertifikate einbauen, da die untergeordneten Zertifizierungsstellen nur Endanwender-Zertifikate ausstellen.

#### ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

policyConstraints EXTENSION ::= {
    WITH SYNTAX {
        SYNTAX          PolicyConstraintsSyntax
        IDENTIFIED BY   id-ce-policyConstraints }
    id-ce-policyConstraints OBJECT IDENTIFIER ::= { 2 5 29 36 }

PolicyConstraintsSyntax ::=
    SEQUENCE SIZE (1..MAX) OF PolicyConstraints

PolicyConstraints ::= SEQUENCE {
    requireExplicitPolicy [0] SkipCerts OPTIONAL,
    inhibitPolicyMapping  [1] SkipCerts OPTIONAL }

SkipCerts ::= INTEGER (0..MAX)

```

#### Allgemeine Konformitätsanforderungen

Die *policyConstraints*-Erweiterung kann nur in Zertifikaten benutzt werden, die für Zertifizierungsstellen ausgestellt worden sind. Zertifikate, in denen das Teilfeld *requireExplicitPolicy* enthalten ist, müssen in den nachfolgenden Zertifikaten einen anerkannten Bezeichner für die verwendeten Zertifizierungsrichtlinien enthalten. Konforme Zertifizierungsstellen dürfen keine Zertifikate erstellen, bei denen die *policyConstraints*-Erweiterung als eine leere Folge kodiert ist.

Zertifizierungsstellen müssen die Fähigkeit besitzen, diese Erweiterung in Zertifikate einzubauen, und falls sie benutzt wird, diese als *critical* zu kennzeichnen. Systeme müssen die Fähigkeit besitzen, diese Erweiterung zu verarbeiten.

### ISIS-Konformitätsanforderungen

Die Benutzung der *policyConstraints*-Erweiterung ist optional.

**Tabelle 31: Implementations-technische Informationen über *policyConstraints***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ		KLASSIFIKATION			
				Zertifizierungsstellen	Zeitempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES]										
policyConstraints													

### Z 2.3.9.13. Namensraum für Namen von Zertifikatsinhabern in Zertifikatketten

#### Zweck

Durch die *nameConstraints*-Erweiterung wird der Namensraum definiert, in dem Namen von Zertifikatsinhabern in nachfolgenden Zertifikaten eines Zertifizierungspfades liegen müssen. Die durch diese Erweiterung angegebenen Beschränkungen betreffen den *subject* DN-Namen oder die Alternativnamen *subjectAltName* eines Zertifikatsinhabers. Syntaktisch bestehen die Beschränkungen aus einer Folge von zugelassenen *permittedSubtrees*- oder verbotenen *excludedSubtrees*-Teilbaumnamen. Namen, die in einem *excludedSubtree*-Teilbaum liegen, sind ungültig, auch wenn sie ebenfalls zu einem *permittedSubtree* Teilbaum gehören. Einzelne *GeneralSubtree*-Teilbäume werden durch einen Teilbaumtypnamen *base* und die Baumtiefe *minimum* und *maximum* spezifiziert.

#### ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue      OCTET STRING }

nameConstraints EXTENSION ::= {
    WITH SYNTAX {
        SYNTAX          NameConstraintsSyntax
        IDENTIFIED BY   id-ce-nameConstraints }
  }

```

```

id-ce-nameConstraints OBJECT IDENTIFIER ::= { 2 5 29 30 }
NameConstraintsSyntax ::= SEQUENCE {
    permittedSubtrees [0] GeneralSubtrees OPTIONAL,
    excludedSubtrees [1] GeneralSubtrees OPTIONAL }
GeneralSubtrees ::=
    SEQUENCE SIZE (1..MAX) OF GeneralSubtree
GeneralSubtree ::= SEQUENCE {
    base GeneralName,
    minimum [0] BaseDistance DEFAULT 0,
    maximum [1] BaseDistance OPTIONAL }
BaseDistance ::= INTEGER (0..MAX)

```

### Allgemeine Konformitätsanforderungen

Die *nameConstraints*-Erweiterung darf nur in Zertifikaten für Zertifizierungsstellen benutzt, d. h. diese Erweiterung kann wiederum nur von der zuständigen Behörde in Zertifikate eingebaut werden. Die Erweiterung muß stets als *critical* markiert werden.

Im Fall von Alternativnamen dürfen nur solche Namensformen als *base* verwendet werden, die eine wohl definierte hierarchische Struktur haben.

Die Teilfelder *minimum* und *maximum* sollten nicht benutzt werden, d.h. die Teilbäume sollten stets in ihrer vollen Tiefe behandelt werden. Namensformate wie *rfc822*, *dNSName* und *uniformResourceIdentifier*, die sich auf den ASN.1-Typ *IA5String* zurückführen lassen dürfen das "\*" -Platzhaltersymbol für Teilstrings benutzen. In RFC- und URI-Namen wirkt sich die Erweiterung nur auf den Namensteil aus, der den Rechnernamen betrifft. Beschränkungen von Namen des *directoryName*-Namenstyps sollen auf das *subject*-Zertifikatsfeld und die *subjectAltName*-Erweiterungen der *directoryName*-Namenstypen angewandt werden. Beschränkungen von Namen des *x400Address*-Namenstyps sollen die *subjectAltName*-Erweiterungen der *x400Address*-Namenstypen angewandt werden

Systeme müssen die Fähigkeit besitzen, diese Erweiterung verarbeiten zu können.

### ISIS-Konformitätsanforderungen

Die Benutzung der *nameConstraints*-Erweiterung ist bei der Generierung von Zertifikaten für Zertifizierungsstellen verboten. Die Benutzung dieser Erweiterung ist nur in einer mehr als 2-stufigen Zertifizierungshierarchie sinnvoll.

**Tabelle 32: Implementations-technische Informationen über *policyConstraints***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP					RELEVANZ			KLASSIFIKATION				
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung		
	(BEISPIELE)	(BEISPIELE)	[BYTES]													
name Constraints																

### Z 2.3.9.14. Nutzungsdauer von privaten Schlüsseln

#### Zweck

Das *privateKeyUsagePeriod*-Erweiterungsfeld dient zur Festlegung von unterschiedlichen Gültigkeitsdauern von Zertifikaten und privaten Schlüsseln, die für digitale Signaturzwecke benutzt werden.

#### ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue      OCTET STRING }

```

```

privateKeyUsagePeriod EXTENSION ::= {
    WITH SYNTAX {
        SYNTAX          PrivateKeyUsagePeriod
        IDENTIFIED BY   id-ce-privateKeyUsagePeriod }
id-ce-privateKeyUsagePeriod OBJECT IDENTIFIER ::= { 2 5 29 16 }

```

```

PrivateKeyUsagePeriod ::= SEQUENCE {
    notBefore      [0] GeneralizedTime OPTIONAL,
    notAfter       [1] GeneralizedTime OPTIONAL }

```

#### Allgemeine Konformitätsanforderungen

Falls diese Erweiterung benutzt wird, so sollte sie als *non-critical* gekennzeichnet werden.

#### ISIS-Konformitätsanforderungen

Die Benutzung der *privateKeyUsagePeriod*-Erweiterung ist bei der Erstellung von Zertifikaten optional. Es wird empfohlen, sie nicht zu verwenden.

**Tabelle 33: Implementations-technische Informationen über *privateKeyUsagePeriod***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ		KLASSIFIKATION			
				Zertifizierungsstellen	Zeitmeldedienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 48										
pKUPeriod extnID critical extnValue notBefore notAfter	SEQUENCE { { 2 5 29 16 }, TRUE, OCT STRING SEQ { "19980101000000Z", "20030101000000Z" }	30 2E 06 03 55 1D 10 01 01 FF 04 24 03 22 18 0F 31 39 39 ... 18 0F 20 30 30 ...	48										

### Z 2.3.9.15. Private Zertifikatserweiterungen

Das allgemeine X.509v3-Zertifikatsformat gestattet auch die Definition von privaten Erweiterungen. Im PKIX-Profil [PKIX PRO 97] wurde bisher nur die private Erweiterung *authorityInfoAccess* und in der TeleSec-Spezifikation [TS ZF 98] wurde die private Erweiterung *liabilityLimitationFlag* eingeführt. Im Rahmen des ISIS-Profiles wurden die zusätzlichen privaten Erweiterungen *procuration* (Vertretungsmacht), *admission* (Zulassungsinformation), *dateOfCertGen* (Erstellungsdatum eines Zertifikates), *monetaryLimit* (monetäre Beschränkung), *iCCSN* (Chipkarten-Seriennummer), *declarationOfMajority* (Volljährigkeitserklärung), *pKReference* (Chipkarten-Schlüsselreferenz) und *restriction* (sonstige Einschränkungen) für die Themenbereiche (13) bis (17) der Tabelle 1 festgelegt.

Wenn als *critical* markierte Erweiterungen von einer Anwendung nicht erkannt werden, muß das Zertifikat zurückgewiesen werden. Bei *non-critical* markierten Erweiterungen kann die Anwendung diese verarbeiten, muß aber nicht. ISIS-konforme Anwendungen sollten alle hier definierten Erweiterungen erkennen und verarbeiten können, auch wenn sie aus internationalen Interoperabilitätsgründen als *non-critical* gekennzeichnet sind.

#### Z 2.3.9.15.1. ZUGRIFF AUF INFORMATIONEN UND DIENSTE DURCH ZERTIFIZIERUNGSSTELLEN

##### Zweck

Die *authorityInfoAccess*-Erweiterung enthält Informationen wie man auf Dienste der Zertifizierungsstelle und Informationen über die Zertifizierungsstelle zugreifen kann. Hierunter fallen On-line-Validierungsdienste und Daten über Zertifizierungsrichtlinien. Zu diesen Daten gehören beispielsweise Dienstadressen und Informationen über Zertifizierungsrichtlinien. Informationen über die Aufbewahrungsorte von Sperrlisten fallen jedoch nicht unter diesen

Erweiterungstyp, denn sie werden durch die *cRLDistributionPoints*-Erweiterung abgedeckt. Die Syntax der *authorityInfoAccess*-Erweiterung wird durch den Typ *AuthorityInfoAccessSyntax* definiert, der seinerseits aus einer Folge von mindestens einer Zugriffsbeschreibung *AccessDescription* besteht. Eine einzelne Zugriffsbeschreibung verweist auf ein bestimmtes Zugriffsformat *accessMethod* und den zugehörigen Zugriffsort *accessLocation*, der Zusatzinformationen über diejenige Zertifizierungsstelle enthält, die das Zertifikat ausgestellt und dabei die Zugriffsbeschreibung integriert hat.

Die private *authorityInfoAccess*-Erweiterung wurde in PKIX unter dem Objektbezeichnerzweig *id-pe* (pe, private extensions) definiert.

Die Zugriffsbeschreibungen wurden in PKIX unter dem Objektbezeichnerzweig *id-ad* (ad, access descriptors) definiert.

Gegenwärtig sind im PKIX-Profil unter diesem Zweig der Objektbezeichner *id-ad-ocsp* für den Zugang zum On-line-Validierungsdienst OCSP (on-line certificate status protocol) [PKIX OCSP 97] und der Objektbezeichner *id-ad-caIssuers* für den Zugriff auf Informationen übergeordneter Zertifizierungsstellen festgelegt. Beide Objektbezeichner werden in der Komponente *accessMethod* der Struktur *AccessDescription* kodiert.

### ASN.1-Definitionen

```
Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

authorityInfoAccess EXTENSION ::= {
    WITH SYNTAX {
        SYNTAX          AuthorityInfoAccessSyntax
        IDENTIFIED BY   id-pe-authorityInfoAccess }
id-pe OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 1 }
id-pe-authorityInfoAccess OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 1 1 }
id-ad OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 48 }
id-ad-ocsp OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 48 1 }
id-ad-caIssuers OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 48 2 }
AuthorityInfoAccessSyntax ::=
    SEQUENCE SIZE (1..MAX) OF AccessDescription
AccessDescription ::= SEQUENCE {
    accessMethod     OBJECT IDENTIFIER,
    accessLocation   GeneralName }
```

Allgemeine Konformitätsanforderungen

Die *authorityInfoAccess*-Erweiterung kann in Zertifikaten für Teilnehmer oder für Zertifizierungsstellen benutzt werden und muß immer als *non-critical* gekennzeichnet werden.

ISIS-Konformitätsanforderungen

Die Benutzung der *authorityInfoAccess*-Erweiterung ist in Zertifikaten optional. Werden Zugriffsinformationen in Zertifikate integriert, so müssen die Zertifikate zurückgezogen werden, wenn sich diese Dienstadresse ändert. Anstelle dieser Erweiterung können die erforderlichen Informationen auch auf andere Art und Weise (siehe Anhang II) bereitgestellt werden, um eine größere Flexibilität zu erreichen.

Es wird empfohlen diese Erweiterung in das Benutzerzertifikat aufzunehmen, und zwar um eine OCSP-Dienstadresse anzugeben. Es wird empfohlen, die Angabe in Form einer URI mit Transportprotokoll http zu nutzen. Andere Transportprotokolle sind zulässig.

**Tabelle 34: Implementations-technische Informationen über *authorityInfoAccess***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ	KLASSIFIKATION						
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer		obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 128												
authorityInfoAccess	SEQUENCE {	30 5D	95												
extnID	{ 1 3 6 1 5 5 7 1 1 },	06 08 2B 06 01 05 05 07 01 01													
critical	FALSE														
extnValue	OCT STR SEQ OF {	04 51 30 4F													
AccessDescript.	SEQUENCE {	30 2D													
accessMethod	{ 1 3 6 1 5 5 7 48 2 },	06 08 2B 06 01 05 05 07 30 02													
accessLocation	"http://www.ca.de/ ~policies/info.cp }	86 21 68 74 74 70 3A 2F 2F ...													
AccessDescript.	SEQUENCE {	30 1E													
accessMethod	{ 1 3 6 1 5 5 7 48 1 },	06 08 2B 06 01 05 05 07 30 01													
accessLocation	"http://www.ocsp.de } } }	86 12 68 74 74 70 3A 2F 2F ...													

## Z 2.3.9.15.2. KENNZEICHNUNG DER NUTZUNGSBESCHRÄNKUNG DES SIGNATURSCHLÜSSELS

Zweck

Die durch TeleSec [TS ZF 98] definierte private Erweiterung *liabilityLimitationFlag* dient zur Anzeige in einem Zertifikat, ob eine Beschränkung der Nutzung des Signaturschlüssels auf bestimmte Anwendungen nach Art und Umfang vorliegt. Das Flag wird dann benutzt, wenn die zugehörige Beschränkungsinformation als Attribut in einem Attributzertifikat enthalten ist. Die Voreinstellung für diese Erweiterung hat den Wert FALSE, der entweder anzeigt, daß keine Beschränkungen vorliegen oder daß Beschränkungen als Erweiterungen oder Attribute direkt im Zertifikat integriert sind.

ASN.1-Definitionen

```
Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

liabilityLimitationFlag EXTENSION ::= {
    WITH SYNTAX {
        SYNTAX          BOOLEAN DEFAULT FALSE
        IDENTIFIED BY   certExtensionLiabilityLimitationFlag }

certExtensionLiabilityLimitationFlag OBJECT IDENTIFIER
    ::= { 0 2 262 1 10 12 0 }
```

ISIS-Konformitätsanforderungen

Die Benutzung der *liabilityLimitationFlag*-Erweiterung in Zertifikaten ist für Zertifizierungsstellen obligatorisch, falls eine Beschränkung der Nutzungsart vorliegt und die Beschränkungsangaben als Attribut in einem Attributzertifikat vorliegen. Die Erweiterung soll in jedem Fall als *non-critical* gekennzeichnet werden. ISIS-konforme Systeme und Anwendungen müssen die *liabilityLimitationFlag*-Erweiterung erkennen und verarbeiten können.

Tabelle 35: Implementations-technische Informationen über *liabilityLimitationFlag*

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP								RELEVANZ		KLASSIFIKATION		
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung		
	(BEISPIELE)	(BEISPIELE)	[BYTES] 16													
liabilityLim.Flag extnID	SEQUENCE { { 1 3 36 8 3 1 },	30 0E 06 07 02 82 06	16													
critical extnValue	FALSE OCT STR TRUE }	01 0A 0C 00 04 03 01 01 FF														

## Z 2.3.9.15.3.

## ERSTELLUNGSDATUM EINES ZERTIFIKATES

Zweck

Die ISIS-spezifische private Erweiterung *dateOfCertGen* dient zur Anzeige des Erstellungsdatum eines Zertifikates. Sie wird durch den Objektbezeichner *id-isis-at-dateOfCertGen* referenziert.

ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue      OCTET STRING }

dateOfCertGen EXTENSION ::= {
    SYNTAX          DateOfCertGenSyntax
    IDENTIFIED BY  id-isis-at-dateOfCertGen }

id-isis-at OBJECT IDENTIFIER ::= { 1 3 36 8 3 }

id-isis-at-dateOfCertGen OBJECT IDENTIFIER ::= { 1 3 36 8 3 1 }

DateOfCertGenSyntax ::= GeneralizedTime

```

Statische Semantik

Bei der Kodierung der Datums- und Zeitpunkte ist für *GeneralizedTime* das in Abschnitt Z 2.3.5 beschriebene Format YYYYMMDDHHMMSSZ zu beachten.

ISIS-Konformitätsanforderungen

Die Benutzung der privaten Erweiterung *dateOfCertGen* durch Zertifizierungsstellen bei der Erzeugung von Zertifikaten ist optional und dabei als *non-critical* zu markieren. ISIS-konforme Systeme und Anwendungen müssen diese private Erweiterung erkennen können.

**Tabelle 36: Implementations-technische Informationen über *dateOfCertGen***

BEZEICHNER	WERTEBEREICH	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP							RELEVANZ				KLASSIFIKATION				
	ENZEILWERTE			Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung					
	(BEISPIELE)	(BEISPIELE)	[BYTES] 28																
dateOfCert G extnID critical extnValue	SEQUENCE { { 1 3 36 8 3 1 }, FALSE, OCT. STRING "19980101000000 Z" }	30 1A 06 05 2B 24 08 03 01  04 11 18 0F 31 39 39 38 30 31 30 31 30 30 30 ...	28																

*Z 2.3.9.15.4. VERTRETUNGSMACHT*Zweck

Die ISIS-spezifische, private Erweiterung *procuration* dient zur Anzeige der Vertretungsmacht für eine dritte Person. Sie wird durch den Objektbezeichner *id-isis-at-procuration* referenziert. Die zugehörige *ProcurationSyntax* enthält in der Komponente *signingFor* entweder den Namen der vertretenen Person (Teilkomponente *thirdPerson*) oder einen Verweis auf deren zugehöriges Basiszertifikat (Teilkomponente *certRef*) und in den optionalen Komponenten *country* und *typeSubstitution* das Land, für das die Vertretungsmacht gelten soll, sowie die Art der Vertretung. Durch das SEQUENCE-OF-Konstrukt kann in der Erweiterung die Vertretungsmacht für mehrere dritte Personen angegeben werden.

ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue      OCTET STRING }

```

```

procuration EXTENSION ::= {
    SYNTAX          ProcurationSyntax
    IDENTIFIED BY   id-isis-at-procuration }

```

```

id-isis-at-procuration OBJECT IDENTIFIER ::= {1 3 36 8 3 2}

```

```

ProcurationSyntax ::= SEQUENCE OF {
    country                PrintableString(SIZE(2)) OPTIONAL,
    typeOfSubstitution    DirectoryString OPTIONAL,
    signingFor            SigningFor }

SigningFor ::= CHOICE {
    thirdPerson           GeneralName,
    certRef              IssuerAndSerial }

IssuerAndSerial ::= SEQUENCE {
    issuer                GeneralNames,
    serial               CertificateSerialNumber }

CertificateSerialNumber ::= INTEGER

DirectoryString ::= CHOICE {
    printableString      PrintableString (SIZE (1..maxSize))
    teletexString        TeletexString (SIZE (1..maxSize))
    bmpString            BMPString (SIZE (1..maxSize))
    universalString      UniversalString (SIZE (1..maxSize)) }

```

### ISIS-Konformitätsanforderungen

Die Benutzung der privaten Erweiterung *procuration* durch Zertifizierungsstellen bei der Erzeugung von Zertifikaten ist optional und dabei als *non-critical* zu markieren. Der Systemparameter *maxSize* wird für *ProcurationSyntax* auf den Wert 128 festgelegt.

ISIS-konforme Systeme und Anwendungen müssen die private Erweiterung *procuration* erkennen und verarbeiten können.

**Tabelle 37: Implementations-technische Informationen über *procuration***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ		KLASSIFIKATION			
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 256										
procuration extnID critical extnValue	SEQUENCE { { 1 3 36 8 3 2 }, FALSE, OCTET STRING {	30 2D 06 05 2B 24 08 03 02 04 24	47										
ProcurationSyn signingFor type-id value surAndGivenN surName	SEQUENCE OF { [0] SEQUENCE { { 1 3 36 8 4 1 }, [0] EXPLICIT SEQ { SEQUENCE { "Name",	30 22 80 20 30 1E 06 05 2B 24 08 04 01 A0 15 30 13 30 11 13 04 4E 61 6D 65											

givenName	SEQ OF {"Vorname"} }}}}}	30 09 13 07 56 6F 73 6E...																		
-----------	-----------------------------	-------------------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

### Z 2.3.9.15.5. ZULASSUNG

#### Zweck

Die ISIS-spezifische, private Erweiterung *admission* mit der Syntax *AdmissionSyntax* dient zur Anzeige von Zulassungen, wie beispielsweise einer berufsrechtlichen Zulassung. Sie wird durch den Objektbezeichner *id-isis-at-admission* referenziert.

Durch die relativ komplexe Struktur von *AdmissionSyntax* werden folgende Konzepte und Anforderungen berücksichtigt:

- Angabe externer Stellen (wie Berufsverbände, Kammern, Vereinigungen, Behörden, Firmen usw.), die für die Überprüfung der Inhalte berufsrechtlicher Zulassungsinformationen verantwortlich sind, durch die Komponente *admissionAuthority*.
- Angabe von Namensinstanzen, die für die Verwaltung von sog. Code- bzw. Berufsbezeichnungslisten verantwortlich sind, durch die Komponente *namingAuthority*. Durch unterschiedliche Codelisten können Hierarchien hinsichtlich der Berufe, Spezialisierungen, Disziplinen, Tätigkeitsfelder usw. ausgedrückt werden.
- Eindeutige Identifizierung bestimmter Berufe, Spezialisierungen, Disziplinen, Tätigkeitsfelder usw. durch die Komponente *professionItems*, die entweder eine Berufsbezeichnung oder genau einen oder mehrere Werte aus einer zugehörigen Codeliste enthält. Die Komponente *addProfessionInfo* dient zur Anzeige zusätzlicher Berufsinformationen.
- Unterstützung der automatischen oder manuellen Auswertbarkeit der Erweiterung durch die Komponente *namingAuthority*, die als Folge dreier Teilkomponenten definiert ist. Durch die Teilkomponenten *namingAuthorityId*, *namingAuthorityUrl* und *namingAuthorityText* werden ein Objektbezeichner zur Identifizierung der verantwortlichen Namensinstanz, eine URL zur Lokalisierung der Codeliste und ein Textstring benutzt, der beispielsweise die Stelle, das Land und den Codelistennamen enthalten kann. Das Verfahren für die Vergabe von neuen Objektbezeichnern, zum Beispiel für die Komponente *namingAuthorityId*, ist in Kapitel Z.2.3.9.15.11 beschrieben.
- Angabe von Zulassungsinformationen ohne eine Beteiligung von externen Stellen und Namensinstanzen durch die alleinige Benutzung der Komponente *professionItems*. In diesem Fall führt die Zertifizierungsstelle die Überprüfung der Zulassungsinformation selbst durch.

#### ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

```

```

admission          EXTENSION ::= {
    SYNTAX              AdmissionSyntax
    IDENTIFIED BY       id-isis-at-admission }
id-isis-at-admission OBJECT IDENTIFIER ::= { 1 3 36 8 3 3 }
id-isis-at-namingAuthorities OBJECT IDENTIFIER ::=
    { 1 3 36 8 3 11 }
AdmissionSyntax   ::= SEQUENCE {
    admissionAuthority  GeneralName OPTIONAL,
    contentsOfAdmissions SEQUENCE OF Admissions }
Admissions       ::= SEQUENCE {
    admissionAuthority  [0] GeneralName OPTIONAL,
    namingAuthority     [1] NamingAuthority OPTIONAL,
    professionInfos     SEQUENCE OF ProfessionInfo }
NamingAuthority  ::= SEQUENCE {
    namingAuthorityId   OBJECT IDENTIFIER OPTIONAL,
    namingAuthorityUrl  IA5String OPTIONAL,
    namingAuthorityText DirectoryString OPTIONAL }
ProfessionInfo   ::= SEQUENCE {
    namingAuthority     [0] NamingAuthority OPTIONAL,
    professionItems     SEQUENCE OF DirectoryString
    registrationNumber PrintableString OPTIONAL,
    addProfessionInfo  OCTET STRING OPTIONAL }
DirectoryString  ::= CHOICE {
    printableString     PrintableString (SIZE (1..maxSize))
    teletexString       TeletexString (SIZE (1..maxSize))
    bmpString           BMPString (SIZE (1..maxSize))
    universalString     UniversalString (SIZE (1..maxSize)) }

```

### Statische Semantik

Die Komponente *admissionAuthority* innerhalb von *AdmissionSyntax* dient als Voreinstellung für die Komponente *admissionAuthority* innerhalb von *Admissions*. Durch letztere kann die Voreinstellung überschrieben werden, falls es sich hierbei um eine andere verantwortliche Stelle handelt.

Die Komponente *namingAuthority* innerhalb von *Admissions* dient als Voreinstellung für die Komponente *namingAuthority* innerhalb von *ProfessionInfo*. Durch letztere kann die Voreinstellung überschrieben werden, falls es sich hierbei um eine andere Namensinstanz handelt.

Für automatisch verarbeitbare Berufsinformationen muß mindestens eine der optionalen Komponenten *namingAuthorityId* oder *namingAuthorityUrl* vorhanden sein. Insbesondere kann diese Kombination in Anwendungen zur eindeutigen Erkennung bestimmter Berufe verwendet werden. Die optionale Komponente *namingAuthorityText* kann in diesem Fall noch einen entsprechenden Anzeigetext enthalten.

Falls die beiden optionalen Komponenten *namingAuthorityId* und *namingAuthorityUrl* fehlen und nur die optionale Komponente *namingAuthorityText* vorhanden ist, so sind in

diesem Fall die Berufsinformationen nur manuell verarbeitbar und dienen lediglich zur Anzeige.

### ISIS-Konformitätsanforderungen

Die Benutzung der privaten Erweiterung *admission* durch Zertifizierungsstellen bei der Erzeugung von Zertifikaten ist optional und dabei als *non-critical* zu markieren. Der Systemparameter *maxSize* wird für *AdmissionSyntax* auf den Wert 128 festgelegt. ISIS-konforme Systeme und Anwendungen müssen die private Erweiterung *admission* erkennen und verarbeiten können.

**Tabelle 38: Implementations-technische Informationen über *admission***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ		KLASSIFIKATION		
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 256									
admission extnID critical extnValue AdmissionSynt	SEQUENCE { { 1 3 36 8 3 3 }, FALSE, OCTET STRING	<b>30 21</b> <b>06 05 2B 24 08 03 03</b> <b>04 18</b>	35									
contentsOfAdm Admissions ProfessionInfos ProfessionInfo professionItems	SEQUENCE OF { SEQUENCE { SEQUENCE OF { SEQUENCE { SEQ OF {"Dipl.- Phys" } } } } }	<b>30 14</b> <b>30 12</b> <b>30 10</b> <b>30 0E</b> <b>30 0C 13 0A 44 69 70</b> <b>6C 2E 2D 50 68 79 73</b>										

### Z 2.3.9.15.6. MONETÄRE BESCHRÄNKUNG

#### Zweck

Die ISIS-spezifische, private Erweiterung *monetaryLimit* dient zur Anzeige einer monetären Beschränkung. Sie wird durch den Objektbezeichner *id-isis-at-monetaryLimit* referenziert. Außer den Komponenten *amount* und *exponent*, aus denen sich der Beschränkungswert gemäß  $amount \cdot 10^{\text{exponent}}$  ergibt, muß die Währung im Teilfeld *currency* angegeben werden. Die folgende Tabelle enthält eine Übersicht einiger internationalen Währungen und deren zugehörige Abkürzungen.

ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

monetaryLimit EXTENSION ::= {
    SYNTAX          MonetaryLimitSyntax
    IDENTIFIED BY   id-isis-at-monetaryLimit }

id-isis-at-monetaryLimit OBJECT IDENTIFIER ::= {1 3 36 8 3 4}

MonetaryLimitSyntax ::= SEQUENCE {
    currency        PrintableString (SIZE(3)),
    amount          INTEGER,
    exponent        INTEGER }

```

ISIS-Konformitätsanforderungen

Die Benutzung der privaten Erweiterung *monetaryLimit* durch Zertifizierungsstellen bei der Erzeugung von Zertifikaten ist optional und muß als *non-critical* markiert werden. Die Länge der Wertefelder von *amount* und *exponent* ist auf 1 Byte begrenzt. Beispiele für zulässige Werte für *currency* können aus der folgenden Tabelle entnommen werden. ISIS-konforme Systeme und Anwendungen müssen die private Erweiterung *monetaryLimit* erkennen und verarbeiten können.

Eine Nutzung dieser Extension für die Angabe einer Haftungsbeschränkung der Zertifizierungsstelle ist nicht vorgesehen. Angaben über Haftungsbeschränkungen einer Zertifizierungsstelle sollten Gegenstand der Allgemeinen Geschäftsbedingungen (AGB) einer Zertifizierungsstelle sein. Die Aufnahme einer geeigneten Erweiterung in ISIS zur Angabe der Haftungsbeschränkung der Zertifizierungsstelle ist für eine Folgeversion von ISIS vorgesehen.

**Tabelle 39: Beispiele für internationale Währungen für das Feld *currencies***

LAND	WÄHRUNG		LAND	WÄHRUNG		LAND	WÄHRUNG	
	ABK.	HEX-CODE		ABK.	HEX-CODE		ABK.	HEX-CODE
Deutschland	DEM	13 03 44 45 4D	Europa	EUR	13 03 45 55 52	USA	USD	13 03 55 53 44

**Tabelle 40: Implementations-technische Informationen über *monetaryLimit***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFI- KATSTYP				RELE- VANZ		KLASSIFI- KATION			
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 24										
MonetaryLimit ExtnID	SEQUENCE { { 1 3 36 8 3 4 },	30 16 06 05 2B 24 08	24										
Critical ExtnValue	FALSE OCTET STRING	03 04 04 0D											
Currency Amount Exponent	SEQUENCE { "DEM", 1, 4 } }	30 0B 13 03 44 45 4D 02 01 01 02 01 04											

**Z 2.3.9.15.7. VOLLJÄHRIGKEIT**Zweck

Die ISIS-spezifische, private Erweiterung *declarationOfMajority* dient zur Anzeige der Volljährigkeit eines Teilnehmers. Sie wird durch den Objektbezeichner *id-isis-at-declaration-OfMajority* referenziert und ist als ein Auswahltyp der Typnamen *notYoungerThan*, *fullAgeAtCountry* und *dateOfBirth* definiert. Die erste Variante *notYoungerThan* zeigt ein Mindestalter an. Die zweite Variante *fullAgeAtCountry* dient zur Anzeige der Volljährigkeit eines Teilnehmers für ein bestimmtes Land. Diese Variante enthält die Teilkomponente *fullAge*, die anzeigt, ob ein Zertifikatsinhaber volljährig ist, sowie die Teilkomponente *country*, die das Land anzeigt, nach dessen Gesetz die Volljährigkeit zu beachten ist (nach Art. 7 EGBGB richtet sich dies nach der Staatsangehörigkeit des Betroffenen). Die dritte Variante *dateOfBirth* enthält das Geburtsdatum des Teilnehmers.

ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue      OCTET STRING }

```

```

declarationOfMajority EXTENSION ::= {

```

```
SYNTAX          DeclarationOfMajoritySyntax
IDENTIFIED BY   id-isis-at-declarationOfMajority }
```

```
id-isis-at-declarationOfMajority OBJECT IDENTIFIER
 ::= { 1 3 36 8 3 5 }
```

```
DeclarationOfMajoritySyntax ::= CHOICE {
  notYoungerThan      [0] IMPLICIT INTEGER,
  fullAgeAtCountry    [1] IMPLICIT SEQUENCE {
    fullAge            BOOLEAN DEFAULT TRUE,
    country            PrintableString (SIZE(2))
  }
  dateOfBirth         [2] GeneralizedTime }
```

### Statische Semantik

Im Feld *notYoungerThan* können beliebige Grenzwerte festgelegt werden. Bei der Kodierung von *dateOfBirth* ist das im Abschnitt Z.2.3.9.5 beschriebene Format YYYYMMDD zu beachten.

### ISIS-Konformitätsanforderungen

Die Benutzung der privaten Erweiterung *declarationOfMajority* durch Zertifizierungsstellen bei der Erzeugung von Zertifikaten ist optional und dabei als *non-critical* zu markieren. ISIS-konforme Systeme und Anwendungen müssen die private Erweiterung *declarationOfMajority* erkennen und verarbeiten können.

**Tabelle 41: Implementations-technische Informationen über *declarationOfMajority***

BEZEICHNER	WERTEBEREICH EINZELWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP							RELEVANZ				KLASSIFIKATION					
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung						
	(BEISPIELE)	(BEISPIELE)	[BYTES] 23																	
decl.OfMaj. extnID critical extnValue notYo.Than	SEQUENCE { { 1 3 36 8 3 5 }, FALSE, OCTETSTRING [0] IMPLICIT 18 }	30 0C 06 05 2B 24 08 03 05 04 03 80 01 12	14																	

### Z.2.3.9.15.8. CHIPKARTEN-SERIENNUMMER

#### Zweck

Das Signaturgesetz und die Signaturverordnung weisen der Signaturkomponente eine wesentliche Rolle innerhalb einer öffentlichen Sicherheitsinfrastruktur zu. Signaturzertifikate sollten aus diesem Grund eine Verbindung zwischen der Signaturkomponente (z.B. ICC,

integrated circuit card) und dem Signierenden herstellen. Die ISIS-spezifische, private Erweiterung *iCCSN* (integrated circuit card serial number) dient zur Anzeige der Seriennummer im Chipkartenbereich [DIN SigG/V 98]. Sie wird durch den Objektbezeichner *id-isis-at-ICCSN* referenziert. Somit kann nach einer Authentisierung der ICC die Seriennummer ICCSN der Chipkarte mit dem Inhalt der privaten Erweiterung *iCCSN* aus dem Signaturzertifikat überprüft werden. Voraussetzung für die Erzeugung dieses Feldes ist, daß bei der Personalisierung der Chipkarte diese Information der Zertifizierungsstelle z.B. in Form des ICC-Authentisierungszertifikates zur Überprüfung vorgelegen hat.

### ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

iCCSN        EXTENSION ::= {
    SYNTAX          ICCSNSyntax
    IDENTIFIED BY   id-isis-at-ICCSN }

id-isis-at-ICCSN OBJECT IDENTIFIER ::= { 1 3 36 8 3 6 }

ICCSNSyntax ::= IMPLICIT OCTETSTRING (SIZE(8..12))
  
```

### Statische Semantik

Der Oktettstring ist gemäß [DIN SigG/V 98, Abb. 8] zu kodieren.

### ISIS-Konformitätsanforderungen

Die Benutzung der privaten Erweiterung *iCCSN* durch Zertifizierungsstellen bei der Erzeugung von Zertifikaten ist optional und dabei als *non-critical* zu markieren. ISIS-konforme Systeme und Anwendungen im Chipkartenbereich müssen die private Erweiterung *iCCSN* erkennen und verarbeiten können.

**Tabelle 42: Implementations-technische Informationen über *iCCSN***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP								RELEVANZ	KLASSIFIKATION						
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung		private Erweiterung	critical-Markierung	non-critical-Markierung				
	(BEISPIELE)	(BEISPIELE)	[BYTES] 34																
iCCSN	SEQUENCE {	30 18	26																
extnID	{ 1 3 36 8 3 6 },	06 05 2B 24 08 03 06																	
critical	FALSE,																		
extnValue	OCTETSTRING }	04 0F XX ... XX																	

## Z 2.3.9.15.9. CHIPKARTEN-REFERENZIERUNG ÖFFENTLICHER SCHLÜSSEL

Zweck

So wie der öffentliche Schlüssel der Wurzelzertifizierungsstelle in einer Chipkarten-Betriebssystem eigenen Datei gespeichert wird und zur Verfügung steht, können auch weitere öffentliche Schlüssel von Zertifizierungsstellen aus Zertifikaten extrahiert und in einer Chipkarten-Betriebssystem eigenen Datei gespeichert werden. Diese öffentliche Schlüssel können bei der Verifikation einer Signatur als "Sicherheitsanker" benutzt werden.

Zu diesem Zweck wurde die private Erweiterung *pKReference* definiert, die das Acronym des Zertifikatausstellers (z.B. DEPCA für RegTP) in Verbindung mit der Seriennummer des Zertifikats enthalten muß. Der Name des Zertifikaterstellers und die Seriennummer sind hierbei Angaben aus dem direkt übergeordneten Zertifikat im Zertifizierungspfad, da sie zur Identifikation des Zertifikats der Zertifizierungsstelle dienen. Sie wird durch den Objektbezeichner *id-isis-at-pkReference* referenziert.

Zur Verifikation einer digitalen Signatur genügt es nicht, diese kryptographisch mit einem "Sicherheitsanker" zu verifizieren, sondern es wird das komplette Zertifikat benötigt, da das Zertifikat einer Zertifizierungsstelle ebenfalls wichtige Informationen oder Restriktionen beinhalten kann, wie beispielsweise eine Haftungsgrenze.

ASN.1 Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

pKReference      EXTENSION ::= {
    SYNTAX          PKReferenceSyntax
    IDENTIFIED BY   id-isis-at-pKReference }

id-isis-at-pKReference OBJECT IDENTIFIER
    ::= { 1 3 36 8 3 7 }

PKReferenceSyntax ::= OCTETSTRING (SIZE(20))

```

Statische Semantik

1.-2. Byte:	2-Byte-Länderkennung "DE" für Deutschland
3.-5. Byte:	3-Byte-Acronym des Zertifikaterstellers, "PCA" für die RegTP
6.-20. Byte:	Seriennummer

ISIS-Konformitätsanforderungen

Die Benutzung der privaten Erweiterung *pKReference* durch Zertifizierungsstellen bei der Erzeugung von Zertifikaten ist optional und dabei als *non-critical* zu markieren. ISIS-konfor-

me Systeme und Anwendungen im Chipkartenbereich müssen die private Erweiterung *pK-Reference* erkennen und verarbeiten können.

**Tabelle 43: Implementations-technische Informationen über *pKReference***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFI- KATSTYP				RELE- VANZ		KLASSIFI- KATION		
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 33									
pKReference extnID critical extnValue	SEQUENCE { { 1 3 36 8 3 7 }, FALSE, OCT. OCT.STR. "DEPCA"1 }	30 11 06 05 2B 24 08 03 07 04 08 04 06 44 45 50 43 41 01	19									

#### Z 2.3.9.15.10. SONSTIGE EINSCHRÄNKUNGEN

##### Zweck

Die ISIS-spezifische, private Erweiterung *restriction* dient zur Anzeige von sonstigen Einschränkungen.

##### ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue      OCTET STRING }

```

```

restriction EXTENSION ::= {
    SYNTAX          RestrictionSyntax
    IDENTIFIED BY   id-isis-at-restriction }

```

```

id-isis-at-restriction OBJECT IDENTIFIER ::= { 1 3 36 8 3 8 }

```

```

RestrictionSyntax ::= DirectoryString

```

```

DirectoryString ::= CHOICE {
    printableString PrintableString (SIZE (1..maxSize))
    teletexString   TeletexString (SIZE (1..maxSize))
    bmpString       BMPString (SIZE (1..maxSize))
    universalString UniversalString (SIZE (1..maxSize)) }

```

##### ISIS-Konformitätsanforderungen

Die Benutzung der privaten Erweiterung *restriction* durch Zertifizierungsstellen bei der Erzeugung von Zertifikaten ist optional und dabei als *non-critical* zu markieren. Der System-

parameter *maxSize* wird für *RestrictionSyntax* auf den Wert 128 festgelegt. ISIS-konforme Systeme und Anwendungen müssen die private Erweiterung *restriction* erkennen und verarbeiten können.

**Tabelle 44: Implementations-technische Informationen über *restriction***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP							RELEVANZ	KLASSIFIKATION		
				Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional		Standarderweiterung	private Erweiterung	critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 141											
restriction extnID critical extnValue	SEQUENCE { { 1 3 36 8 3 8 }, FALSE, OCTET STRING "Sonstige Zulassungsinfo"}	30 22 06 05 2B 24 08 03 08 04 20 13 17 53 6F 6E 73 ...	36											

#### Z 2.3.9.15.11. VERGABE WEITERER PRIVATER ERWEITERUNGEN

Externe Stellen wie Berufsverbände, Kammern, Vereinigungen, Behörden, Firmen usw. die anwendungsspezifische Informationen in Zertifikaten benötigen, die durch die derzeit beschriebenen Zertifikatsfelder nicht abgedeckt sind, müssen entsprechende neue private Erweiterungen und/oder neue Attribute beantragen. Hierzu gehören der Vorschlag einer ASN.1-Struktur und eine Beschreibung von deren Semantik. Objektbezeichner werden bei der TeleTrusT geführt und müssen gesondert beantragt werden. ISIS-spezifische Objektbezeichner für Signaturgesetz-Interoperabilität sind unter dem Objektbezeichnerzweig *id-isis* (1.3.36.8) festgelegt. Unterhalb von *id-isis* gibt es bisher die Zweige *id-isis-cp* (1.3.36.8.1) für Zertifizierungsrichtlinien, *id-isis-kp* (1.3.36.8.2) für Schlüsselnutzungsarten, sowie *id-isis-at* (1.3.36.8.3) für private Erweiterungen und Attribute.

## Z 3. ANHANG <sup>2</sup> NAMENSKONVENTIONEN

Namen werden in Zertifikaten an sehr vielen Stellen zu unterschiedlichen Zwecken benutzt und können dabei unterschiedlichen Formaten unterworfen sein. Im Rahmen der ISIS-Spezifikation spielt daher die Festlegung der Namenskonventionen eine wichtige Rolle. Die folgende Tabelle gibt eine Übersicht über das Vorkommen und die Benutzung von Namen in Zertifikaten. In der ersten Spalte werden die technischen Namen der relevanten Zertifikatskomponenten aufgeführt. Die zweite Spalte enthält Verweise auf Abschnitte dieses Dokumentes, in denen die einzelnen Komponenten beschrieben sind. Die Bedeutung der Komponenten wird in der dritten Spalte aufgeführt.

**Tabelle 50: Benutzung von Namen in Zertifikaten**

KOMPONENTE	REFERENZ	BEDEUTUNG
issuer	Z 2.3.4	Identifizierung einer Zertifizierungsstelle
subject	Z 2.3.6	Identifizierung eines Zertifikatsinhabers
subjectAltName	Z 2.3.9.5	Zusätzliche alternative Namen von Zertifikatsinhabern
issuerAltName	Z 2.3.9.6	Zusätzliche alternative Namen von Zertifizierungsstellen

### Z 3.1. Anhang <sup>2.1</sup> Eindeutige Identifizierung von Signaturschlüsselinhabern

Konform zum Signaturgesetz soll der Name in einem Zertifikat eine Person identifizieren, d.h. der Name muß die Person praktisch eindeutig benennen. Es reicht nicht, daß eine Person über weitere Angaben im Zertifikat identifiziert werden kann, da die Bindung der digitalen Signatur an einen Namen und einer damit verbundenen Identität entscheidend ist.

Das Signaturgesetz verlangt nun eine eindeutige Identifizierung des Signaturschlüssel-Inhabers. Es wird dabei genügen, eine "praktische" Eindeutigkeit zu fordern, wie sie auch jetzt schon im Rechtsverkehr üblich ist. Adressat eines solchen Namens ist die verifizierende Person, die in die Lage versetzt werden soll, über diesen Namen Vertrauen in die geleistete Unterschrift zu haben. Daraus folgt, daß dieser Name sprechend im Sinne einer Verwendung durch Personen sein sollte und daß dieser Name diejenigen Daten enthalten sollte, die im jeweiligen Anwendungskontext als üblich erachtet werden.

Eine Regelung dieser Art schließt nicht aus, daß ein Signaturschlüssel-Inhaber verschiedene Namen führen kann. Diese sind dann in der Regel abhängig von den verschiedenen Rollen des Signaturschlüssel-Inhabers. Eine Privatperson könnte ein Zertifikat für die Verwendung zu Kommunikation mit Behörden besitzen. Dieses würde die Person als Staatsbürger identifizieren. Dieselbe Person könnte ein weiteres Zertifikat besitzen, welches sie für private Geschäftszwecke verwendet. Dieses Zertifikat könnte beispielsweise die postalische Adresse beinhalten. Daneben sind ebenfalls Zertifikate denkbar, die eine Verbindung zu einem bestimmten Arbeitgeber bestätigen und für Erklärungen im Namen des Unternehmens verwen-

det wird. Ähnliches gilt für Zertifikate, die von Standesvereinigungen als Bestätigung der Mitgliedschaft ausgestellt werden.

### **Z 3.2. Anhang 2.2 Pseudonymisierung von Signaturschlüsselinhavern**

Da Zertifikate gemäß Signaturgesetz nur für natürliche Personen ausgestellt werden können, sind auch Zertifikate von Zertifizierungsstellen, Verzeichnisdiensten und Zeitstempeldiensten an natürliche Personen gebunden. Anstelle eines Namens kann jedes Signaturschlüssel-Zertifikat aber auch ein unverwechselbares Pseudonym enthalten, das dem Signaturschlüssel-Inhaber eindeutig zugeordnet ist. Diese Regelung gilt für Endbenutzer, den Zeitstempeldienst, den Verzeichnisdienst und für Zertifizierungsstellen gleichermaßen.

Um Pseudonyme als solche kenntlich machen zu können, wird im Rahmen der ISIS-Spezifikation das *subject*-Namensfeld (siehe Kapitel Z 2.3.6) verwendet, das entweder ein Pseudonym oder den gesetzlichen Namen aufnehmen kann. Um Pseudonyme mehrfach vergeben zu können, muß als weiteres Attribut *serialNumber* dem Namen hinzugefügt werden. Hierdurch wird die Eindeutigkeit der Pseudonyme gewährleistet.

### **Z 3.3. Anhang 2.3 Benutzung von Distinguished Names**

Formal besteht der technische Name eines Signaturschlüssel-Inhabers aus einer Sequenz von Attributen. Jedes Attribut besteht aus einer lesbaren Zeichenkette zusammen mit einer expliziten Typbezeichnung. Dies ermöglicht in vielen Fällen eine leichtere automatische Verarbeitung von Bezeichnungen, beispielsweise auch bei der Darstellung der Namen von Signaturschlüssel-Inhabern durch konforme Software.

Besonders geeignet für eine Namenvergabe ist der X.500 *Distinguished Name*, da dieser bereits eine große Auswahl an Typenbezeichnungen bietet. Die verwendeten Attribute sollten nur auf die Person und ihre Rolle bezogen sein und keine weitere Angaben über die Zertifizierungsstelle oder Zertifikatsmerkmale, wie beispielsweise die Seriennummer, enthalten.

Sofern eine Zertifizierungsstelle X.500 *Distinguished Names* zur technischen Identifikation von Signaturschlüssel-Inhabern verwenden will, sollten diese aus den oben angegebenen Attributen zusammengesetzt sein. Die so konstruierten *Distinguished Names* können (müssen aber nicht) in einem X.500-Verzeichnisdienst existieren. Die primäre Verwendung *der Distinguished Names* ist die Angabe von Attributen, die eine Person für eine bestimmte Anwendung technisch eindeutig identifiziert.

Eine Zertifizierungsstelle kann auch extern vorgebene *Distinguished Names* verwenden. Dies kann beispielsweise der Fall sein, wenn eine externe Stelle (Kammer, Behörde, Unternehmen, usw.) die Rolle einer Registrierungsstelle ausführt und Zertifikate für Mitarbeiter bei einer Zertifizierungsstelle bezieht. In einem solchen Fall muß sich die Zertifizierungsstelle

davon überzeugen, daß die im angegebenen *Distinguished Name* vorgegebenen Attribute den Signaturschlüssel-Inhaber korrekt identifizieren.

Die vorgeschlagenen Typen, ihre Kurzbezeichnungen und Objektbezeichner finden sich in folgenden Tabelle.

**Tabelle 51: Attribute in *Distinguished Name*-Typen**

OBJEKTBEZEICHNER	KURZ-FORM	BEDEUTUNG DES ATTRIBUTES, BEISPIELE
COUNTRY	(C)	Bezeichnung des Landes nach ISO-3166, im Rahmen des Signaturgesetzes sollten Namen immer einen Hinweis auf das Land enthalten, beispielsweise C=DE .
ORGANIZATION	(O)	Bezeichnung eines Unternehmens, sollte der üblichen Bezeichnung des Unternehmens im externen Sprachgebrauch entsprechen.
ORGANI-ZATIONAL UNIT	(OU)	Bezeichnung einer untergeordneten Organisationseinheit oder Abteilung innerhalb eines Unternehmens. Sollte nicht zur Unterscheidung von örtlich getrennten Niederlassungen verwendet werden.
COMMON NAME	(CN)	Die übliche Bezeichnung einer Person. Hier sollte der Name der Person so eingetragen sein, wie er im Personalausweis steht.
SURNAME	(S)	Nachname einer Person. Dieses Attribut kann zu besseren Unterscheidung bei unterschiedlichen Konventionen verwendet werden. Beispiel: CN=Ludwig van Beethoven, S=Beethoven
PSEUDONYM	PN	Pseudonym einer Person. PN=Rumpelstilzchen
LOCALITY	(L)	Angabe eines geographischen Ortes. Hier werden beispielsweise Städte, Gemeinden oder Bundesländer beschrieben. L=Darmstadt   L=64283 Darmstadt   L=Darmstadt 64283
STREET ADDRESS	(ST)	Angabe einer Straße als Teil einer postalischen Adresse ST=Dolivostr. 15   ST=Dolivostraße 15
TITLE	(T)	Angabe eines Titels: T= Prof.   T=Dr..   T=Graf
SERIAL NUMBER	(SER)	Angabe einer Seriennummer, sofern zwischen verschiedenen Personen unterschieden werden muß. Kann auch zur Beschreibung eines Geburtsdatums gebraucht werden. CN=Vorname Name, SER=7, O=Organisation, C=DE
STATE OR PROVINCE	(SP)	Präziser zur Angabe von Bundesländern, wenn nicht <i>Locality</i> verwendet wird SP=HESSEN
EMAIL ADDRESS	(EMAIL)	Sofern dieses Attribut im Distinguished Name kodiert werden soll. Angabe einer electronic Mail Adresse im Format des RFC 822 ( user@domain). Die Mailadresse sollte mit gleichem Inhalt auch in den alternativeName angegeben werden.

OBJEKTBEZEICHNER	KURZ-FORM	BEDEUTUNG DES ATTRIBUTES, BEISPIELE
DOMAIN COMPONENT	(DC)	Definition eines <i>DomainName</i> -Teilnamens CN=Vorname Name, EMAIL=user@orgunit.org.de, DC=org, DC=de Ermöglicht die Kodierung von DNS-Namen als Teil von <i>Distinguished Names</i> [RFC 2247 98]

### Z 3.4. Anhang 2.4 Benutzung von E-Mail-Adressen

Namen im Format für E-Mail-Adressen im Internet sind global eindeutig. Sie werden vom Betreiber eines bestimmten Mailsystems vergeben. Da E-Mail-Adressen auch nicht-sprechend (beispielsweise u1123@ccso.uiuc.edu) sein können, sollte ein normal lesbarer Name beigefügt werden, der wiederum ein Name oder ein Pseudonym sein kann. Bei der Anzeige sollte neben der Mailadresse auch der Name angezeigt werden.

Analog zu den extern vergebenen *Distinguished Names* muß der Signaturschlüsselinhaber hier bei der Registrierung den Nachweis erbringen, daß er der legitime Benutzer dieser E-Mail Adresse ist.

E-Mail-Adressen können entweder mit einer Zuweisung EMAIL=Attribut im *Distinguished Name* oder als *rfc822name* in den alternativen Namen festgelegt werden. Sind beide vorhanden (etwa aus Kompatibilitätsgründen für Software, die alternative Namen in Zertifikaten nicht auswerten kann und diese Informationen immer aus dem *subject*-Feld liest), so müssen die beiden Mailadressen übereinstimmen. Der Spezifikation der Mailadresse im alternativen Namen ist – in Anlehnung an PKIX – der Vorzug zu geben.

### Z 3.5. Anhang 2.5 X.500 Directory Distinguished Names

Der X.509 Standard für Zertifikate stammt ursprünglich aus der Normierungsarbeit für den weltweiten Directorystandard X.500. Die in X.509v3 spezifizierten *Distinguished Names* sind ihrem Ursprung nach Namen innerhalb dieses Verzeichnisdienstes. Durch die Verwendung dieses Zertifikatformats auch außerhalb des X.500 Kontextes wurde diese Vorgabe obsolet. In vielen Anwendungen wird der *Distinguished Name* als eine geordnete Folge von Attributen verarbeitet und nicht für Zugriffe auf ein X.500 Verzeichnis verwendet.

Im Rahmen des ISIS-Profiles sollte der im Zertifikat angegebene *Distinguished Name* nicht für Zugriffe auf einen Verzeichnisdienst verwendet werden. Wenn eine Zertifizierungsstelle explizit einen X.500-Verzeichnisdienstnamen in einem Zertifikat angeben möchte, so sollte dieser in den *subjectAltName* als *directoryName* gehalten werden. Anwendungen, die diesen Namen vorfinden, können diesen als Namen innerhalb des X.500-Directory verwenden.

## Z 4. ANHANG <sup>2</sup>I ERMITTLUNG VON DIENSTADRESSEN VON ZERTIFIZIERUNGSSTELLEN

Zertifizierungsstellen bieten neben dem Zertifizierungssdienst einen Verzeichnisdienst und einen Zeitstempeldienst an. Daneben können auch noch weitere Dienste angeboten werden, etwa ein Dienst für den Abruf von Sperrlisten oder weitere Prüf- und Beglaubigungsdienste.

Zur Verwendung dieser Dienste sollte die Anwendungsinfrastruktur in der Lage sein, aus dem Namen einer Zertifizierungsstelle oder aus zusätzlichen Angaben in den Zertifikaten, die technische Adressen dieser Dienste automatisch zu ermitteln. Beispielsweise ist es bei der Prüfung einer Signatur mit Hilfe des Verzeichnisdienstes notwendig, die technische Adresse des Verzeichnisdienstes aus den Angaben im Signaturschlüssel-Zertifikat oder dem Zertifikat der Zertifizierungsstelle zu ermitteln, wenn diese Information nicht direkt im Zertifikat enthalten ist.

Zur Angabe von Dienstadressen existieren in PKIX private Erweiterungen für X.509 Zertifikate (*authorityInfoAccess*, siehe 2.3.9.15.1). Hier werden Dienstadressen der Zertifizierungsstelle explizit im Signaturschlüsselzertifikat des Signaturschlüssel-Inhabers abgelegt. Vorteil dieser Lösung ist die explizite Angabe der Adressen, so daß die Ermittlung der technischen Adressen auf Seite der Anwendung sehr einfach zu realisieren ist. Nachteil ist dabei, daß die Adressierung explizit erfolgt. Eine Veränderung der Adressen, beispielsweise bei der Übernahme einer Zertifizierungsstelle durch eine andere, kann nachträglich nicht ohne Sperrung und Neuerstellung aller betroffenen Zertifikate erfolgen. Ebenso ist die Einführung neuer Dienste nur schwer möglich.

Alternativ zur expliziten Angabe der Dienste und Dienstadresse im Zertifikat können die Dienstadressen aus den Namensangaben der Zertifizierungsstelle gebildet werden. Zum einen kann die Adresse eines Dienstes durch eine Kombination des üblichen Dienstnamens mit dem Namen der Zertifizierungsstelle gebildet werden. Damit ist es für Zertifizierungsstellen leicht möglich, neue Dienste anzubieten. Als zweite Möglichkeit bietet sich an, die angebotenen Dienste und deren Adressen in einem öffentlich zugänglichen Verzeichnis zu hinterlegen. Der Eintrag in diesem Verzeichnis wird durch den Namen der Zertifizierungsstelle identifiziert.

Im folgenden wird exemplarisch beschrieben, wie eine Anwendung die in den Zertifikaten angegebenen Felder zur Ermittlung von Dienstadressen verwenden kann.

In der privaten PKIX Erweiterung *authorityInfoAccess* werden Dienst und Dienstadresse explizit angegeben. Der Anwendung ist der Objektbezeichner des gewünschten Dienstes bekannt. Über diesen identifiziert sie den entsprechenden Eintrag und verwendet die dort angegebene Dienstadresse. Wenn eine Zertifizierungsstelle diese Methode verwendet, sollte sie der Entwicklung der PKIX Protokolle folgen und diese auch vollständig implementieren.

Aus dem Feld *issuerAltName* in der Ausprägung *dNSName* kann eine Anwendung die Adressen des Dienstes ermitteln, indem sie das unter [RFC 2052 96] beschriebene Verfahren anwendet. Dies ist die empfohlene Vorgehensweise, sofern eine Zertifizierungsstelle ihre Dienste über das Internet anbieten möchte.

Aus dem Feld *issuerAltName* in der Ausprägung *rfc822Name* kann eine Anwendung aus dem Teilnamen nach dem @-Zeichen auf den Domainnamen der Zertifizierungsstelle schließen und dann ebenfalls das unter [RFC2052 96] beschriebene Verfahren anwenden. Gegebenenfalls muß der Vorgang mit einer verkürzten Version des Namens wiederholt werden, sofern keine Dienstadresse zu ermitteln war. Eine alleinige Angabe des *rfc822Name* in einem Zertifikat wird nicht empfohlen, so daß das hier kurz beschriebene Verfahren nur als letzte Möglichkeit in Betracht kommen sollte.

Über das Feld *ipAddress* könnte eine Anwendung den direkten Kontakt zu einem Dienst an dieser Adresse versuchen. Diese Vorgehensweise wird nicht empfohlen.

Über die Angaben im Feld *directoryName* kann ein Eintrag im globalen X.500 Verzeichnisdienst ermittelt werden. In diesem Eintrag kann die Anwendung aus dem Attribut für den gewünschten Dienst die Dienstadresse ermitteln. Dies ist der empfohlene Weg für Anwendungen, sofern die Zertifizierungsstelle ihre Dienste und Dienstadressen im X.500 Verzeichnisdienst vorhalten möchte.

Über die Angaben im Feld *issuer* kann ebenfalls ein Eintrag im globalen X.500 Verzeichnisdienst ermittelt werden. Analog zum oben angegebenen Beispiel legt die Zertifizierungsstelle hier ihre Dienste und Dienstadressen ab. Der Verwendung des Feldes *directoryName* ist der Vorzug gegenüber dem Feld *issuer* zu geben. Aus Kompatibilitätsgründen kann es sinnvoll sein, beide Felder in den Zertifikaten anzugeben. Dann sollten über beide dort angegebenen Namen dieselben Informationen über Dienste abrufbar sein.

Bei allen diesen Methoden ist darauf zu achten, daß die Angaben der Dienste und Dienstadressen manipuliert werden können. X.509v3-konforme Anwendungen sollten vor der Verwendung der angegebenen Dienste sicherstellen, daß sie mit dem gewünschten Dienst mit der gewünschten Zertifizierungsstelle verbunden sind. Ebenfalls sind Vorkehrungen zu treffen, um bei fehlerhaften oder fehlenden Angaben den Benutzer der Anwendung zu informieren.

## Z 5. ANHANG <sup>2</sup>II OBJEKTBEZEICHNER

Die folgende Tabelle enthält eine Übersicht über alle Objektbezeichner der X.509v3-Zertifikate, die im Abschnitt Z benutzt wurden.

**Tabelle 52: Objektbezeichner**

OBJEKTBEZEICHNERNUMMERN	OBJEKTBEZEICHNERNAMEN	REFERENZ
0	itu-t	
2	administration	
262	bmpt	
1	telekom	
10	security	
12	?	
0	liabiltyLimitationFlag	Z 2.3.9.15.2
1	iso	
2	member-body	
840	data country code, USA	
10045	ansi-x9-62	
1	ecdsa-with-SHA1	Z 2.3.7
2	id-PublicKeyType	
1	id-ecPublicKey	Z 2.7.3
113549	rsadsi	
1	pkcs	
1	pkcs-1	
1	rsaEncryption	Z 2.3.7
5	sha1WithRSAEncryption	Z 2.3.7
3	identified-organization	
6	dod	
1	internet	
5	security	
5	mechanisms	
7	pkix	
1	id-pe, private extensions	
1	authorityInfoAccess	Z 2.3.9.15.1
2	id-qt, qualifier types	
1	cps	Z 2.3.9.4
2	unotice	Z 2.3.9.4
3	id-kp, key purposes	
8	timeStamping	Z 2.3.9.3
9	OCSPsigning	Z 2.3.9.3
9	id-pda, personal data attributes	Z 2.3.6
3	id-pda-pseudonym	2.3.6
48	id-ad, access description	
1	ocsp	Z 2.3.9.15.1
2	caIssuers	Z 2.3.9.15.1
14	OIW	
3	secsig	
2	algorithm	
11	rsaSignature	Z 2.3.7
12	dsa	Z 2.3.7
20	dsaCommon	Z 2.3.7
27	dsaWithSHA1	Z 2.1

## Fortsetzung von Tabelle 52: Objektbezeichner

OBJEKTBEZEICHNERNUMMERN	OBJEKTBEZEICHNERNAMEN	REFERENZ
1	teletrust	
3	algorithm	
36	signatureAlgorithm	
3	rsaSignature	
3	rsaSignatureWithsha1	Z 2.1
3	rsaSignatureWithripemd160	Z 2.1
1	ecsieSign	
1	ecsieSignWithshs1	Z 2.1
2	ecsieSignWithripemd160	Z 2.1
2	signatureScheme	
4	sigS_ISO9796-2	
2	sigS_ISO9796-2Withrsa	Z 2.1
2	sigS_ISO9796-2rnd	Z 2.1
8	id-isis	
1	id-isis-cp	
1	id-isis-cp-sigconform	Z 2.3.9.4
2	id-isis-kp	
1	id-isis-kp-directoryService	Z 2.3.9.3
3	id-isis-at	
1	id-isis-at-dateOfCertGen	Z 2.3.9.15.3
2	id-isis-at-procuration	Z 2.3.9.15.4
3	id-isis-at-admission	Z 2.3.9.15.5
4	id-isis-at-monetaryLimit	Z 2.3.9.15.6
5	id-isis-at-declarationOfMajority	Z 2.3.9.15.7
6	id-isis-at-iCCSN	Z 2.3.9.15.8
7	id-isis-at-pKReference	Z 2.3.9.15.9
8	id-isis-at-restriction	Z 2.3.9.15.10
9	id-isis-at-retrieveIfAllowed	Abschnitt V
10	id-isis-at-requestedCertificate	Abschnitt V
11	id-isis-at-namingAuthorities	Z 2.3.9.15.5
12	id-isis-at-certInDirSince	Abschnitt V
13	id-isis-at-certHash	Abschnitt V
2	joint-iso-ccitt	
5	ds	
4	attributeType	
3	commonName	Z 2.3.4
4	surName	Z 2.3.4
5	serialNumber	Z 2.3.4
6	countryName	Z 2.3.4
7	localityName	Z 2.3.4
8	stateOrProvinceName	Z 2.3.4
10	organizationName	Z 2.3.4
11	organizationalUnit	Z 2.3.4
12	title	Z 2.3.4
15	businessCategory	Z 2.3.4
17	postalCode	Z 2.3.4
47	givenName	Z 2.3.4

## Fortsetzung von Tabelle 52: Objektbezeichner

OBJEKTBEZEICHNERNUMMERN	OBJEKTBEZEICHNERNAMEN	REFERENZ
2	algorithm	
5	encryptionAlgorithm	
8	rsa	Z 2.3.7
	id-ce, certificate extensions	
1	subjectDirectoryAttributes	Z 2.3.9.11
	subjectKeyIdentifier	Z 2.3.9.8
1	keyUsage	Z 2.3.9.2
29	privateKeyUsagePeriod	Z 2.3.9.14
9	subjectAltName	Z 2.3.9.5
14	issuerAltName	Z 2.3.9.6
15	basicConstraints	Z 2.3.9.1
16	nameConstraints	Z 2.3.9.13
17	cRLDistributionPoints	Z 2.3.9.9
18	certificatePolicies	Z 2.3.9.4
19	policyMapping	Z 2.3.9.10
30	authorityKeyIdentifier	Z 2.3.9.7
31	policyConstraints	Z 2.3.9.12
32	extKeyUsage	Z 2.3.9.3
33		
35		
36		
37		

## Z 6. ANHANG IV ASN.1 DEFINITIONEN

Dieser Abschnitt enthält eine Zusammenfassung aller ASN.1-Definitionen in alphabetischer Reihenfolge, die im Abschnitt Z benutzt werden.

<b>AccessDescription</b>	<b>::= SEQUENCE</b> { accessMethod OBJECT IDENTIFIER, accessLocation GeneralName }
<b>admission</b>	<b>EXTENSION ::=</b> { SYNTAX AdmissionSyntax IDENTIFIED BY id-isis-at-admission }
<b>Admissions</b>	<b>::= SEQUENCE</b> { admissionAuthority [0] GeneralName OPTIONAL, namingAuthority [1] NamingAuthority OPTIONAL, professionInfos SEQUENCE OF ProfessionInfo }
<b>AdmissionSyntax</b>	<b>::= SEQUENCE</b> { admissionAuthority GeneralName OPTIONAL, contentsOfAdmissions SEQUENCE OF Admissions }
<b>Algorithmidentifier</b>	<b>::= SEQUENCE</b> { algorithm OBJECT IDENTIFIER, parameters ANY DEFINED BY algorithm OPTIONAL }
<b>atAdmission</b>	<b>ATTRIBUTE ::=</b> { WITH SYNTAX AdmissionSyntax SINGLE VALUE ID id-isis-at-admission }
<b>atDeclarationOfMajority</b>	<b>ATTRIBUTE ::=</b> { WITH SYNTAX DeclarationOfMajoritySyntax SINGLE VALUE ID id-isis-at-declarationOfMajority }
<b>atMonetaryLimit</b>	<b>ATTRIBUTE ::=</b> { WITH SYNTAX MonetaryLimitSyntax SINGLE VALUE ID id-isis-at-monetaryLimit }
<b>atProcuration</b>	<b>ATTRIBUTE ::=</b> { WITH SYNTAX ProcurationSyntax SINGLE VALUE ID id-isis-at-procuration }
<b>atRestriction</b>	<b>ATTRIBUTE ::=</b> { WITH SYNTAX RestrictionSyntax SINGLE VALUE ID id-isis-at-restriction }
<b>AttCertValidityPeriod</b>	<b>::= SEQUENCE</b> { notBeforeTime GeneralizedTime, notAfterTime GeneralizedTime }
<b>ATTRIBUTE</b>	<b>::= CLASS</b> { &id OBJECT IDENTIFIER UNIQUE, &Type }  WITH SYNTAX { SYNTAX &Type

IDENTIFIED BY	&id }
<b>Attribute</b>	::= SEQUENCE { type AttributeType, values SET OF AttributeValue
<b>AttributeCertificate</b>	::= SEQUENCE { tbsAttributeCertificate TBSAttributeCertificate, signatureAlgorithm AlgorithmIdentifier, signature BIT STRING }
<b>AttributesSyntax</b>	::= SEQUENCE SIZE (1..MAX) OF Attribute
<b>AttributeType</b>	::= ATTRIBUTE.&id
<b>AttributeType</b>	::= OBJECT IDENTIFIER
<b>AttributeTypeAndValue</b>	::= SEQUENCE { type AttributeType, value AttributeValue }
<b>AttributeValue</b>	::= ANY DEFINED BY AttributeType
<b>AttributeValue</b>	::= ATTRIBUTE.&Type
<b>authorityInfoAccess</b> EXTENSION	::= {  WITH SYNTAX { SYNTAX AuthorityInfoAccessSyntax IDENTIFIED BY id-ce-authorityInfoAccess }
<b>AuthorityInfoAccessSyntax</b>	::= SEQUENCE SIZE (1..MAX) OF AccessDescription
<b>authorityKeyIdentifier</b> EXTENSION	::= {  WITH SYNTAX { SYNTAX AuthorityKeyIdentifier IDENTIFIED BY id-ce-authorityKeyIdentifier }
<b>AuthorityKeyIdentifier</b>	::= SEQUENCE { keyIdentifier [0] KeyIdentifier OPTIONAL, authorityCertIssuer [1] GeneralNames OPTIONAL, authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL }
<b>BaseDistance</b>	::= INTEGER (0..MAX)
<b>basicConstraints</b> EXTENSION	::= {  WITH SYNTAX { SYNTAX BasicConstraintsSyntax IDENTIFIED BY id-ce-basicConstraints }
<b>BasicConstraintsSyntax</b>	::= SEQUENCE { cA BOOLEAN DEFAULT FALSE, pathLenConstraint INTEGER (0..MAX) OPTIONAL }
<b>Certificate</b>	::= SEQUENCE { tbsCertificate TBSCertificate, signatureAlgorithm AlgorithmIdentifier, signature BIT STRING }
<b>certificatePolicies</b> EXTENSION	::= {  WITH SYNTAX {

SYNTAX	CertificatePoliciesSyntax
IDENTIFIED BY	id-ce-extKeyUsage }
<b>CertificatePoliciesSyntax</b>	::= SEQUENCE SIZE (1..MAX) OF PolicyInformation
<b>CertificateSerialNumber</b>	::= INTEGER
<b>CertPolicyId</b>	::= OBJECT IDENTIFIER
<b>CRLDistPointsSyntax</b>	::= SEQUENCE SIZE (1..MAX) OF DistributionPoint
<b>cRLDistributionPoints</b> EXTENSION	::= { WITH SYNTAX { SYNTAX CRLDistPointsSyntax IDENTIFIED BY id-ce- cRLDistributionPoints }
<b>dateOfCertGen</b> EXTENSION	::= { SYNTAX DateOfCertGenSyntax IDENTIFIED BY id-isis-at-dateOfCertGen }
<b>DateOfCertGenSyntax</b>	::= GeneralizedTime
<b>declarationOfMajority</b> EXTENSION	::= { SYNTAX DeclarationOfMajoritySyntax IDENTIFIED BY id-isis-at-declarationOfMajority }
<b>DeclarationOfMajoritySyntax</b>	::= CHOICE { notYoungerThan [0] IMPLICIT INTEGER, fullAgeAtCountry [1] IMPLICIT SEQUENCE { fullAge BOOLEAN DEFAULT TRUE, country PrintableString (SIZE(2)) DEFAULT "DE" } dateOfBirth [2] GeneralizedTime }
<b>DirectoryString</b>	::= CHOICE { printableString PrintableString (SIZE (1..maxSize)) teletexString TeletexString (SIZE (1..maxSize)) bmpString BMPString (SIZE (1..maxSize)) universalString UniversalString (SIZE (1..maxSize)) }
<b>DistributionPoint</b>	::= SEQUENCE { distributionPoint [0] DistributionPointName OPTIONAL, reasons [1] ReasonFlags OPTIONAL, cRLIssuer [2] GeneralNames OPTIONAL }
<b>DistributionPointName</b>	::= CHOICE { fullName [0] GeneralNames, nameRelativeToCRLIssuer [1] RelativeDistinguishedName }
<b>dsa</b> ALGORITHM PARAMETER DSAParameters	::= {algorithm 12}
<b>dsaCommon</b> ALGORITHM PARAMETER NULL	::= {algorithm 20}
<b>DSAParameters</b>	::= SEQUENCE { prime1 INTEGER, prime2 INTEGER, base INTEGER }
<b>DSAPublicKey</b>	::= INTEGER
<b>ECDSAPublicKey</b>	::= OCTET STRING

<b>Ecdsa-SigValue</b>	::= SEQUENCE { r                  INTEGER, s                  INTEGER }
<b>EDIPartyName</b>	::= SEQUENCE { nameAssigner      [0] DirectoryString OPTIONAL, partyName         [1] DirectoryString }
<b>EXTENSION</b>	::= CLASS { &id               OBJECT IDENTIFIER UNIQUE, &ExtType }
<b>WITH SYNTAX</b> {	
SYNTAX	&ExtnType
IDENTIFIED BY	&id }
<b>Extension</b>	::= SEQUENCE { extnId            OBJECT IDENTIFIER, critical          BOOLEAN DEFAULT FALSE, extnValue         OCTET STRING }
<b>Extensions</b>	::= SEQUENCE (1..MAX) OF Extension
<b>extKeyUsage</b> EXTENSION	::= {
<b>WITH SYNTAX</b> {	
SYNTAX	ExtKeyUsageSyntax
IDENTIFIED BY	id-ce-extKeyUsage }
<b>ExtKeyUsageSyntax</b>	::= SEQUENCE SIZE (1..MAX) OF KeyPurposeId
<b>GeneralName</b>	::= CHOICE { otherName          [0] OTHER-NAME, rfc822Name         [1] IA5String, dNSName            [2] IA5String, x400Address        [3] ORAddress, directoryName      [4] Name, ediPartyName       [5] EDIPartyName, uniformResourceIdentifier[6] IA5String, iPAddress          [7] OCTET STRING, registeredID       [8] OBJECT IDENTIFIER }
<b>GeneralNames</b>	::= SEQUENCE SIZE (1..MAX) OF GeneralName
<b>GeneralSubtree</b>	::= SEQUENCE { base               GeneralName, minimum            [0] BaseDistance DEFAULT 0, maximum            [1] BaseDistance OPTIONAL }
<b>GeneralSubtrees</b>	::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree
<b>iCCSN</b> EXTENSION	::= {
SYNTAX	ICCSNSyntax
IDENTIFIED BY	id-isis-at-iCCSN }
<b>ICCSNSyntax</b>	::= IMPLICIT OCTETSTRING (SIZE(15..23))
<b>issuerAltName</b> EXTENSION	::= {
<b>WITH SYNTAX</b> {	
SYNTAX	IssuerAltName
IDENTIFIED BY	id-ce-issuerAltName }

<b>IssuerAndSerial</b>	::= SEQUENCE { issuer GeneralName serial CertificateSerialNumber }
<b>IssuerAltName</b>	::= GeneralNames
<b>IssuerSerial</b>	::= SEQUENCE { issuer GeneralNames, serial CertificateSerialNumber, issuerUID UniqueIdentifier OPTIONAL }
<b>KeyIdentifier</b>	::= OCTET STRING
<b>KeyPurposeId</b>	::= OBJECT IDENTIFIER
<b>KeySize</b>	::= INTEGER
<b>keyUsage EXTENSION</b>	::= { WITH SYNTAX { SYNTAX KeyUsage IDENTIFIED BY id-ce-keyUsage }
<b>KeyUsage</b>	::= BIT STRING { digitalSignature (0), nonRepudiation (1), keyEncipherment (2), dataEncipherment (3), keyAgreement (4), keyCertSign (5), cRLSign (6), encipherOnly (7), decipherOnly (8) }
<b>liabilityLimitationFlag EXTENSION</b>	::= { WITH SYNTAX { SYNTAX BOOLEAN DEFAULT FALSE IDENTIFIED BY certExtensionLiabilityLimitationFlag }
<b>monetaryLimit EXTENSION</b>	::= { SYNTAX MonetaryLimitSyntax IDENTIFIED BY id-isis-at-monetaryLimit }
<b>MonetaryLimitSyntax</b>	::= SEQUENCE { currency PrintableString (SIZE(3)), amount INTEGER, exponent INTEGER }
<b>Name</b>	::= CHOICE { RDNSequence }
<b>nameConstraints EXTENSION</b>	::= { WITH SYNTAX { SYNTAX NameConstraintsSyntax IDENTIFIED BY id-ce-nameConstraints }
<b>NameConstraintsSyntax</b>	::= SEQUENCE { permittedSubtrees [0] GeneralSubtrees OPTIONAL, excludedSubtrees [1] GeneralSubtrees OPTIONAL }
<b>NamingAuthority</b>	::= SEQUENCE {

namingAuthorityId	OBJECT IDENTIFIER OPTIONAL,
namingAuthorityUrl	IA5String OPTIONAL,
namingAuthorityText	DirectoryString OPTIONAL }
<b>OTHER-NAME</b>	::= SEQUENCE {
type-id	OBJECT IDENTIFIER,
value	[0] EXPLICIT ANY DEFINED BY type-id }
<b>pkReference</b>	EXTENSION ::= {
SYNTAX	PKReferenceSyntax
IDENTIFIED BY	id-isis-at-pkReference }
<b>PKReferenceSyntax</b>	::= OCTETSTRING (SIZE(20))
<b>policyConstraints</b>	EXTENSION ::= {
WITH SYNTAX {	
SYNTAX	PolicyConstraintsSyntax
IDENTIFIED BY	id-ce-policyConstraints }
<b>PolicyConstraints</b>	::= SEQUENCE {
requireExplicitPolicy	[0] SkipCerts OPTIONAL,
inhibitPolicyMapping	[1] SkipCerts OPTIONAL }
<b>PolicyConstraintsSyntax</b>	::=
SEQUENCE SIZE (1..MAX) OF PolicyConstraints	
<b>PolicyInformation</b>	::= SEQUENCE {
policyIdentifier	CertPolicyId,
policyQualifiers	SEQUENCE SIZE (1..MAX) OF PolicyQualifierInfo OPTIONAL }
<b>policyMappings</b>	EXTENSION ::= {
WITH SYNTAX {	
SYNTAX	PolicyMappingsSyntax
IDENTIFIED BY	id-ce-policyMappings }
<b>PolicyMappingsSyntax</b>	::= SEQUENCE SIZE (1..MAX) OF
SEQUENCE {	
issuerDomainPolicy	CertPolicyId,
subjectDomainPolicy	CertPolicyId }
<b>PolicyQualifierId</b>	::= OBJECT IDENTIFIER
<b>PolicyQualifierInfo</b>	::= SEQUENCE {
policyQualifierId	PolicyQualifierId,
qualifier	ANY DEFINED BY policyQualifierId }
<b>privateKeyUsagePeriod</b>	EXTENSION ::= {
WITH SYNTAX {	
SYNTAX	PrivateKeyUsagePeriod
IDENTIFIED BY	id-ce-privateKeyUsagePeriod }
<b>PrivateKeyUsagePeriod</b>	::= SEQUENCE {
notBefore	[0] GeneralizedTime OPTIONAL,
notAfter	[1] GeneralizedTime OPTIONAL }
<b>procuration</b>	EXTENSION ::= {
SYNTAX	ProcurationSyntax
IDENTIFIED BY	id-isis-at-procuration }

<b>ProcurationSyntax</b> ::= SEQUENCE OF { country PrintableString (SIZE(2)) OPTIONAL, typeOfSubstitution DirectoryString OPTIONAL, signingFor SigningFor }
<b>ProfessionInfo</b> ::= SEQUENCE OF { namingAuthority [0] NamingAuthority OPTIONAL, professionItems SEQUENCE OF DirectoryString, registrationNumber PrintableString OPTIONAL, addProfessionInfo OCTET STRING OPTIONAL }
<b>Pseudonym</b> ::= DirectoryString
<b>RDNSequence</b> ::= SEQUENCE OF RelativeDistinguishedName
<b>ReasonFlags</b> ::= BIT STRING { unused(0), keyCompromise(1), cACompromise(2), affiliationChanged(3), superseded(4), cessationOfOperation (5), certificateHold(6) }
<b>RelativeDistinguishedName</b> ::= SET OF AttributeTypeAndValue
<b>restriction</b> EXTENSION ::= { SYNTAX RrestrictionSyntax IDENTIFIED BY id-isis-at-restriction }
<b>RestrictionSyntax</b> ::= DirectoryString
<b>rsa</b> ALGORITHM PARAMETER KeySize ::= { encryptionAlgorithm 1 }
<b>rsaEncryption</b> ALGORITHM PARAMETER NULL ::= { pkcs-1 1 }
<b>RSAPublicKey</b> ::= SEQUENCE { modulus INTEGER, publicExponent INTEGER }
<b>rsaSignature</b> ALGORITHM PARAMETER NULL ::= { algorithm 11 }
<b>SigningFor</b> ::= CHOICE { thirdPerson GeneralName, cerRef IssuerAndSerial }
<b>SkipCerts</b> ::= INTEGER (0..MAX)
<b>subjectAltName</b> EXTENSION ::= { WITH SYNTAX { SYNTAX SubjectAltName IDENTIFIED BY id-ce-subjectAltName }
<b>SubjectAltName</b> ::= GeneralNames
<b>subjectDirectoryAttributes</b> EXTENSION ::= { WITH SYNTAX { SYNTAX AttributesSyntax IDENTIFIED BY id-ce-subjectDirectoryAttributes } }
<b>subjectKeyIdentifier</b> EXTENSION ::= {

WITH SYNTAX { SYNTAX SubjectKeyIdentifier IDENTIFIED BY id-ce-subjectKeyIdentifier }
<b>SubjectKeyIdentifier</b> ::= KeyIdentifier
<b>SubjectPublicKeyInfo</b> ::= SEQUENCE { algorithm AlgorithmIdentifier, subjectPublicKey BIT STRING }
<b>TBSAttributeCertificate</b> ::= SEQUENCE { version Version DEFAULT v1, subject CHOICE { baseCertificateID [0] IssuerSerial, subjectName [1] GeneralNames }, issuer GeneralNames, signature AlgorithmIdentifier, serialNumber CertificateSerialNumber, attrCertValidityPeriod AttCertValidityPeriod, attributes SEQUENCE OF Attribute, issuerUniqueID UniqueIdentifier OPTIONAL, extensions Extensions OPTIONAL }
<b>TBSCertificate</b> ::= SEQUENCE { version [0] EXPLICIT Version DEFAULT v1, serialNumber CertificateSerialNumber, signature AlgorithmIdentifier, issuer Name, validity Validity, subject Name, subjectPublicKeyInfo SubjectPublicKeyInfo, issuerUniqueID [1] IMPLICIT UniqueIdentifier OPTIONAL, subjectUniqueID [2] IMPLICIT UniqueIdentifier OPTIONAL, extensions [3] EXPLICIT Extensions Optional }
<b>Time</b> ::= CHOICE { utcTime UTCTime, generalizedTime GeneralizedTime }
<b>UniqueIdentifier</b> ::= BIT STRING
<b>Validity</b> ::= SEQUENCE { notBefore Time, notAfter Time }
<b>Version</b> ::= INTEGER { v1(0), v2(1), v3(2) }

Objektbezeichner

<b>algorithm</b> OBJECT IDENTIFIER ::= { 1 3 14 3 2 }
<b>ansi-x9-62</b> OBJECT IDENTIFIER ::= { 1 2 840 10045 }
<b>certExtensionLiabilityLimitationFlag</b> OBJECT IDENTIFIER ::= { 0 2 262 1 10 12 0 }
<b>certificateExtension</b> OBJECT IDENTIFIER ::= { 2 5 29 }
<b>ecamvSign</b> OBJECT IDENTIFIER ::= { 1 3 36 3 3 2 }
<b>ecdsa-with-sha1</b> OBJECT IDENTIFIER ::= { 1 2 840 10045 1 }

<b>encryptionAlgorithm</b> OBJECT IDENTIFIER ::= { 2 5 8 1 }
<b>id-ad</b> OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 48 }
<b>id-ad-caIssuers</b> OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 48 2 }
<b>id-ad-ocsp</b> OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 48 1 }
<b>id-ce</b> OBJECT IDENTIFIER ::= { 2 5 29 }
<b>id-ce-authorityKeyIdentifier</b> OBJECT IDENTIFIER ::= { 2 5 29 35 }
<b>id-ce-basicConstraints</b> OBJECT IDENTIFIER ::= { 2 5 29 19 }
<b>id-ce-certificatePolicies</b> OBJECT IDENTIFIER ::= { 2 5 29 32 }
<b>id-ce-cRLDistributionPoints</b> OBJECT IDENTIFIER ::= { 2 5 29 31 }
<b>id-ce-extKeyUsage</b> OBJECT IDENTIFIER ::= { 2 5 29 37 }
<b>id-ce-issuerAltName</b> OBJECT IDENTIFIER ::= { 2 5 29 18 }
<b>id-ce-keyUsage</b> OBJECT IDENTIFIER ::= { 2 5 29 15 }
<b>id-ce-nameConstraints</b> OBJECT IDENTIFIER ::= { 2 5 29 30 }
<b>id-ce-policyConstraints</b> OBJECT IDENTIFIER ::= { 2 5 29 36 }
<b>id-ce-policyMappings</b> OBJECT IDENTIFIER ::= { 2 5 29 33 }
<b>id-ce-privateKeyUsagePeriod</b> OBJECT IDENTIFIER ::= { 2 5 29 16 }
<b>id-ce-subjectAltName</b> OBJECT IDENTIFIER ::= { 2 5 29 17 }
<b>id-ce-subjectDirectoryAttributes</b> OBJECT IDENTIFIER ::= { 2 5 29 9 }
<b>id-ce-subjectKeyIdentifier</b> OBJECT IDENTIFIER ::= { 2 5 29 14 }
<b>id-ecPublicKey</b> OBJECT IDENTIFIER ::= { 1 2 840 10045 2 1 }
<b>id-kp</b> OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 3 }
<b>id-kp-time-Stamping</b> OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 3 7 }
<b>id-pe</b> OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 1 }
<b>id-pe-authorityInfoAccess</b> OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 1 1 }
<b>id-pda</b> OBJECT IDENTIFIER ::= { id-pkix 9 }
<b>id-pda-pseudonym</b> AttributeType ::= { id-pda 3 }
<b>id-pkix</b> OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 }
<b>id-pkix-kp</b> OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 3 }
<b>id-pkix-kp-time-Stamping</b> OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 3 8 }
<b>id-pkix-kp-OCSPSigning</b> OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 3 9 }
<b>id-publicKeyType</b> OBJECT IDENTIFIER ::= { 1 2 840 10045 2 }
<b>id-isis</b> OBJECT IDENTIFIER ::= { 1 3 36 8 }
<b>id-isis-at</b> OBJECT IDENTIFIER ::= { 1 3 36 8 3 }
<b>id-isis-at-admission</b> OBJECT IDENTIFIER ::= { 1 3 36 8 3 3 }

<b>id-isis-at-dateOfCertGen</b> OBJECT IDENTIFIER ::= { 1 3 36 8 3 1 }
<b>id-isis-at-declarationOfMajority</b> OBJECT IDENTIFIER ::= { 1 3 36 8 3 5 }
<b>id-isis-at-iCCSN</b> OBJECT IDENTIFIER ::= { 1 3 36 8 3 6 }
<b>id-isis-at-monetaryLimit</b> OBJECT IDENTIFIER ::= { 1 3 36 8 3 4 }
<b>id-isis-at-pKReference</b> OBJECT IDENTIFIER ::= { 1 3 36 8 3 7 }
<b>id-isis-at-procuration</b> OBJECT IDENTIFIER ::= { 1 3 36 8 3 2 }
<b>id-isis-at-restriction</b> OBJECT IDENTIFIER ::= { 1 3 36 8 3 8 }
<b>id-isis-kp</b> OBJECT IDENTIFIER ::= { 1 3 36 8 2 }
<b>id-isis-kp-directoryService</b> OBJECT IDENTIFIER ::= { 1 3 36 8 2 1 }
<b>pkcs-1</b> OBJECT IDENTIFIER ::= { 1 2 840 113549 1 1 1 }

# Abschnitt II: Verzeichnisdienste

## V 1. EINLEITUNG ZUM TEIL VERZEICHNISDIENSTE

### VORZEITIGE BEENDIGUNG DER GÜLTIGKEITSDAUER EINES ZERTIFIKATES

Das grundlegende Prinzip der Vertrauensbildung in öffentlichen Sicherheitsinfrastrukturen beruht auf dem Vertrauen in die von Zertifizierungsstellen ausgestellten und von ihnen signierten Zertifikate, durch die letztlich eine Bindung der zugehörigen öffentlichen Schlüssel an die jeweiligen Schlüsselinhaber erreicht wird.

Bei der Erstellung eines Zertifikates wird u.a. eine Gültigkeitsdauer für dieses Zertifikat festgelegt und als Bestandteil in das Zertifikat integriert. Der Zeitpunkt der Zertifikaterstellung muß dabei nicht mit dem Beginn der Gültigkeitsdauer übereinstimmen. Er kann durch die private und optionale Zertifikaterweiterung *dateOfCertGen* [Abschnitt Z 2.3.9.15.3] als Bestandteil des Zertifikates angegeben werden. Prinzipiell kann davon ausgegangen werden, daß ein Zertifikat während seiner gesamten Gültigkeitsdauer benutzt wird. Es gibt jedoch bestimmte Situationen, die eine vorzeitige Beendigung der Gültigkeitsdauer eines Zertifikates veranlassen bzw. erzwingen, d.h. das betreffende Zertifikat von der zuständigen Zertifizierungsstelle zu sperren ist. Die Ursachen und Gründe, die zu einer Sperrung eines Zertifikates führen, können unterschiedlicher Natur sein, wie z.B. bei den folgenden Situationen:

- Ein Teilnehmer benötigt sein Zertifikat während der restlich verbliebenen Gültigkeitsdauer nicht mehr und möchte seine Signaturkomponente unbrauchbar machen.
- Die Namensangaben eines Zertifikatsinhabers oder der zugehörigen Zertifizierungsstelle haben sich geändert und dadurch die Grundlage für die Bindung zwischen öffentlichen Schlüsseln und Personen entzogen.
- Es besteht der Verdacht oder der Nachweis einer Kompromittierung des privaten Schlüssels eines Zertifikatsinhabers oder der zugehörigen Zertifizierungsstelle und damit die Möglichkeit potentieller Angriffe.

### MINIMIERUNG DES RESTRISIKOS FÜR ANGRIFFE

Sobald eine mißbräuchliche Nutzung eines Signaturschlüssels nicht mehr ausgeschlossen werden kann (z.B. aufgrund des Verlusts oder Diebstahls der Signaturkomponente), muß mit möglichen Angriffen und damit mit einer Verletzung und Gefährdung der Sicherheit während der restlich verbliebenen Gültigkeitsdauer eines Zertifikates gerechnet werden. Für diese kritischen Situationen müssen entsprechende Vorkehrungen und Sicherheitsmaßnahmen vorgesehen werden, deren Ziel es ist, das Restrisiko für Angriffe und dadurch verursachte Gefährdung der Sicherheitsinfrastruktur zu minimieren.

In diesem Zusammenhang spielen die Themenbereiche “Sperrlistenmanagement” und “Verzeichnisdienste” eine entscheidende Rolle, die in diesem Dokument eingehend dargestellt werden. Zu der geschilderten Problematik existieren bereits zahlreiche international etablierte Verfahren und Mechanismen wie z.B. CRL (certificate revocation list, Sperrliste von Zertifikaten) [ITU-T X.509 97, RFC 2459 99], CDP (certificate distribution point, Verteilungspunkt

für Sperrlisten), OpenCDP (open CRL distribution process, Mechanismen zur Verteilung von Sperrlisten oder OCSP (online certificate status protocol, Protokoll zur Online-Abfrage des Zustandes von Zertifikaten) [PKIX OCSP 98].

Alle genannten Verfahren und Mechanismen für das Sperrlistenmanagement und Verzeichnisdienste lassen sich unabhängig von deren technischer Realisierung grob danach klassifizieren, welche Informationen sie enthalten, woher diese Informationen stammen, wie diese Informationen strukturiert sind und wem diese Informationen zugänglich gemacht werden, welchen Grad der Aktualität diese Informationen enthalten und wie sicher diese Informationen erstellt und verwaltet werden.

#### UNMITTELBARE KONSEQUENZEN FÜR TEILNEHMER UND ZERTIFIZIERUNGSSTELLEN

Als unmittelbare Folge der Forderung nach einer Minimierung des Restrisikos für Angriffe lassen sich für Teilnehmer und Zertifizierungsstellen folgende Konsequenzen ziehen:

##### Zertifizierungsstellen sollten möglichst frühzeitig

- Sperrsituationen erkennen,
- Meldungen von Teilnehmern erhalten,
- die Aktualisierung von Sperrlisten durchführen,
- Updates an ihren Verzeichnisdienst zur Verfügung stellen

und

- nur Online-Dienste mit bestimmten Sicherheitsrichtlinien zulassen.

##### Teilnehmer sollten möglichst frühzeitig

- Sperrsituationen erkennen,
- Meldung an die zugehörige Zertifizierungsstelle machen,
- sich aktuelle Sperrlisten beschaffen

und

- bei der Validierung von Signaturen in geeigneter Weise Online-Dienste zur Statusabfrage von Zertifikaten nutzen.

In den nächsten Kapiteln werden die Anfragen und Antworten des Verzeichnisdienstes sowie die einzelnen Sperrlistenfelder beschrieben. Jeder Abschnitt ist dabei durch die Punkte “Zweck”, “ASN.1-Definitionen”, “Allgemeine Konformitätsanforderungen” und “ISIS-Konformitätsanforderungen” untergliedert, die folgende Bedeutung haben:

##### Zweck

Unter diesem Unterpunkt wird die Bedeutung des betreffenden Sperrlistenfeldes beschrieben.

### ASN.1-Definitionen

Dieser Unterpunkt enthält die ASN.1-Definitionen der Verzeichnisdienstanfragen und -antworten sowie der Sperrlistenfelder gemäß der X.509-Norm. Für ISIS-spezifische Sperrlistenfelder, die Objektbezeichner, private Erweiterungen oder Attribute betreffen, werden eigene ASN.1-Definitionen angegeben.

### Allgemeine Konformitätsanforderungen

An dieser Stelle wird eine Zusammenstellung von wesentlichen internationalen Konformitätsanforderungen gegeben. Konformitätsanforderungen sind Aussagen in Normen oder Empfehlungen, die festlegen, was in einem bestimmten Kontext zu tun ist, was getan werden darf oder was nicht getan werden darf. Aus Gründen der Interoperabilität sind deshalb von der vorliegenden Signatur-Interoperabilitätsspezifikation auch internationale oder nationale Festlegungen, wie sie beispielsweise in [PKIX OCSP 98], [PKIX PRO 97], [DIN SigG/V 98] oder [MTRUST 96] getroffen wurden, zu beachten.

### ISIS-Konformitätsanforderungen

Dieser Unterabschnitt enthält Informationen über Einschränkungen und Anwendung der allgemeinen Konformitätsanforderungen hinsichtlich der durch [PKIX OCSP 98] bzw. X.509 möglichen Optionen. Im Rahmen der Signatur-Interoperabilitätsspezifikation ISIS werden insbesondere weitere, durch die Normen und Empfehlungen zugelassene Strukturelemente wie spezielle Objektbezeichner, private Erweiterungen oder Attribute festgelegt oder die Benutzung bestimmter Elemente verboten.

Desweiteren enthält dieser Unterpunkt für Sperrlisten implementations-technische Informationen über einzelne Sperrlistenfelder und deren Unterstrukturen in einer tabellarischen Übersicht (Muster siehe folgende Tabelle). Die erste Spalte enthält den ASN.1-Bezeichner des betreffenden Sperrlistenfeldes. Falls ein Sperrlistenfeld aus einer zusammengesetzten Struktur besteht, so werden auch die Bezeichner der Teilfelder aufgeführt. In der zweiten Spalte werden der zugelassene Wertebereich bzw. die zugelassenen Einzelwerte der Sperrlistenfelder dargestellt. Die dritte Spalte zeigt den Hexadezimalcode des Sperrlistenfeldes. Die vierte Spalte enthält die aktuelle Länge der Beispielfelder. Aus diesen Werten wird eine maximale Länge abgeleitet, die als Empfehlung vorgegeben und in der Spalte durch Graudruck hervorgehoben wird. In den Spalten 5 bis 8 wird die Bedeutung eines Feldes entweder als obligatorisch, verboten oder optional gekennzeichnet. Die Spalten 9 bis 12 dienen zur Klassifikation von bestimmten Sperrlistenfeldern, die als Erweiterungen bezeichnet werden.

Sperrlistenformate sind nach der abstrakten ASN.1-Syntax definiert [ITU-T X.681 94] und konkrete Sperrlisten werden nach den ASN.1-Transfersyntaxregeln [ITU-T X.690 94] kodiert, deren Kenntnis vorausgesetzt wird. Datentypen und Datenwerte werden nach ASN.1 durch das rekursive Schema "Typ-Länge-Wert" kodiert. Die Typ-Komponente (auch als sog. Tag-Feld bezeichnet) spezifiziert hierbei den Typ einer Sperrlistenstruktur, die Längenkomponente enthält die Länge des folgenden Sperrlistenfeldes in Bytes und die Wert-Komponente enthält das eigentliche Nutzdatenfeld, das seinerseits aus Unterstrukturen gemäß des Schemas "Typ-Länge-Wert" aufgebaut sein kann. Der Wertebereich der Wert-Komponente ist durch das Tag-

Feld bestimmt. Prinzipiell besitzt nur die Typ-Komponente eine feste Kodierung und die beiden anderen Komponenten haben eine variable Länge. Aus diesem Grund haben die zweite und die dritte Spalte der beschriebenen Tabelle überwiegend nur Beispielcharakter und dienen zur Illustration der Kodierung. Ebenso soll die in der vierten Spalte angegebene Längenangabe nur als minimale Länge verstanden werden, die ein System oder eine Anwendung unterstützen soll. In den angegebenen Beispielen sind die Tagfelder durch Fettschrift, die Längen durch Normalschrift und die Nutzdatenfelder durch Kursivschrift hervorgehoben.

**Tabelle 1: Implementations-technische Informationen**

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFI- KATSTYP	RELE- VANZ	KLASSIFI- KATION								
	(BEISPIELE)	(BEISPIELE)	[BYTES]	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
		<b>Tag</b> Länge Wert												

## V 2. VERZEICHNISDIENST

### V 2.1. Anforderungen aus dem Signaturgesetz und der Signaturverordnung

#### Anforderungen an das Sperren von Zertifikaten

Das Signaturgesetz [SigG 97] und die Signaturverordnung [SigV 97] regeln, wann Zertifikate zu sperren sind, wer Zertifikate sperren kann, wer eine Sperrung veranlassen kann, wie eine Sperrung vorgenommen werden muß und welche Angaben eine Sperrung enthalten muß.

Nach der Signaturverordnung [SigV 97, §4(1) 1.] hat ein Signaturschlüsselinhaber unverzüglich die Sperrung seines Signaturschlüssels zu veranlassen, wenn der private Signaturschlüssel verloren gegangen ist oder nicht mehr benötigt wird.

Gemäß Signaturgesetz [SigG 97, §8(1) Satz 1, §8(2), §8(3)] und Signaturverordnung [SigV 97, §9(1)] kann die Sperrung eines Zertifikates nicht nur vom Signaturschlüsselinhaber, sondern auch von seinem Vertreter oder von einer dritten Personen, von der Angaben in dieses Zertifikat aufgenommen wurden, veranlaßt werden. Desweiteren darf die Zertifizierungsstelle selbst ein Zertifikat sperren, wenn das Zertifikat auf Grund falscher Angaben erwirkt wurde oder sie ihre Tätigkeit einstellt und diese nicht von einer anderen Zertifizierungsstelle fortgeführt wird. Nach Signaturgesetz [SigG 97, §13(5) Satz 2] kann auch die zuständige Behörde (RegTP) eine Sperrung von Endanwender-Zertifikaten anordnen, falls sie einer Zertifizierungsstelle die Genehmigung entzogen hat und überzeugt ist, daß die von dieser Zertifizierungsstelle ausgestellten Zertifikate nicht hinreichend fälschungssicher sind oder daß zur Anwendung der Signaturschlüssel eingesetzte technische Komponenten Sicherheitsmängel aufweisen.

Zur Sperrung eines Zertifikats muß nach Signaturverordnung [SigV 97, §9(2)] entweder ein schriftlicher oder ein mit einer digitalen Signatur versehener Antrag einer berechtigten Personen der Zertifizierungsstelle vorliegen oder ein vereinbartes Authentisierungsverfahren angewandt worden sein.

Gemäß Signaturgesetz [SigG 97, §8(1) Satz 2/3] und Signaturverordnung [SigV 97, §9(3)] ist die Sperrung von Zertifikaten mit Angabe des Datums und der Uhrzeit im Verzeichnis eindeutig kenntlich zu machen und darf nicht rückgängig gemacht werden. Dieser Zeitpunkt gibt an, ab wann die Sperrung gilt. Eine rückwirkende Sperrung ist unzulässig.

Stellt eine Zertifizierungsstelle ihre Tätigkeit ein und die von ihr ausgestellten noch gültigen Zertifikate werden von keiner anderen Zertifizierungsstelle übernommen, so muß sie nach Signaturgesetz [SigG 97, §11(1)] und Signaturverordnung [SigV 97, §14(2)] diese Zertifikate sperren.

### Anforderungen an den Verzeichnisdienst

Das Signaturgesetz [SigG 97] und die Signaturverordnung [SigV 97] regeln welche Zertifikate abrufbar oder nachprüfbar gehalten werden müssen, wie lange Zertifikate in einem Verzeichnis gehalten werden müssen, wie Zertifikatsverzeichnisse geschützt werden müssen, und wie Auskünfte eines Verzeichnisdiensts beschaffen sein müssen.

Signaturschlüssel-Zertifikate und Attribut-Zertifikate müssen nach [SigG 97, §4(5)+§5(1)] des Signaturgesetzes nachprüfbar gehalten werden. Darüber hinaus müssen Zertifikate von Zertifizierungsstellen nicht nur nachprüfbar, sondern auch abrufbar sein; Endanwenderzertifikate hingegen dürfen nur mit Zustimmung des Signaturschlüssel-Inhabers abrufbar gehalten werden.

Die Signaturverordnung [SigV 97, §8] fordert, daß Zertifikate 35 Jahre nachprüfbar gehalten werden müssen. Das Nachprüfen selber ist im Einzelfall zu ermöglichen, d.h. es sind so lange Online-Mechanismen zur Verfügung zu stellen, wie der im Zertifikat aufgeführte Algorithmus mit den dazugehörigen Parametern nach [SigV 97, §17(2)] als geeignet beurteilt wird.

Die Zertifikatsverzeichnisse müssen nach Signaturgesetz [SigG 97, §14(3)] und Signaturverordnung [SigV 97, §16(4)] vor unbefugter Veränderung und unbefugtem Abruf geschützt werden, so daß nur befugte Personen Eintragungen und Veränderungen vornehmen können, die Sperrung von Zertifikaten nicht unbemerkt rückgängig gemacht werden kann und die Auskünfte auf ihre Echtheit überprüft werden können. Auskünfte vom Zertifikatsverzeichnis müssen beinhalten, ob die nachgeprüften Zertifikate zum angegebenen Zeitpunkt vorhanden und nicht gesperrt waren.

Sperrlisten können nur Aufschluß über gesperrte Zertifikate geben, beinhalten aber keine Informationen über alle ausgestellten Zertifikate. Da nach Signaturverordnung [SigV 97, §16(4)] gefordert ist, daß der Verzeichnisdienst nicht nur über eine mögliche Sperrung Auskunft gibt, sondern auch darüber, ob ein Zertifikat jemals ausgestellt wurde, reichen für eine Signaturgesetz-Konformität keine Sperrlisten-Mechanismen aus, wie sie üblicherweise in diesem Zusammenhang verwendet werden.

Der Verzeichnisdienst im Sinne des Signaturgesetzes ist ein Dienst, der Auskunft über den Status aller jemals ausgestellten Zertifikate geben kann. Er verfügt über ein eigenes Signaturschlüsselpaar, so daß seine Auskünfte auf Echtheit überprüft werden können. Der Verzeichnisdienst muß von den Zertifizierungsstellen angeboten werden und sollte von der RegTP zertifiziert werden.

Der Verzeichnisdienst im Sinne des Signaturgesetzes ist von Diensten zu unterscheiden, die üblicherweise als Verzeichnisdienst (X.500 oder LDAP Directory Service) bezeichnet werden. Im folgenden Dokument wird mit Verzeichnisdienst der Verzeichnisdienst im Sinne des Signaturgesetzes bezeichnet. Andere Dienste werden zur Unterscheidung mit dem englischen Begriff Directory Service benannt.

Für den Verzeichnisdienst im Sinne des Signaturgesetzes soll eine definierte Untermenge (Profil) des Protokolls *Online Certificate Status Protocols* (OCSP) verwendet werden, welches innerhalb der Internet Engineering Task Force (IETF) entwickelt wird. Die Version 1 Draft 7 des OCSP ist Basis für dieses Dokument. Es wird erwartet, daß sich das Protokoll in der Definition noch ändern kann. Insbesondere die Identifikation der Zertifikate sowie die Ausarbeitung weiterer Ergebniswerte wurde bereits in der Arbeitsgruppe diskutiert und wird in die neueren Versionen Einfluß finden.

Der derzeitige Stand bietet aber bereits alle Möglichkeiten, um den Anforderungen des Signaturgesetzes an den Verzeichnisdienst gerecht zu werden. Da das OCSP im internationalen Kontext entwickelt wird, halten wir es für sinnvoll, kein eigenes Protokoll für die Zwecke des Signaturgesetzes zu entwickeln.

In diesem Dokument sind keine speziellen Datenaustauschformate für Sperrlisten bzw. OCSP-Anfragen und -Antworten beschrieben. Sperrlisten werden in der Regel über X.500 oder LDAP verteilt. Zum Transport von Verzeichnisdienstanfragen und -antworten über HTTP sind in Kapitel V 2.3 Erläuterungen zu finden..

## **V 2.2. Online-Verzeichnisdienst OCSP**

Regelmäßig verteilte Sperrlisten können Sperrinformationen nur in festen Intervallen der Anwenderinfrastruktur verfügbar machen. Der Zeitraum zwischen offizieller Sperrung eines Zertifikates und dem Bekanntwerden der Sperrung in der Anwenderinfrastruktur ist also von der Häufigkeit der Erstellung der Sperrlisten abhängig. Der vertretbare Zeitraum, der zwischen der Sperrung und dem Bekanntwerden der Sperrung vergeht, ist abhängig vom Verwendungszweck der geleisteten Signatur.

Da keine allgemein akzeptable Zeitverzögerung festgelegt werden kann, wird mit dem Verzeichnisdienst ein Auskunftsdienst definiert, der ein zeitnahes Sperren (innerhalb von ca. zehn Minuten) und eine direkte Abfrage der Sperrungen ermöglicht. Ein ebenso zeitnahes Sperren mit CRLs realisieren zu wollen würde bedeuten, daß alle zehn Minuten eine neue CRL erstellt und verteilt werden müßte. Dies ist nicht praktikabel. Die Verwendung eines Verzeichnisdienstes zur Prüfung des Sperrzustandes von Zertifikaten ersetzt die sonst übliche Methode der Prüfung mittels CRLs.

Es gibt bisher nur eine einzige Definition eines On-line Protokolls, das Statusinformationen zu Zertifikaten liefert, nämlich OCSP – online certificate status protocol [PKIX OCSP 98]. Der OCSP-Verzeichnisdienst wird durch die zwei ASN.1-Strukturtypen *OCSPRequest* und *OCSPResponse* definiert, mit denen Anfragen an den Verzeichnisdienst und zugehörige Antworten des Verzeichnisdienstes realisiert werden.

### V 2.2.1. ANFRAGEN AN DEN VERZEICHNISDIENST

Die Anfrage zur Gültigkeit eines Zertifikates muß immer an die Zertifizierungsstelle gerichtet werden, die das betreffende Zertifikat ausgestellt hat. Dies würde bedeuten, daß der Anwenderinfrastruktur die Dienstadressen aller Zertifizierungsstellen bzw. deren Verzeichnisdienste bekannt sein müßten. Die Ableitung einer Dienstadresse aus dem Namen der Zertifizierungsstelle wurde in [Z 4 Anhang II] schon kurz beschrieben. Jeder Directory Service einer Zertifizierungsstelle sollte daher Informationen über die Dienstadressen der anderen Verzeichnisdienste halten. Dies kann auch durch einen Verweis auf ein signiertes Dokument erfolgen, welches alle Zertifizierungsstellen sowie deren Dienstadressen enthält.

#### Zweck

Die Anfrage des OCSP soll das zu prüfende Zertifikat festlegen. Die Anfrage besteht aus einem Anfrageblock mit einer Folge von Zertifikaten bzw. Zertifikatsbezeichnern. Jeder einzelnen Zertifikatsanfrage sowie der gesamten Anfrage können Erweiterungen angefügt werden. Zur Authentisierung des Anfragenden kann die Anfrage signiert werden. Eine Anfrage an den Verzeichnisdienst enthält folgende Daten:

- Protokollversion
- ein oder mehrere Zertifikatbezeichner
- zusätzliche Erweiterungen je Zertifikat und Erweiterungen je Anfrage
- optional den technischen Namen des Anfragenden
- eine optionale Signatur der Anfrage

Eine Anfrage bezieht sich auf ein oder mehrere Zertifikate. Üblicherweise wird die Anwenderinfrastruktur in einer Anfrage nur nach einem einzigen Zertifikat fragen, da für jedes Zertifikat einer Zertifikatskette der Verzeichnisdienst der jeweils ausstellenden Zertifizierungsstelle kontaktiert werden muß. Das Protokoll unterstützt die Anfrage mehrerer Zertifikate, um Anwenderinfrastrukturen eine Optimierung der Zahl der Anfragen zu ermöglichen.

Der Zeitpunkt einer Anfrage, auf den hin das Vorhandensein und der Zustand eines Zertifikats geprüft werden soll, muß nicht in der OCSP-Anfrage enthalten sein, es genügt, daß die OCSP-Antwort den Zeitpunkt enthält, seit dem das Zertifikat im Verzeichnis vorhanden ist, und daß darüberhinaus der Zeitpunkt der Antworterstellung und im Fall gesperrter Zertifikate der Sperrzeitpunkt enthalten sind (siehe Kapitel V 2.2.2).

#### ASN.1-Definitionen

```
OCSPRequest ::= SEQUENCE {
    tbsRequest          TBSRequest
    optionalSignature    [0] Explicit Signature OPTIONAL }

TBSRequest ::= SEQUENCE {
    version              [0] EXPLICIT Version DEFAULT v1,
    requestorName        [1] EXPLICIT GeneralName OPTIONAL,
    requestList          SEQUENCE OF Request,
```

---

```

    requestExtensions    [2] EXPLICIT Extensions OPTIONAL }
Version                ::= INTEGER { v1(0) }
Signature              ::= SEQUENCE {
    signatureAlgorithm   AlgorithmIdentifier,
    signature            BIT STRING,
    certs                [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL}
Request                ::= SEQUENCE {
    reqCert              CertID,
    singleRequestExtensions [0] EXPLICIT Extensions OPTIONAL }
CertID                 ::= SEQUENCE {
    hashAlgorithm        AlgorithmIdentifier,
    issuerNameHash       OCTET STRING,
    issuerKeyHash        OCTET STRING,
    serialNumber         CertificateSerialNumber }
CertificateSerialNumber ::= INTEGER

```

### Beschreibung der Einzelkomponenten:

- OCSPREQUEST

Anfragen an den OCSP-Verzeichnisdienst werden durch die Struktur *OCSPRequest* spezifiziert, die ihrerseits aus den Teilfeldern *tbsRequest* und *optionalSignature* besteht. Das Feld *tbsRequest* enthält die eigentliche Anfrage an den Verzeichnisdienst.

- TBSREQUEST

Das *tbsRequest*-Teilfeld enthält eine Versionsnummer *version*, den optionalen technischen Namen des Anfragenden *requestorName*, eine Folge von Einzelanfragen *requestList*, sowie optionale Anfrageerweiterungen *requestExtensions*.

- VERSION

Das *version*-Teilfeld enthält die Versionsnummer der OCSP-Anfrage. Die Voreinstellung für dieses Feld hat den Wert 0, der die Version 1 anzeigt.

### Allgemeine Konformitätsanforderungen

Derzeit sind keine Konformitätsanforderungen in [PKIX OCSP 98] definiert.

### ISIS-Konformitätsanforderungen

Gegenwärtig ist nur die Version v1 mit dem Wert 0 definiert und über das DEFAULT Konstrukt auch vorbesetzt.

- REQUESTORNAME

Das optionale *requestorName*-Teilfeld enthält den Namen des Anfragenden, der seine Anfrage an den Verzeichnisdienst bei Bedarf signieren kann.

Allgemeine Konformitätsanforderungen

Falls eine OCSP-Anfrage signiert wird, so muß der Anfragende seinen Namen in dem *requestorName*- Teilfeld angeben.

ISIS-Konformitätsanforderungen

Falls die Anfrage an den Verzeichnisdienst signiert ist, so muß der angegebene technische Name mit dem technischen Namen des Signierers übereinstimmen.

- REQUESTLIST

Das *requestList*-Teilfeld enthält die Anfragen selbst. Es können mehrere Anfragen zu verschiedenen Zertifikaten gleichzeitig an einen Verzeichnisdienst geschickt werden.

Allgemeine Konformitätsanforderungen

Derzeit sind keine Konformitätsanforderungen in [PKIX OCSP 98] definiert.

ISIS-Konformitätsanforderungen

Die Anzahl der Anfragen im *requestList*-Feld muß mindestens eins betragen. Verzeichnisdienste müssen mehrere Anfragen verarbeiten können.

- REQUESTEXTENSIONS

Die Komponente *requestExtensions* dient zur Aufnahme von Erweiterungen, die für die gesamte OCSP-Anfrage gelten. Sie ist vom Typ *Extensions*, deren Syntax im Abschnitt Z 2.3.9 beschrieben ist.

Allgemeine Konformitätsanforderungen

In [PKIX OCSP 98] wird die RequestExtension und ResponseExtension (siehe 2.2.2) *Nonce* definiert, die die Anfrage kryptografisch an die Antwort bindet, um Wiedereinspielungsangriffe (replay attacks) zu verhindern. Die Erweiterung wird durch den Objektbezeichner *id-pkix-ocsp-nonce* identifiziert.

*id-pkix-ocsp-nonce* OBJECT IDENTIFIER ::= { id-pkix-ocsp 2 }

ISIS-Konformitätsanforderungen

Im Rahmen des ISIS Profils sind zur Zeit keine Anfrageerweiterungen für das *requestExtensions*-Teilfeld der *tbsRequest*-Struktur definiert.

- SIGNATURE

Das *optionalSignature*-Signaturfeld besteht aus der Beschreibung des verwendeten Signaturalgorithmus *signatureAlgorithm* und der Signatur *signature* selbst. Daneben kann vom Signierer noch eine Folge von Zertifikaten *certs* beigefügt werden, die dem Verzeichnisdienst das Auffinden von Zertifikaten für den Zertifizierungspfad ersparen.

#### Allgemeine Konformitätsanforderungen

Die Signatur wird über die gesamte *tbsRequest*-Struktur berechnet. Falls eine OCSP-Anfrage signiert wird, so muß der Anfragende seinen Namen im *requestorName*-Teilfeld der *tbsRequest*-Struktur angeben.

#### ISIS-Konformitätsanforderungen

Bei der Erzeugung einer Verzeichnisdienstanfrage ist die Benutzung einer Signatur im *optionalSignature*-Teilfeld optional. Falls eine Anfrage von der Anwenderinfrastruktur signiert wird und eine optionale Folge von Zertifikaten *certs* enthält, so muß diese zumindest das Teilnehmerzertifikat und kann darüberhinaus optional das Zertifikat der Zertifizierungsstelle enthalten. Zusätzlich kann noch das Zertifikat der ausstellenden Behörde beigefügt werden.

- REQUEST

In der Datenstruktur *Request* werden die einzelnen Zertifikate bezeichnet, deren Status die Anwenderinfrastruktur abfragen möchte. Der Request besteht aus einer Bezeichnung des Zertifikates *certID* und einer optionalen Folge von Anfrageerweiterungen *singleRequest-Extensions*, die sich auf dieses Zertifikat beziehen.

- CERTID

Ein Zertifikat wird mit der Datenstruktur *CertID* des *reqCert*-Teilfeldes eindeutig beschrieben. Die Kombination aus dem Namen des Ausstellers *issuerNameHash* und der Seriennummer *serialNumber* des betreffenden Zertifikates wird als Identifikator des Zertifikats verwendet. Das Feld *hashAlgorithm* enthält den Objektbezeichner eines geeigneten Hashalgorithmus. Das Feld *issuerNameHash* enthält das Ergebnis der Anwendung der Hashfunktion auf den nach DER-kodierten Namen des Ausstellers. Das Feld *issuerKeyHash* enthält das Ergebnis der Anwendung der Hashfunktion auf den Wert des Feldes *subjectPublicKey* (ohne den ASN.1-Tag und die Längenbeschreibung) aus dem Zertifikat des Ausstellers.

#### Allgemeine Konformitätsanforderungen

Der Hauptgrund für die Verwendung der beiden Hashwerte *issuerNameHash* und *issuerKeyHash*, um den Zertifikatsaussteller zu identifizieren, liegt darin, daß es möglich sein kann, daß zwei Zertifizierungsstellen den gleichen Namen wählen. Sie werden jedoch niemals den gleichen öffentlichen Schlüssel verwenden.

### ISIS-Konformitätsanforderungen

Um Anwenderinfrastrukturen auch Prüfungen von Zertifikaten zu ermöglichen, ohne daß das Zertifikat der entsprechenden Zertifizierungsstelle vorliegen muß, wird im ISIS-Profil für Anfragen an den Verzeichnisdienst erlaubt, als *issuerKeyHash*-Teilfeld einen Octet-String der Länge 0 zu kodieren. Damit genügt das Wissen über den Namen des Ausstellers und die Seriennummer des Zertifikates. Die Antworten des Verzeichnisdienstes müssen aber immer den *issuerKeyHash* mit dem entsprechenden Wert belegen.

- SINGLEREQUESTEXTENSIONS

Die Komponente *singleRequestExtensions* dient zur Aufnahme von Erweiterungen, die für einzelne Anfragen gelten. Sie ist vom Typ *Extensions*, deren Syntax im Abschnitt Z 2.3.9 beschrieben ist.

### Allgemeine Konformitätsanforderungen

Die Unterstützung von Erweiterungen ist optional. Das *critical* flag sollte für keine Erweiterung gesetzt werden. Unbekannte Erweiterungen dürfen ignoriert werden, sofern das *critical* flag nicht gesetzt ist.

### ISIS-Konformitätsanforderungen

Im Rahmen des ISIS Profils ist eine optionale Erweiterung *retrieveIfAllowed* für das *singleRequestExtensions*-Teilfeld der *Request*-Struktur vorgesehen, mit der die anfragende Anwenderinfrastruktur den Verzeichnisdienst anweisen kann, bei der Antwort das gesamte betroffene Zertifikat zu liefern, sofern dieses abrufbar gehalten wird. Die Erweiterung muß als *critical* markiert werden.

Außerdem wird eine optionale Erweiterung *certHash* für das *singleRequestExtensions*-Teilfeld der *Request*-Struktur vorgesehen, die den Hashwert des betreffenden Zertifikates enthält. Die Erweiterung muß als *non-critical* markiert werden. Diese Komponente etabliert eine kryptographische Bindung zwischen der Bytefolge des Zertifikates und der Antwort des Verzeichnisdienstes.

### ASN.1 Definitionen

```

id-isis OBJECT IDENTIFIER ::= { 1 3 36 8 }
id-isis-at OBJECT IDENTIFIER ::= { 1 3 36 8 3 }
id-isis-at-retrieveIfAllowed OBJECT IDENTIFIER ::=
                                     { 1 3 36 8 3 9 }

retrieveIfAllowed EXTENSION ::= {
    SYNTAX                               BOOLEAN DEFAULT FALSE
    IDENTIFIED BY                          id-isis-at-retrieveIfAllowed }
id-isis-at-certHash OBJECT IDENTIFIER ::= { 1 3 36 8 3 13 }
certHash EXTENSION ::= {
    SYNTAX                               CertHashSyntax
  
```

IDENTIFIED BY	id-isis-at-certHash }
<b>CertHashSyntax</b>	::= SEQUENCE {
hashAlgorithm	AlgorithmIdentifier,
certificateHash	OCTET STRING }

## V 2.2.2.           ANTWORTEN DES VERZEICHNISDIENSTES

Der Prüfzeitpunkt muß weder in der OCSP-Anfrage noch in der OCSP-Antwort enthalten sein, es genügt, daß die OCSP-Antwort den Erstellungszeitpunkt der Antwort und im Fall gesperrter Zertifikate den Sperrzeitpunkt enthält. Der Verzeichnisdienst liefert somit nicht direkt die Information, daß zu einem bestimmten Zeitpunkt ein Zertifikat gesperrt war, sondern er liefert den Zeitpunkt, ab dem ein Zertifikat gesperrt wurde. Die Anwenderinfrastruktur muß diesen Zeitpunkt in Bezug auf die zu prüfende Signatur auswerten, um festzustellen, ob das Zertifikat zum fraglichen Zeitpunkt bereits gesperrt war. Ist zur aktuellen Zeit ein Zertifikat nicht gesperrt, so war es auch zu keinem früheren Zeitpunkt gesperrt. Somit ist auch im Fall nicht gesperrter Zertifikate der Prüfzeitpunkt irrelevant. Auch bei der Frage nach unbekanntem Zertifikaten, d.h. nach Zertifikaten, die nicht von der zugehörigen Zertifizierungsstelle ausgestellt wurden, ist der Prüfzeitpunkt nicht relevant. Ein Zertifikat, das nicht von der zugehörigen Zertifizierungsstelle ausgestellt wurde, war niemals und wird niemals in der Liste der ausgestellten Zertifikate enthalten sein. Es gibt jedoch auch die Möglichkeit, daß ein Zertifikat zwar schon ausgestellt, aber noch nicht in den Verzeichnisdienst eingestellt wurde. Hierfür wurde die ISIS-spezifische Erweiterung *certInDirSince* eingeführt, die in einer OCSP-Antwort enthalten sein kann.

Die Antwort des Dienstes kann also für jedes angefragte Zertifikat grundsätzlich drei verschiedene Ergebnisse liefern: das Zertifikat ist vorhanden seit dem Zeitpunkt *<certInDirSince>* und nicht gesperrt, es ist vorhanden seit dem Zeitpunkt *<certInDirSince>* und gesperrt oder es ist nicht vorhanden (es befindet sich nicht in der Liste der ausgestellten Zertifikate dieser Zertifizierungsstelle). Die beiden ersten Ergebnisse können anstatt der Identifikation des Zertifikates auch das Zertifikat selbst enthalten. Jede dieser Antworten muß vom Verzeichnisdienst signiert werden. Die OCSP-Antwort enthält den Zeitpunkt, seit dem das Zertifikat in den Verzeichnisdienst gestellt wurde, und darüberhinaus im Fall nicht gesperrter Zertifikate die aktuelle Zeit *producedAt* und bei gesperrten Zertifikaten den Sperrzeitpunkt.

### 1. Das Zertifikat ist vorhanden seit dem Zeitpunkt *<certInDirSince>* und nicht gesperrt

Die Antwort enthält den Antwortzeitpunkt in der Teilkomponente *producedAt*. Damit ist sichergestellt, daß das Zertifikat auch zu keinem früheren Zeitpunkt gesperrt war. Sofern das Zertifikat abrufbar gehalten wird und der Abruf bei der Anfrage gewünscht wurde, liefert der Dienst das Zertifikat. Andernfalls wird nur die Statusinformation über das Zertifikat in der Antwort angegeben. Die Antwort "nicht gesperrt" wird mit der aktuellen Uhrzeit in der Teilkomponente *producedAt* versehen und vom Verzeichnisdienst signiert. Hierdurch werden Angriffe durch Wiedereinspielen alter Antworten verhindert.

### 2. Das Zertifikat ist vorhanden seit dem Zeitpunkt *<certInDirSince>* und gesperrt

Die Antwort enthält den Sperrzeitpunkt *revocationTime* und optional auch den Sperrgrund *revocationReason*. Sofern das Zertifikat abrufbar gehalten wird und der Abruf bei der Anfrage gewünscht wurde, liefert der Dienst das Zertifikat. Andernfalls wird nur die Statusinformation *certStatus* über das Zertifikat in der Antwort angegeben. Ein Angriff durch Wiedereinspielen von Antworten mit der Information der Zertifikatssperrung ist nicht möglich. Auskünfte mit Sperrung können deshalb vorproduziert und signiert werden, sie enthalten den Zeit-

punkt der Sperrung. Die Auswertung des Sperrzeitpunkts in Bezug auf die zu prüfende Signatur zu dem angegebenen Zeitpunkt muß von der Anwenderinfrastruktur durchgeführt werden.

### 3. Das Zertifikat war nicht ausgestellt

Diese Antwort enthält das entsprechende Ergebnis “nicht existent”, die aktuelle Uhrzeit sowie den Parameter zur Zertifikatsidentifizierung aus der Anfrage. Eine solche Antwort bedeutet, daß der befragte Verzeichnisdienst keine Auskunft über dieses Zertifikat geben kann, da es nicht von der zugehörigen Zertifizierungsstelle ausgestellt wurde oder es noch nicht in den Verzeichnisdienst eingestellt wurde.

ZUSAMMENSTELLUNG DER PRINZIPIELL IN DEN ANTWORTEN DES VERZEICHNISDIENSTES MÖGLICHEN ZEITPUNKTE:

#### producedAt

Die Komponente *producedAt* enthält den Zeitpunkt der Signatur dieser Antwort durch den Verzeichnisdienst. Dieser Zeitpunkt muß nicht mit der aktuellen Uhrzeit übereinstimmen, da Antworten bei bestimmten lokalen Sicherheitspolitiken wiederverwendet werden können. Der Zeitpunkt sollte nicht in der Zukunft liegen und die Zeitpunkte *thisUpdate* und *revocationTime* stimmen mit *producedAt* überein oder müssen vor diesem liegen. Der Zeitpunkt *nextUpdate* kann in der Zukunft liegen, sofern die Antwort auf einer Sperrliste mit einem festgelegtem Gültigkeitszeitraum basiert.

#### thisUpdate

Die Komponente *thisUpdate* enthält den Zeitpunkt, für den die hier gemachte Aussage gültig ist. Beim On-line Verzeichnisdienst stimmt dieser Zeitpunkt mit dem Zeitpunkt *producedAt* überein.

#### nextUpdate

Die Komponente *nextUpdate* enthält einen Hinweis, wann die Information, auf der die Antwort basiert, erneuert wird. Dieser Zeitpunkt ist nur für OCSP Antworten sinnvoll, die auf CRLs basieren. Beim On-line Verzeichnisdienst stimmt dieser Zeitpunkt immer mit dem Zeitpunkt *thisUpdate* überein.

#### revocationTime

Die Komponente *revocationTime* gibt bei gesperrtem Zertifikat den Zeitpunkt an, zu dem die Sperrung aktiv wurde, d.h. der Sperreintrag gemacht und über den Verzeichnisdienst für die Nutzer des Gesamtsystems verfügbar war.

#### certInDirSince

Der Zeitpunkt *certInDirSince* wurde eingeführt, um folgenden Sonderfall abzufangen: Ein Zertifikat wird in den Verzeichnisdienst eingestellt, wobei der Zeitpunkt der Einstellung in den Verzeichnisdienst nach dem Beginn des Gültigkeitszeitraumes des Zertifikates liegt. Anfragen zu einem Zertifikat vor der Einstellung in den Verzeichnisdienst würden mit ungül-

tig beantwortet werden (Antwort “Zertifikat nicht vorhanden”). Anfragen zu diesem Zertifikat nach der Einstellung würden mit “Zertifikat vorhanden seit und nicht gesperrt” beantwortet werden. Um diese widersprüchlichen Antworten klären zu können, wurde die ISIS-spezifische Erweiterung *certInDirSince* eingeführt, die den Zeitpunkt enthält, seit dem das Zertifikat im Verzeichnis vorhanden ist.

### ASN.1-Definitionen

```

OCSPPResponse ::= SEQUENCE {
    responseStatus      OCSPPResponseStatus,
    responseBytes       [0] EXPLICIT ResponseBytes OPTIONAL }

OCSPPResponseStatus ::= ENUMERATED {
    successful          (0),
    malformedRequest    (1),
    internalError       (2),
    tryLater            (3),
    -- not used         (4),
    sigRequired         (5),
    unauthorized        (6) }

ResponseBytes ::= SEQUENCE {
    responseType        OBJECT IDENTIFIER,
    response             OCTET STRING }

id-ad OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 48 }
id-pkix-ocsp OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 48 1 }
id-pkix-ocsp-basic OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 7 48 1 1 }

BasicOCSPPResponse ::= SEQUENCE {
    tbsResponseData     ResponseData,
    signatureAlgorithm   AlgorithmIdentifier,
    signature            BIT STRING,
    certs                [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL }

ResponseData ::= SEQUENCE {
    version              [0] EXPLICIT Version DEFAULT v1,
    responderID          ResponderID,
    producedAt           GeneralizedTime,
    responses            SEQUENCE OF SingleResponse,
    responseExtensions   [1] EXPLICIT Extensions OPTIONAL }

ResponderID ::= CHOICE {
    byName               [1] Name,
    byKey                [2] KeyHash }

KeyHash ::= OCTET STRING

SingleResponse ::= SEQUENCE {
    certID               CertID,
    certStatus           CertStatus,
    thisUpdate           GeneralizedTime,
    nextUpdate           [0] EXPLICIT GeneralizedTime OPTIONAL,
    singleExtensions     [1] EXPLICIT Extensions OPTIONAL }

CertID ::= SEQUENCE {

```

hashAlgorithm	AlgorithmIdentifier,
issuerNameHash	OCTET STRING,
issuerKeyHash	OCTET STRING,
serialNumber	CertificateSerialNumber }
<b>CertificateSerialNumber</b>	::= INTEGER
<b>CertStatus</b>	::= CHOICE {
good	[0] IMPLICIT NULL,
revoked	[1] IMPLICIT RevokedInfo,
unknown	[2] IMPLICIT UnknownInfo }
<b>RevokedInfo</b>	::= SEQUENCE {
revocationTime	GeneralizedTime,
revocationReason	[0] EXPLICIT CRLReason OPTIONAL }
<b>UnknownInfo</b>	::= NULL
<b>CRLReason</b>	::= ENUMERATED {
unspecified	(0),
keyCompromise	(1),
cACompromise	(2),
affiliationChanged	(3),
superseded	(4),
cessationOfOperation	(5),
certificateHold	(6),
removeFromCRL	(8) }

### Beschreibung der Einzelkomponenten

- **OCSPRESPONSE**

Die Antwort des Verzeichnisdienstes *OCSPResponse* besteht aus dem allgemeinen Ergebnis der Anfrage *responseStatus* und dem optionalen Inhalt der Antwort *responseBytes*.

- **OCSPRESPONSESTATUS**

Das *responseStatus*-Feld der *OCSPResponse*-Struktur dient zur Anzeige des Ergebnisses einer Anfrage an den Verzeichnisdienst. Die möglichen Werte für dieses Feld sind in der folgenden Tabelle beschrieben.

#### Allgemeine Konformitätsanforderungen

Falls eine Anfrage vom Verzeichnisdienst erfolgreich bearbeitet werden konnte, so ist die explizite Antwort im *responseBytes*-Teilfeld von *OCSPResponse* enthalten.

Falls eine Anfrage ein Format enthält, das vom Verzeichnisdienst nicht bearbeitet werden kann, so erfolgt keine explizite Antwort im *responseBytes*-Teilfeld von *OCSPResponse*.

Falls ein interner Fehler im Verzeichnisdienst auftritt, so erfolgt keine explizite Antwort im *responseBytes*-Teilfeld von *OCSPResponse*.

Tabelle 2: Ergebnismeldungen des OCSP-Verzeichnisdienstes

Ergebnis der Anfrage		Bedeutung
Nummer	Name	
0	successful	Erfolgreiche Bearbeitung einer Anfrage
1	malformedRequest	nicht erfolgreiche Bearbeitung einer Anfrage wegen fehlerhaftem Anfrage-Format
2	internalError	Auftreten eines internen Fehlers beim Verzeichnisdienst
3	tryLater	Temporäre Nicht-Verfügbarkeit des Verzeichnisdienstes
4	certRequired	Forderung nach explizitem Vorhandensein des Zertifikates im <i>cert</i> -Teilfeld einer Anfrage, über das Auskünfte angefordert werden
5	sigRequired	Forderung nach signierten Anfragen

Falls der Verzeichnisdienst temporär nicht verfügbar ist, so erfolgt keine explizite Antwort im *responseBytes*-Teilfeld von *OCSPResponse*.

Falls der Verzeichnisdienst das explizite Vorhandensein des Zertifikates, über das Auskünfte eingeholt werden, im *cert*-Teilfeld einer Anfrage verlangt, so erfolgt keine explizite Antwort im *responseBytes*-Teilfeld von *OCSPResponse*.

Falls der Verzeichnisdienst signierte Anfragen verlangt und nicht-signierte Anfragen erhält, so erfolgt keine explizite Antwort im *responseBytes*-Teilfeld von *OCSPResponse*.

### ISIS-Konformitätsanforderungen

Als Ergebnisse von Verzeichnisdienstanfragen sind im Rahmen des ISIS-Profiles die in der folgenden Tabelle dargestellten Ergebnisse erlaubt.

Tabelle 3: Ergebnismeldungen des ISIS-OCSP-Verzeichnisdienstes

Ergebnis der Anfrage			Relevanz		
Nummer	Name	Bedeutung	obligatorisch	verboden	optional
0	successful	erfolgreiche Bearbeitung einer Anfrage			
1	malformedRequest	nicht erfolgreiche Bearbeitung einer Anfrage wegen fehlerhaftem Anfrage-Format			
2	internalError	Auftreten eines internen Fehlers beim Verzeichnisdienst			
3	tryLater	temporäre Nicht-Verfügbarkeit des Verzeichnisdienstes			
4	certRequired	Forderung nach explizitem Vorhandensein des Zertifikates im <i>cert</i> -Teilfeld einer Anfrage, über das Auskünfte angefordert werden			
5	sigRequired	Forderung nach signierten Anfragen			

- **RESPONSEBYTES**

Das *responseBytes*-Feld der *OCSPResponse*-Struktur enthält immer den Objektbezeichner *id-pkix-ocsp-basic* im *responseType*-Feld, sowie die Kodierung von *BasicOCSPResponse* im *response*-Feld der *ResponseBytes*-Struktur.

- **BASICOCSPRESPONSE**

Das *response*-Antwortfeld der *ResponseBytes*-Struktur vom Typ *BasicOCSPResponse* besteht aus den eigentlichen Daten der Antwort *tlsResponseData*, einer Beschreibung eines geeigneten Signaturalgorithmus *signatureAlgorithm*, dem Wert der Signatur *signature* selbst, sowie einer optionalen Folge von Zertifikaten *certs*, die den Zertifizierungspfad zum Verzeichnisdienst enthalten. Der Anwenderinfrastruktur wird bei der Verifikation der Signatur des Verzeichnisdienstes der Abruf weiterer Zertifikate erspart.

Allgemeine Konformitätsanforderungen

Falls die *BasicOCSPResponse*-Komponente eine Folge von Zertifikaten enthält, so darf deren Reihenfolge beliebig sein.

### ISIS-Konformitätsanforderungen

Falls die *BasicOCSPResponse*-Komponente eine Liste von Zertifikaten enthält, so sollte diese Liste das Zertifikat des Verzeichnisdienstes und das passende Zertifikat der ausstellenden Behörde enthalten. Optional kann noch das Zertifikat der Zertifizierungsstelle und das dazu passende Zertifikat der ausstellenden Behörde beigefügt werden.

- RESPONSEDATA

Die Nutzdaten der *BasicOCSPResponse*-Antwort sind in der Struktur *ResponseData* abgelegt, das eine Folge der Teilfelder *version*, *responderID*, *producedAt*, *responses* und optionalen *responseExtensions* ist.

- RESPONDERID

Die Identifizierung des Verzeichnisdienstes erfolgt über das *responderID*-Feld, welches entweder in der *byName*-Form den Namen des Verzeichnisdienstes enthält, so wie er in dem Zertifikat verzeichnet ist, welches zur Signatur der Antwort paßt oder in der *byKey*-Form den SHA-1-Hashwert (der nicht über das Tag- und Längenfild gebildet wird) des öffentlichen Schlüssels des Verzeichnisdienstes enthält, wie es in [PKIX OCSP 98] festgelegt wurde.

### Allgemeine Konformitätsanforderungen

Bei der *byKey*-Form zur Identifizierung des Verzeichnisdienstes erfolgt die SHA-1-Hashwertbildung nicht über das Tag- und Längenfild des öffentlichen Schlüssels des Verzeichnisdienstes.

### ISIS-Konformitätsanforderungen

Im Rahmen der ISIS-Spezifikation ist die Benutzung der *byName*-Form zur Identifikation des Verzeichnisdienstes obligatorisch und die der *byKey*-Form verboten.

- PRODUCED AT

Der Zeitpunkt, zu dem diese Antwort signiert wurde, wird in dem Feld *producedAt* angezeigt, das vom Typ *GeneralizedTime* ist.

### Allgemeine Konformitätsanforderungen

Sofern ein Verzeichnisdienst Antworten für gesperrte Zertifikate erstellt und für späteren Gebrauch signiert, kann dieser Zeitpunkt auch vom Zeitpunkt der Anfrage abweichen. In diesem Fall muß der Zeitpunkt von *producedAt* zwischen dem Sperrzeitpunkt und dem Zeitpunkt der Anfrage liegen.

### ISIS-Konformitätsanforderungen

Bei der Kodierung des *producedAt*-Feldes muß für GeneralizedTime das Format YYYYMMDDHHSSZ (siehe Abschnitt V 3.1.3.4) genommen werden.

- SINGLERESPONSES

Das Feld *responses* enthält eine Folge von Einzelantworten *SingleResponses*, die den entsprechenden Anfragen zugeordnet werden.

#### Allgemeine Konformitätsanforderungen

Die Reihenfolge der Einzelantworten *SingleResponses* im *responses*-Feld ist beliebig, und die Anwenderinfrastruktur muß die einzelnen Felder der Antworten den jeweiligen Anfragen zuordnen. Der Verzeichnisdienst muß zu jeder einzelnen Anfrage aus dem *requestList*-Feld eine Antwort im dem Feld *responses* bereitstellen. Sofern der Verzeichnisdienst dies in einem konkreten Fall nicht kann, muß er mit der Fehlermeldung *internal-Error* antworten.

### ISIS-Konformitätsanforderungen

Es gelten die allgemeinen Konformitätsanforderungen.

- RESPONSEEXTENSIONS

Die Komponente *responseExtensions* dient zur Aufnahme von Erweiterungen, die für die gesamte OCSP-Antwort gelten. Sie ist vom Typ *Extensions*, der im Abschnitt Z 2.3.9 beschrieben ist.

#### Allgemeine Konformitätsanforderungen

In [PKIX OCSP 98] wird die RequestExtension (siehe 2.2.1) und ResponseExtension *Nonce* definiert, die die Anfrage kryptografisch an die Antwort bindet, um Wiedereinspielungsangriffe (replay attacks) zu verhindern. Die Erweiterung wird durch den Objektbezeichner *id-pkix-ocsp-nonce* identifiziert.

id-pkix-ocsp-nonce      OBJECT IDENTIFIER ::= { id-pkix-ocsp 2 }

### ISIS-Konformitätsanforderungen

Im Rahmen des ISIS Profils sind zur Zeit keine neuen Erweiterungen für das *responseExtensions*-Teilfeld der *ResponseData*-Struktur definiert.

- SINGLERESPONSE

Antworten zu den einzelnen Zertifikaten werden in der Struktur *SingleResponse* kodiert, das seinerseits eine Folge der Teilfelder *certID*, *certStatus* und *thisUpdate*, sowie der optionalen Teilfelder *nextUpdate* und *singleExtensions* ist.

- CERTID

Ein Zertifikat wird mit der Datenstruktur *CertID* des *responses*-Teilfeldes eindeutig beschrieben. Die Kombination aus dem Namen des Ausstellers *issuerNameHash* und der Seriennummer *serialNumber* des betreffenden Zertifikates wird als Identifikator des Zertifikates verwendet. Das Feld *hashAlgorithm* enthält den Objektbezeichner eines geeigneten Hashalgorithmus. Das Feld *issuerNameHash* enthält das Ergebnis der Anwendung der Hashfunktion auf den nach DER-kodierten Namen des Ausstellers. Das Feld *issuerKeyHash* enthält das Ergebnis der Anwendung der Hashfunktion auf den Wert des Feldes *subjectPublicKey* (ohne den ASN.1-Tag und die Längenbeschreibung) aus dem Zertifikat des Ausstellers.

Allgemeine Konformitätsanforderungen

Der Verzeichnisdienst liefert üblicherweise im *certID*-Teilfeld nur einen Verweis auf das entsprechende Zertifikat.

ISIS-Konformitätsanforderungen

Im ISIS-Profil für Anfragen an den Verzeichnisdienst ist es erlaubt, als *issuerKeyHash*-Teilfeld einen Octet-String der Länge 0 zu kodieren. Die Antworten des Verzeichnisdienstes müssen aber immer den *issuerKeyHash* mit dem entsprechenden Wert belegen.

- CERTSTATUS

Das *certStatus*-Teilfeld enthält Informationen über den Sperrzustand des in *certID* bezeichneten oder enthaltenen Zertifikates.

Der Zustand *good* sagt aus, daß das Zertifikat von der zugehörigen Zertifizierungsstelle ausgestellt wurde, dem Verzeichnisdienst bekannt ist und zum Zeitpunkt *thisUpdate* nicht gesperrt ist.

Der Zustand *revoked* sagt aus, daß das Zertifikat von der zugehörigen Zertifizierungsstelle ausgestellt wurde, dem Verzeichnisdienst bekannt ist und gesperrt ist.

Der Zustand *unknown* sagt aus, daß das Zertifikat von der zugehörigen Zertifizierungsstelle nicht ausgestellt wurde und dieser nicht bekannt ist.

### Allgemeine Konformitätsanforderungen

Falls ein *revoked* Zustand vorliegt, so wird der Sperrzeitpunkt im Teilfeld *revocationTime* und optional der Sperrgrund im Teilfeld *revocationReason* der Struktur *RevokedInfo* abgelegt.

Für die Sperrgründe gelten die gleichen Regelungen wie bei den Sperrgründen für Sperrlisten (siehe 3.2.3.6.1).

Falls ein *unknown* Zustand vorliegt, so muß im Feld *certID* der Struktur *SingleResponse* der Inhalt des *certID*-Feld in der Struktur *Request* der Anfrage wiederholt werden.

### ISIS-Konformitätsanforderungen

Im ISIS-Profil für den Verzeichnisdienst kann in Verbindung mit den Zuständen *good* und *revoked* die ISIS-spezifische Erweiterung *CertInDirSince* für das *singleExtensions*-Teilfeld der *SingleResponse*-Struktur gesetzt sein, die den Zeitpunkt anzeigt, seit wann das in *certID* bezeichnete oder enthaltene Zertifikat im Verzeichnis vorhanden ist. Im Falle des Zustands *unknown* muß die ISIS-spezifische Erweiterung *CertInDirSince* nicht in der OCSP-Antwort enthalten sein.

Signaturgesetzkonforme Anwendungen müssen *certInDirSince* verarbeiten können.

- THISUPDATE

Im *thisUpdate*-Teilfeld wird der Zeitpunkt der Gültigkeit der Antwort des Verzeichnisdienstes abgelegt.

### Allgemeine Konformitätsanforderungen

OCSP-Antworten, deren *thisUpdate* Zeitpunkt nach der lokalen Systemzeit liegt, sollten als ungültig erachtet werden.

### ISIS-Konformitätsanforderungen

Die Komponente *thisUpdate* enthält den Zeitpunkt, für den die hier gemachte Aussage gültig ist. Beim on-line Verzeichnisdienst stimmt dieser Zeitpunkt mit dem Zeitpunkt *producedAt* überein.

- NEXTUPDATE

Im optionalen *nextUpdate*-Teilfeld kann der Ablaufzeitpunkt der Gültigkeit der Antwort eingetragen werden. Bevor oder ab diesem Zeitpunkt gibt es neuere Informationen über den Status von Zertifikaten.

### Allgemeine Konformitätsanforderungen

OCSP-Antworten, die keinen *nextUpdate* Zeitpunkt enthalten, zeigen an, daß jederzeit neuere Statusinformation zu Zertifikaten vorhanden sein kann.

### ISIS-Konformitätsanforderungen

Im ISIS-Profil ist die Benutzung des *nextUpdate*-Teilfeld verboten, da Antworten des Verzeichnisdienstes im Kontext des Signaturgesetzes nicht für Zeiträume gültig sind.

- **SINGLEEXTENSIONS**

Die Komponente *singleExtensions* dient zur Aufnahme von Erweiterungen, die für einzelne Antworten gelten. Sie ist vom Typ *Extensions*, der im Abschnitt Z2.3.9 beschrieben ist.

### Allgemeine Konformitätsanforderungen

Die Anwenderinfrastruktur muß Antworten mit vorhandenem *singleExtensions*-Teilfeld verarbeiten können. Sie muß aber nicht auf deren Inhalt reagieren. Insbesondere können Erweiterung die Aussagen im *certStatus*-Teilfeld weder modifizieren oder einschränken.

### ISIS-Konformitätsanforderungen

Im Rahmen der Spezifikation soll der Zeitpunkt, zu dem das nachgefragte Zertifikat in den Verzeichnisdienst gestellt wurde, in der Antwort des OCSP-Dienstes enthalten sein. Hierfür wurde die optionale ISIS-spezifische Erweiterung *CertInDirSince* für das *singleExtensions*-Teilfeld der *SingleResponse*-Struktur vorgesehen, mit der der Zeitpunkt angegeben wird, seit wann das Zertifikat im Verzeichnis vorhanden ist. Die Erweiterung kann als *non-critical* markiert werden.

Sofern bei der Anfrage für ein betroffenes Zertifikat die ISIS-Erweiterung *retrieveIfAllowed* gesetzt war und der Inhaber des Zertifikates dem öffentlichen Abruf zugestimmt hat, muß der Verzeichnisdienst das Zertifikat zurückliefern.

Zu diesem Zweck wurde im Rahmen des ISIS Profils die optionale Erweiterung *requestedCertificate* für das *singleExtensions*-Teilfeld der *SingleResponse*-Struktur vorgesehen. Diese Erweiterung muß als *non-critical* markiert werden.

### ASN.1 Definitionen

```
id-isis OBJECT IDENTIFIER ::= { 1 3 36 8 }
id-isis-at OBJECT IDENTIFIER ::= { 1 3 36 8 3 }
id-isis-at-CertInDirSince OBJECT IDENTIFIER ::=
    { 1 3 36 8 3 12 }

CertInDirSince EXTENSION ::= {
    SYNTAX CertInDirSinceSyntax
    IDENTIFIED BY id-isis-at-CertInDirSince }
CertInDirSinceSyntax ::= GeneralizedTime
```

---

```

id-isis-at-requestedCertificate OBJECT IDENTIFIER ::=
    { 1 3 36 8 3 10 }

requestedCertificate EXTENSION ::= {
    SYNTAX RequestedCertificateSyntax
    IDENTIFIED BY id-isis-at-requestedCertificate }

RequestedCertificateSyntax ::= Certificate

id-isis-at-certHash OBJECT IDENTIFIER ::= { 1 3 36 8 3 13 }

certHash EXTENSION ::= {
    SYNTAX CertHashSyntax
    IDENTIFIED BY id-isis-at-certHash }

CertHashSyntax ::= SEQUENCE {
    hashAlgorithm AlgorithmIdentifier,
    certificateHash OCTET STRING }

```

Wenn die ISIS-Erweiterung *retrieveIfAllowed* in der Anfrage gesetzt war, das Zertifikat aber nicht abrufbar ist, muß der Verzeichnisdienst nur den Verweis auf das Zertifikat im *certID*-Teilfeld liefern. Die Anwenderinfrastruktur kann so feststellen, daß ein Zertifikat nicht abrufbar ist.

Die ISIS-Erweiterungen *certHash* ist als *non-critical* zu markieren. Die ISIS-Erweiterung *certHash* etabliert eine kryptographische Bindung zwischen der Bytefolge des Zertifikates und der Antwort des Verzeichnisdienstes.

## V 2.3. Transport von Verzeichnisdienstsanfragen über HTTP

Die Kommunikation erfolgt über die beschriebenen Datenformate, die als Nachrichten über das Hypertext Transfer Protokoll (HTTP) an den Verzeichnisdienst übermittelt werden. Die Verwendung von HTTP [RFC 2068 97] erlaubt eine einfache Erstellung von Software für den Zugriff auf Verzeichnisdienste unter Verwendung erprobter Programmbibliotheken. Weiterhin sind für HTTP Übergangsmöglichkeiten von firmeninternen Netzen in öffentliche Netze, sogenannten Firewalls, definiert und in der Form von HTTP Proxies bereits erprobt.

Sofern in einer speziellen Anwendung die Geheimhaltung der Anfragen gewünscht wird, kann die Übertragung verschlüsselt über Transport Layer Security (TLS) bzw. Secure Socket Layer (SSL) erfolgen. Da diese Sicherungsmechanismen nur einen gesicherten Kanal für HTTP bereitstellen, sind diese transparent zu den hier beschriebenen Definitionen.

### V 2.3.1. DEFINITIONEN FÜR DIE ANFRAGEN

HTTP basierte OCSP Anfragen können entweder die HTTP Zugriffsmethode GET oder POST verwenden, wobei Anfragen, die eine Länge von mehr als 254 Bytes haben, aus-

schließlich mit der Methode POST an den Verzeichnisdienst übermittelt werden müssen. Der Verzeichnisdienst muß Anfragen beider Methoden verstehen und verarbeiten können.

Anfragen an den Verzeichnisdienst mit der Zugriffsmethode GET verwenden folgendes Format, wobei der base-64 DER kodierte Anfrage die ASN1. Struktur *OCSPRequest* aus Kapitel V 2.2.1 zugrunde liegt:

```
GET {url}/{url-kodiertes base-64 DER kodierte Anfrage}
```

Die zu verwendende URL wird aus der lokalen Konfiguration des Clients oder mit einer der in [Z 4 Anhang II] beschriebenen Methoden zur Ermittlung von Dienstadressen abgeleitet.

Eine Anfrage mit der Methode POST verwendet den HTTP Headerfeld "Content-Type" mit dem Wert "application/ocsp-request". Die Länge der Anfrage muß im HTTP Headerfeld "Content-Length" mit der Länge des Inhaltes der HTTP Anfrage belegt werden. Als Inhalt der HTTP Anfrage wird die DER kodierte Anfrage an den Verzeichnisdienst verwendet.

Anfragen mit der Methode POST können entweder binär oder base 64 kodiert erfolgen. Die Anfrage sollte binär übermittelt werden, sofern das Transportmedium dies zuläßt. Verzeichnisdienste müssen beide Kodierungen verarbeiten können.

Anfragen an den Verzeichnisdienst mit der Zugriffsmethode POST verwenden folgendes Format, wobei der base-64 DER kodierte Anfrage die ASN1. Struktur *OCSPRequest* aus Kapitel V 2.2.1 zugrunde liegt:

```
POST {url} ...
Content-Type: application/ocsp-request
Content-Length: ...
{binaer oder base-64 DER kodierte Anfrage}
```

### V 2.3.2. DEFINITIONEN FÜR DIE ANTWORTEN

Eine HTTP Antwort besteht aus den üblichen HTTP Headern, der Inhalt des Dokumentes ist die DER kodierte Antwort, wobei der DER kodierte Antwort die ASN1. Struktur *OCSPResponse* aus Kapitel V 2.2.2 zugrunde liegt. Der Transport kann entweder binär oder base 64 kodiert erfolgen. Die Antwort sollte binär übermittelt werden, sofern das Transportmedium dies zuläßt.

Im HTTP Header "Content-Type" muß der Wert "application/ocsp-response" angegeben werden, im Header "Content-Length" sollte die die Länge des folgenden Dokumentes verzeichnet sein. Andere HTTP Header können vorhanden sein und können von der Anwenderinfrastruktur ignoriert werden.

### *V 2.3.2.1. Abspeichern von Antworten*

Antworten vom Verzeichnisdienst können von der Anwenderinfrastruktur für spätere Nachweise abgespeichert werden. In diesem Fall sollte die Datei in ihrem Format der Antwort des HTTP Protokolls entsprechen. Dabei kann die erste Zeile des HTTP Protokolls (“HTTP..”) weggelassen werden. Minimal vorhanden sein muß die Headerzeile “Content-Type”. Die Dateinamenserweiterung für Antworten des Verzeichnisdienstes ist “.ors”.

## V 2.3.3. VERWENDUNG VON PROXIES

### *V 2.3.3.1. Verwendung von HTTP Proxy Servern*

Um Anwenderinfrastrukturen in Unternehmensnetzwerken eine Kommunikation mit Verzeichnisdiensten über Firewallssysteme zu erlauben, können die Anfragen über einen sogenannten HTTP Proxy Server übermittelt werden. Der Proxy Server dient dabei als Mittler zwischen der Anwenderinfrastruktur und dem Verzeichnisdienst.

Üblicherweise leitet ein Proxy Server eine HTTP Anfrage an einen Rechner weiter, der vom Anwender in der HTTP Anfrage spezifiziert wurde. Diese Methode erlaubt einer Anwenderinfrastruktur den transparenten Zugriff auf Verzeichnisdienste.

### *V 2.3.3.2. Verwendung von SSL Proxy Servern*

Sofern eine Verschlüsselung der Anfragen und Antworten des Verzeichnisdienstes gewünscht wird, kann die HTTP Anfrage über SSL gesichert werden. Sofern auch hier ein Firewallsystem zur Trennung von Netzen eingesetzt wird, kann über die SSL Proxy Spezifikation eine SSL Verbindung zum Verzeichnisdienst aufgebaut werden.

### *V 2.3.3.3. Verwendung eines Verzeichnisdienst Proxies*

Alternativ zum normalen HTTP Proxy Server könnte ein HTTP Proxy Server so konfiguriert werden, daß die Anwenderinfrastruktur alle Anfragen an Verzeichnisdienste an einen – möglicherweise lokal konfigurierten – Verzeichnisdienst sendet. Dort würde der HTTP Proxy die in der HTTP Anfrage kodierte Verzeichnisdienstsanfrage dekodieren und so den korrekten Verzeichnisdienst ermitteln. An diesen wird die Anfrage dann unverändert weitergeleitet. Die Integrität, Authentizität und Aktualität der Antwort ist durch die Zeitangaben und die digitale Signatur sichergestellt.

Eine solche Konfiguration hat den Vorteil, daß die einzelne Anwenderinfrastruktur keinen Zugriff auf ein öffentliches Netz haben muß und auch keine Informationen über die verschiedenen Dienstadressen halten muß.

Zusätzlich könnten Anfragen dann vom Verzeichnisdienst Proxy per SSL verschlüsselt an den jeweiligen Verzeichnisdienst weitergeleitet werden, so daß – etwa bei Großanwendern –

die Daten vertraulich behandelt werden und dieses Verhalten auch unternehmensweit eingeführt werden kann.

#### *V 2.3.3.4. Dienstäquivalenz von Verzeichnisdiensten*

Prinzipiell wäre es möglich, daß jeder der Verzeichnisdienste einen äquivalenten Dienst anbietet, so daß die Anwenderinfrastruktur einen beliebigen Verzeichnisdienst für alle Auskünfte kontaktieren kann. Dies erfordert natürlich, daß die äquivalenten Dienste synchron arbeiten müssen.

Neben einer solchen Implementation wäre es auch möglich, daß Verzeichnisdienst bemerken, wenn Anfragen zu einem falschen Dienst gesendet werden und in der Antwort mittels eines HTTP Redirects die korrekte Dienstadresse angeben.

## **V 2.4. Transport von Verzeichnisdienstanfragen über E-Mail**

Im Rahmen der ISIS-Spezifikation wird die Benutzung von E-Mail für die Kommunikation mit dem Verzeichnisdienst nicht empfohlen, weil die zugehörigen Prozesse asynchron ablaufen und insbesondere den Verifikationsprozeß des Anfragenden unterbrechen. Die Antworten des Verzeichnisdienstes gelangen in die Mailbox des Anfragenden und müssen über manuelle Interaktionen dem Verifikationsprozeß zugänglich gemacht werden.

## V 3. SPERRLISTENMANAGEMENT

### V 3.1. Sperrlistenformate

#### Zweck

Sperrlistenformate und Erweiterungen für Sperrlisten spielen eine wichtige Rolle bei Realisierung von öffentlichen Sicherheitsinfrastrukturen. Gegenwärtig dient das von der PKIX-Arbeitsgruppe, auf dem internationalen Standard X.509 v2 basierende, Sperrlistenprofil [ITU-T X.509 97, PKIX PRO 98] als Grundlage für die Entwicklung zahlreicher Anwendungen und Systemumgebungen.

Der ASN.1-Sperrlistentyp *CertificateList* besteht syntaktisch aus einer Folge von jeweils drei Feldern, die zur Trennung der zu signierenden Daten *tbsCertList*, des zu benutzenden Signaturalgorithmus *signatureAlgorithm* und der eigentlichen Signatur *signature* dienen.

#### ASN.1-Definitionen

```
CertificateList ::= SEQUENCE {  
    tbsCertList          TBSCertList,  
    signatureAlgorithm  AlgorithmIdentifier,  
    signature           BIT STRING }
```

#### Allgemeine Konformitätsanforderungen

In PKIX wird von konformen Zertifizierungsstellen nicht generell die Erstellung von Sperrlisten (CRL, certificate revocation list) gefordert, falls andere Mechanismen für die Zurückziehung oder die Zustandsanzeige von Zertifikaten bereitgestellt werden. Konforme Zertifizierungsstellen, die CRLs erstellen, sollen hierbei die Version 2 (siehe Abschnitt V 3.1.3.1) benutzen und müssen das Datum und den Zeitpunkt (siehe Abschnitt V 3.1.3.5) der nächsten CRL-Erstellung anzeigen.

Von konformen Systemen und Anwendungen wird erwartet, daß sie CRLs der Version 1 und 2 verarbeiten können.

#### ISIS- Konformitätsanforderungen

ISIS-konforme Zertifizierungsstellen müssen CRLs mit der Version 2 erstellen und müssen hierbei das nächste CRL-Erstellungsdatum im *nextUpdate*-Feld der CRL eintragen.

### V 3.1.1. SIGNATURALGORITHMUS

#### Zweck

Das Signaturfeld *signatureAlgorithm* vom Typ *AlgorithmIdentifier* enthält den Bezeichner des kryptographischen Algorithmus, der von der Zertifizierungsstelle zum Signieren der Sperrliste benutzt wird. Hierbei ist zu beachten, daß Signaturalgorithmen immer in Kombination mit Einweg-Hash-Funktionen und digitalen Signaturformaten (message formatting, padding) benutzt werden. Das Signaturfeld besteht syntaktisch aus einer Folge von Teilfeldern

*algorithm* und *parameters*. Das Teilfeld *algorithm* ist ein Objektbezeichner, der zur Identifikation des Algorithmus dient. Der Inhalt des optionalen *parameters*-Teilfeldes ist abhängig vom angegebenen Algorithmus und dem Algorithmusbezeichner.

### ASN.1-Definitionen

```

Certificate ::= SEQUENCE {
    ...,
    signatureAlgorithm AlgorithmIdentifier,
    ... }

AlgorithmIdentifier ::= SEQUENCE {
    algorithm OBJECT IDENTIFIER,
    parameters ANY DEFINED BY algorithm OPTIONAL }

```

### Allgemeine Konformitätsanforderungen

Das Signaturfeld *signatureAlgorithm* muß denselben Algorithmusbezeichner wie das *signature*-Teilfeld der *tbsCertList*-Struktur enthalten.

### ISIS-Konformitätsanforderungen

Das optionale *parameters*-Teilfeld darf nicht zur Übergabe von Parametern an den Algorithmus benutzt werden, da dieses Feld nicht durch die Signatur der Zertifizierungsstelle geschützt ist. Diese Einschränkung gilt nur für den "äußeren" Algorithmusbezeichner der *CertificateList*-Struktur und nicht für den "inneren" Algorithmusbezeichner der *tbsCertList*-Struktur, da letzterer durch die Signatur der Zertifizierungsstelle beglaubigt ist. Trotzdem darf auch der innere Algorithmusbezeichner nicht mit Parametern versehen werden und dessen Komponente *parameters* ist mit dem Wert NULL zu belegen. Zum Signieren geeignete und zugelassene Algorithmen werden von der Regulierungsbehörde für Telekommunikation und Post im Bundesanzeiger veröffentlicht. Die in der Ausgabe von Februar 1998 aufgeführten und geeigneten Kryptoalgorithmen gelten für die kommenden sechs Jahre (14. Februar 1998 – 14. Februar 2004).

**Tabelle 4: Implementations-technische Informationen über signatureAlgorithm**

BEZEICHNER	WERTEBEREICH EINZELWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	RELEVANZ		
				obligatorisch	verboten	optional
	(BEISPIELE)	(BEISPIELE)	20			
signatureAlgorithm Algorithm algorithm parameters	rsaSignatureWithripemd160 SEQUENCE { {1 3 36 3 3 1 2 }, NULL }	30 0A 06 06 2B 24 03 03 01 02 05 00	11			

Die Algorithmen und Parameter, mit denen eine Zertifizierungsstelle eine Sperrliste signiert, müssen mindestens für die Gültigkeitsdauer der Sperrliste als geeignet beurteilt sein. Weitere Einzelheiten zu dem Thema "Signaturalgorithmen" sind in der Teilspezifikation "A2 Signatur" von SigI [A2 99] angegeben. Die maximale Länge des *signatureAlgorithm*-Feldes beträgt 20 Bytes.

### V 3.1.2. SIGNATUR EINER SPERRLISTE

#### Zweck

Das Signaturfeld *signature* enthält eine digitale Signatur, die für das in ASN.1-DER kodierte Sperrlistenfeld *tbsCerList* berechnet wird. Bei der Berechnung der Signatur wird das Sperrlistenfeld *tbsCerList* als Eingabe in eine Einweg-Hash-Funktion benutzt. Auf den Ergebniswert der Hashfunktion wird der private Schlüssel der Zertifizierungsstelle angewandt und als ASN.1-Bitstring kodiert. Er liefert die konkrete digitale Signatur der Sperrliste im Signaturfeld *signature*. Durch den Signaturvorgang beglaubigt eine Zertifizierungsstelle die Gültigkeit der im Sperrlistenfeld *tbsCerList* enthaltenen Informationen und gewährleistet insbesondere die Ungültigkeit der betreffenden Zertifikate.

#### ASN.1-Definition

```
Certificate ::= SEQUENCE {
    . . .
    signature BIT STRING }
```

#### Allgemeine Konformitätsanforderungen

Geeignete Signaturformate finden sich in den Spezifikationen [PKCS1 93, Abschnitt 8.1] und [DIN SigG/V 98, Anhang A]. Üblicherweise wird das Ergebnis der Einweg-Hash-Funktion an die Signaturkomponente übergeben. Die Komponente ergänzt gegebenenfalls den ihr übergebenen Hashwert um zusätzliche Komponenten, bevor die eigentliche mathematische Signaturfunktion angewendet wird (siehe Teilspezifikation "A2 Signatur" von SigI).

#### ISIS-Konformitätsanforderungen

Bei der Erzeugung von Sperrlisten ist die Benutzung des Sperrlistenfeldes *signature* obligatorisch. Es dürfen nur die in der Teilspezifikation "A2 Signatur" von SigI [A2 99] aufgeführten Signaturalgorithmen und Signaturformate benutzt werden.

**Tabelle 5: Implementations-technische Informationen über *signature***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	RELEVANZ		
				obligatorisch	verboten	optional
	(BEISPIELE)	(BEISPIELE)	261			
Signature	BITSTRING --256 Byte-Schlüssellänge	03 82 01 01 . . .	261			

### V 3.1.3. ZU SIGNIERENDE SPERRLISTENINFORMATIONEN

#### Zweck

Das Sperrlistenfeld *tbsCertList* besteht aus einer Folge von optionalen und vorgeschriebenen Teilfeldern, die Informationen enthalten, die in unmittelbarem Zusammenhang mit dem Sperren von Zertifikaten stehen. Die in dieser Struktur vorgeschriebenen Bestandteile *signature*, *issuer* und *thisUpdate* dienen zur Kennung des benutzten Signaturalgorithmus, zur Identifikation des Erstellers der Sperrliste, sowie zur Angabe des Erstellungsdatums der Sperrliste. Die optionalen Teilfelder *version*, *nextUpdate*, *revokedCertificates* und *crlExtensions* der Sperrlistenstruktur enthalten die Version der aktuellen CRL, das Erstellungsdatums der nächsten Sperrliste, Listen von gesperrten Zertifikaten und mögliche CRL-Erweiterungen. Zertifikate werden innerhalb der Folge *revokedCertificates* durch ihre Seriennummer *userCertificate*, das Datum der Sperrung *revocationDate* und durch mögliche zertifikatsspezifische CRL-Eintragserweiterungen *crlEntryExtensions* als gesperrte Zertifikate gekennzeichnet.

#### ASN.1-Definitionen

```

CertificateList ::= SEQUENCE {
    tbsCertList      TBSCertList,
    ... }

TBSCertList ::= SEQUENCE {
    version          Version OPTIONAL,
    signature        AlgorithmIdentifier,
    issuer           Name,
    thisUpdate       Time,
    nextUpdate       Time OPTIONAL,
    revokedCertificates SEQUENCE OF SEQUENCE {
        userCertificate CertificateSerialNumber,
        revocationDate  Time,
        crlEntryExtensions Extensions OPTIONAL } OPTIONAL,
    crlExtensions    [0] EXPLICIT Extensions OPTIONAL }

```

**Tabelle 6: Implementations-technische Informationen über *tbsCertList***

FELD	ZERTIFIKATSTYP			RELEVANZ			FELD	ZERTIFIKATSTYP			RELEVANZ			
	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten		optional	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten
Version							nextUpdate							
Signature							revokedCertificates							
Issuer							crlExtensions							
ThisUpdate														

ISIS-Konformitätsanforderungen

Bei der Erzeugung von Sperrlisten ist die Verwendung des Feldes *crlExtensions* obligatorisch, da die beiden Sperrlistenenerweiterungen *authorityKeyIdentifier* und *cRLNumber* verwendet werden müssen. Außerdem ist auch die Benutzung der Teilfelder *version* und *nextUpdate* obligatorisch.

*V 3.1.3.1. Versionsnummer*Zweck

Das optionale Versionsfeld *version* gibt die Version des Sperrlistenformates an.

ASN.1-Definitionen

```
TBSCertList ::= SEQUENCE {
    version
    ... }
    Version OPTIONAL,

Version ::= INTEGER { v1(0), v2(1) }
```

Allgemeine Konformitätsanforderungen

Sperrlisten, die optionale CRL-Erweiterungen *crlExtensions* und/oder optionale CRL-Eintragserweiterungen *crlEntryExtensions* enthalten, sollen das Versionsfeld mit der Version v2 verwenden. Sperrlisten, die keine optionale Erweiterungen enthalten, sollen die Version v1 verwenden und bei der Kodierung das Versionsfeld weglassen.

ISIS-Konformitätsanforderungen

Bei der Erzeugung von Sperrlisten muß die Version v2 benutzt werden.

**Tabelle 7: Implementations-technische Informationen über *version***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	RELEVANZ		
				obligatorisch	verboten	optional
	(BEISPIELE)	(BEISPIELE)	3			
version	1	02 01 01	3			

### V 3.1.3.2. Signatur

#### Zweck

Das Signaturfeld enthält den Bezeichner des Algorithmus, der von der Zertifizierungsstelle zum Signieren der Sperrliste benutzt wird.

#### ASN.1-Definitionen

```
TBSCertList ::= SEQUENCE {
    ...,
    signature      AlgorithmIdentifier,
    ... }
```

```
AlgorithmIdentifier ::= SEQUENCE {
    algorithm      OBJECT IDENTIFIER,
    parameters    ANY DEFINED BY algorithm OPTIONAL }
```

#### Allgemeine Konformitätsanforderungen

Das Signaturfeld *signature* der *tbsCertList*-Struktur muß denselben Algorithmusbezeichner wie im *signatureAlgorithm*-Feld der *CertificateList*-Struktur enthalten.

#### ISIS-Konformitätsanforderungen

Bei der Erstellung von Sperrlisten ist die Benutzung des Sperrlistenfeldes *signature* obligatorisch. Zum Signieren geeignete Algorithmen werden von der Regulierungsbehörde für Telekommunikation und Post im Bundesanzeiger veröffentlicht. Die für die kommenden sechs Jahre (14. Februar 1998 – 14. Februar 2004) als geeignet beurteilten Kryptoalgorithmen sind in der Teilspezifikation "A2 Signatur" von SigI [A2 99] aufgelistet.

**Tabelle 8: Implementations-technische Informationen über *signature***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	RELEVANZ		
				obligatorisch	verboten	optional
	(BEISPIELE)	(BEISPIELE)	1120			
signatureAlgorithm Algorithm Algorithm Parameters	rsaSignatureWithripemd160 SEQUENCE { {1 3 36 3 3 1 2 }, NULL }	30 0A 06 06 2B 24 03 03 01 02 05 00	11			

### V 3.1.3.3. *Namen von Sperrlistenerstellern*

#### Zweck

Das *issuer*-Namensfeld dient zur technischen Identifikation der Instanz bzw. Zertifizierungsstelle, die die betreffende Sperrliste erstellt und signiert hat.

Es sind bei der technischen Namensgebung von Zertifizierungsstellen nur Namen gemäß der X.500-Syntax [ITU-T X.500 97] für *distinguished name*-Typen zugelassen. *Der distinguished name* ist vom Typ *RDNSequence* und somit aus einer Folge von *AttributeType*- und *AttributeValue*-Paaren zusammengesetzt. *AttributeType* wird i.a. durch X.500 festgelegt, und für *AttributeValue* wird der Typ *DirectoryString* (für den unspezifischen Typ *ANY*) verwendet, der seinerseits ein Auswahltyp von *PrintableString*, *TeletexString*, *UniversalString* und *BMPString* ist. Eine Übersicht der möglichen Objektbezeichner für *AttributeType* ist in der folgenden Tabelle gegeben.

#### ASN.1-Definitionen

```

TBSCertList      ::= SEQUENCE {
    ...,
    issuer         Name,
    ... }

Name             ::= CHOICE { RDNSequence }

RDNSequence     ::= SEQUENCE OF RelativeDistinguishedName

RelativeDistinguishedName ::= SET OF AttributeTypeAndValue

AttributeTypeAndValue ::= SEQUENCE {
    type          AttributeType,
    value         AttributeValue }

AttributeType   ::= OBJECT IDENTIFIER

AttributeValue  ::= ANY DEFINED BY AttributeType

DirectoryString ::= CHOICE {
    printableString PrintableString (SIZE (1..maxSize))
    teletexString   TeletexString (SIZE (1..maxSize))
    bmpString       BMPString (SIZE (1..maxSize))
    universalString UniversalString (SIZE (1..maxSize)) }

```

#### Allgemeine Konformitätsanforderungen

Bei der Konstruktion von *DirectoryString* ist stets die restriktivste Auswahl zu treffen und deshalb der minimalste Zeichensatz zur Repräsentation von *AttributeValue* wählen. Die Reihenfolge, in der die Zeichensätze auf ihre konkrete Anwendbarkeit hin geprüft werden sollen, lautet somit: *PrintableString*, *TeletexString*, *BMPString* und *UniversalString*.

Der Name einer Zertifizierungsstelle kann nach [ITU-T X.509 97] auch alternativ oder zusätzlich zum *issuer*-Feld im optionalen *extensions*-Feld unter *issuerAltName* [Abschnitt Z 2.3.9.6] angegeben werden. Im ersten Fall kann das *issuer*-Feld als leere Folge kodiert werden und die *issuerAltName*-Erweiterung muß als *critical*, d.h. als

“wichtige und zu berücksichtigende” Erweiterung gekennzeichnet werden. Weitere Informationen über Namenskonventionen sind in Z 3 Anhang I] zu finden.

### ISIS-Konformitätsanforderungen

Bei der Erstellung von Sperrlisten ist die Benutzung des *issuer*-Feldes obligatorisch. Diese Anforderung ergibt sich aus der Notwendigkeit einer eindeutigen technischen Benennung der Zertifizierungsstelle. Das *issuer*-Feld soll mit dem technischen Namen der Zertifizierungsstelle belegt werden, damit die Konformität zu vielen Anwendungen im internationalen Kontext gewährleistet bleibt. Namen von Zertifizierungsstellen enthalten zumindest die obligatorischen Attribute *organization* und *countryName*. Alle anderen Attribute sind optional.

### Beispiele für technische Namen von Zertifizierungsstellen

(1) Technischer Name der RegTP

OU=Wurzelzertifizierungsstelle, O=RegTP, C=DE

(2) Technischer Name der RegTP mit Acronym

CN=DEPCA, OU=Wurzelzertifizierungsstelle, O=RegTP, C=DE

(3) Technischer Name einer untergeordneten Zertifizierungsstelle

CN=Name-der-ZS, O=Organisation-der-ZS, C=DE

Die Länge der *AttributeValue*-Stringtypen ist durch den Systemparameter *maxSize* festgelegt, dessen Wert für die einzelnen Attribute gemäß der folgenden Tabelle begrenzt ist. Hieraus ergeben sich die in der Längenspalte angegebenen maximalen Längen der Attribute inklusive der ASN.1-Kontrollinformationen, die eine Länge von 11 Bytes haben.

**Tabelle 9: Implementations-technische Informationen über Längen von Attributtypen**

OBJEKTBEZEICHNER		MAXSIZE	LÄNGE	OBJEKTBEZEICHNER		MAXSIZE	LÄNGE
NAME	NUMMER	[BYTES]	[BYTES]	NAME	NUMMER	[BYTES]	[BYTES]
commonName	{ 2 5 4 3 }	64	75	organizationName	{ 2 5 4 10 }	64	75
surName	{ 2 5 4 4 }	32	43	organizationalUnit	{ 2 5 4 11 }	64	75
serialNumber	{ 2 5 4 5 }	3	14	title	{ 2 5 4 12 }	10	21
countryName	{ 2 5 4 6 }	2	13	businessCategory	{ 2 5 4 15 }	32	43
localityName	{ 2 5 4 7 }	32	43	postalCode	{ 2 5 4 17 }	10	21
stateOrProvince	{ 2 5 4 8 }	32	43	givenName	{ 2 5 4 47 }	32	43

Für das *issuer*-Feld bestehend aus nur obligatorischen Attributen ergibt sich aus dem Längensfeld der Tabelle 9 die Maximallänge von  $75+13+2$  (ASN.1-Kontrollinformation) = 90 Bytes und für das *issuer*-Feld bestehend aus allen Attributen der Tabelle 9 von  $75+43+14+13+43+43+75+75+21+43+21+43+4$  (ASN.1-Kontrollinformation) = 513 Bytes.

**Tabelle 10: Implementations-technische Informationen über *issuer***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	RELEVANZ		
	(BEISPIELE)	(BEISPIELE)	513	obligatorisch	verboten	optional
issuer	SEQUENCE OF {	30 53	85			
countryName value	SET OF SEQUENCE { { { 2 5 4 6 }, "DE" } }	31 0B 30 09 06 03 55 04 06 13 02 44 45	13			
organization Name value	SET OF SEQUENCE { { { 2 5 4 10 }, "regtp" } }	31 0E 30 0C 06 03 55 04 0A 13 05 72 65 67 74 70	16			
organizational Unit value	SET OF SEQUENCE { { { 2 5 4 11 }, "Wurzelzertifizierungsstelle" } }	31 24 30 22 06 03 55 04 0B 13 1B 57 75 72 7A 65 6C 7A 65 72 74 69 66 69 7A 69 65 72 75 6E 67 73 73 74 65 6C 6C 65	38			
commonName value	SET OF SEQUENCE { { { 2 5 4 3 }, "DEPCA" } } }	31 0E 30 0C 06 03 55 04 03 13 05 44 45 50 43 41	16			

#### V 3.1.3.4. Datum und Zeitpunkt der Erstellung von Sperrlisten

##### Zweck

Das *thisUpdate*-Datums- und Zeitfeld gibt das Datum und den Zeitpunkt der Erstellung einer Sperrliste an und kann dabei entweder im *UTCTime*- oder im *GeneralizedTime*-Datums- und Zeitformat kodiert werden. Dieser Zeitpunkt kann durch die Standard-ASN.1-Zeittypen *UTCTime* (coordinated universal time, Weltzeit) oder *GeneralizedTime* (allgemeines Datums- und Zeitformat) repräsentiert werden, die Datums- und Zeitangaben bis auf Sekundengenauigkeit sowie die Angabe von Zeitverschiebungen der lokalen gegenüber der Weltzeit gestatten. Die Hauptunterschiede zwischen beiden Formaten bestehen darin, daß mit dem verallgemeinerten Zeittyp kleinere Zeiteinheiten und vollständige Jahreszahlen angegeben werden können.

ASN.1-Definitionen

```

TBSCertList ::= SEQUENCE {
    ...,
    thisUpdate      Time,
    ... }

Time ::= CHOICE {
    utcTime          UTCTime,
    generalizedTime  GeneralizedTime }

```

Allgemeine Konformitätsanforderungen

Zur Kodierung des Zeitpunktes der Erstellung einer Sperrliste ist bis zum Jahr 2049 als Zeittyp stets der Typ *UTCTime* und ab dem Jahr 2050 der Typ *GeneralizedTime* zu benutzen. Zertifizierungsstellen sollen bei der Verwendung eines dieser Typen die Werte von Zeitpunkten in Greenwich-Zeit (GMT, Greenwich Mean Time) bis auf Sekundengenauigkeit ausdrücken, wobei auch die Null-Sekunde zu kodieren ist. Bei der Kodierung der Datums- und Zeitangaben sind für *GeneralizedTime* das Format YYYYMMDDHHMMSSZ und für *UTCTime* das Format YYMMDDHHMMSSZ zu beachten. Die Bedeutung der einzelnen Felder der Datums- und Zeitformate ist in der folgenden Tabelle zusammengefaßt.

Tabelle 11: Bedeutung der Felder in Datums- und Zeitformaten

Datumsangaben		Zeitangaben	
Feld	Bedeutung	Feld	Bedeutung
YYYY	vollständige Jahreszahl, nur bei <i>GeneralizedTime</i>	HH	Stunde 00, 01, ..., 23
YY	letzte zwei Ziffern der Jahreszahl, nur bei <i>UTCTime</i>	MM	Minute 00, 01, ..., 59
MM	Monat 01, 02, ..., 12	SS	Sekunde 00, 01, ..., 59
DD	Tag 01, 02, ..., 31	Z	GMT

Bei der Benutzung des *UTCTime* Typs ist das 2-stellige Jahresfeld YY gemäß der folgenden Konvention (links) zu interpretieren.

Für das 2-stellige Jahresfeld YY gilt nach [MTRUST 96] die folgende Konvention (rechts), die jedoch nicht kompatibel zu [ITU-T X.509 97], [PKIX PRO 97] und [MISPC 97] ist.

$\text{Jahr}(YY) = \begin{cases} 19YY &   YY \in [50,99] \\ 20YY &   YY \in [0,49] \end{cases}$	$\text{Jahr}(YY) = \begin{cases} 19YY &   YY \in [65,99] \\ 20YY &   YY \in [0,64] \end{cases}$
---	---

Die Inkompatibilität betrifft die Zeiträume zwischen 1950 und 1964, sowie zwischen 2050 und 2064. Der erste Zeitraum (1950 bis 1964) bereitet keine Probleme, da es hierfür noch keine Sperrlisten gibt. Sperrlisten, deren Gültigkeitsdauern in den zweiten Jahreszeitraum (2050 bis 2064) fallen, sollten zur Kodierung ebenfalls im *GeneralizedTime*-Format erstellt werden.

### ISIS-Konformitätsanforderungen

Bei der Erzeugung von Sperrlisten muß das allgemeine Datums- und Zeitformat *GeneralizedTime* verwendet werden, bei dessen Kodierung außerdem das Format YYYYMMDDHHMMSSZ genommen werden muß.

**Tabelle 12: Implementations-technische Informationen über *thisUpdate***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	RELEVANZ		
				obligatorisch	verboten	optional
	(BEISPIELE)	(BEISPIELE)	17			
thisUpdate	GeneralizedTime "19980101000000Z"	18 0F 31 39 39 38 30 31 30 31 30 30 30 30 30 30 5A	17			

### V 3.1.3.5. Datum und Zeitpunkt der Erstellung der nächsten Sperrliste

#### Zweck

Das optionale *nextUpdate*-Datums- und Zeitfeld gibt das Datum und den Zeitpunkt der Erstellung der nächsten Sperrliste an und kann dabei entweder im *UTCTime*- oder im *GeneralizedTime*-Datums- und Zeitformat kodiert werden.

#### ASN.1-Definitionen

```

TBSCertList ::= SEQUENCE {
    ...,
    nextUpdate      Time OPTIONAL,
    ... }

Time ::= CHOICE {
    utcTime          UTCTime,
    generalizedTime  GeneralizedTime }

```

#### Allgemeine Konformitätsanforderungen

Aus praktischen Gründen darf eine neue Sperrliste schon vor dem spezifizierten Zeitpunkt erzeugt werden, sie muß aber spätestens bis zu dem angegebenen Zeitpunkt verfügbar sein. Zur Kodierung des Zeitpunktes der

Erstellung einer nächsten Sperrliste ist bis zum Jahr 2049 als Zeittyp stets der Typ *UTCTime* und ab dem Jahr 2050 der Typ *GeneralizedTime* zu benutzen. Zertifizierungsstellen sollen bei der Verwendung eines dieser Typen die Werte von Zeitpunkten in Greenwich-Zeit (GMT, Greenwich Mean Time) bis auf Sekundengenauigkeit ausdrücken, wobei auch die Null-Sekunde zu kodieren ist. Bei der Kodierung der Datums- und Zeitangaben sind für *GeneralizedTime* das Format YYYYMMDDHHMMSSZ und für *UTCTime* das Format YYMMDDHHMMSSZ zu beachten. Die Bedeutung der einzelnen Felder der Datums- und Zeitformate ist in der Tabelle 11 zusammengefaßt. Konforme Zertifizierungsstellen sollen bei der Erstellung von Sperrlisten mit der Version v2 das *nextUpdate*-Datums- und Zeitfeld benutzen.

### ISIS-Konformitätsanforderungen

Bei der Erzeugung von Sperrlisten muß stets der Zeitpunkt der Erstellung einer neuen Sperrliste angegeben werden und hierfür das allgemeine Datums- und Zeitformat *GeneralizedTime* verwendet werden, bei dessen Kodierung außerdem das Format YYYYMMDDHHMMSSZ genommen werden muß.

Es ist verboten, zeitlich überlappende CRLs auszustellen. Somit entspricht *nextUpdate* auch dem Ende der Gültigkeit einer Sperrliste. Der Zeitraum zwischen den durch *thisUpdate* und *nextUpdate* definierten Zeitpunkten muß hinreichend kurz sein, beispielsweise eine Stunde oder ein Tag.

**Tabelle 13: Implementations-technische Informationen über *nextUpdate***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	RELEVANZ		
	(BEISPIELE)	(BEISPIELE)	17	obligatorisch	verboten	optional
<i>nextUpdate</i>	<i>GeneralizedTime</i> "19980102010000Z"	18 0F 31 39 39 38 30 31 30 32 30 31 30 30 30 30 5A	17			

### V 3.1.3.6. Sperrlisteneinträge

#### Zweck

Das optionale *revokedCertificates*-Feld repräsentiert die Liste von widerrufenen bzw. gesperrten Zertifikaten einer Zertifizierungsstelle. Jedes einzelne gesperrte Zertifikat wird in dieser Liste über seine Seriennummer *userCertificate* vom Typ *CertificateSerialNumber* aufgeführt und ist durch die Kombination aus der Seriennummer und dem Namen der Zertifizierungsstelle *issuer* global eindeutig identifiziert. Desweiteren enthält das *revokedCertificates*-Feld den Zeitpunkt der Sperrung *revocationDate* vom Typ *Time*, der – wie in 3.2.3.4 beschrieben – zu behandeln ist. Weitere Informationen über gesperrte Zertifikate können in dem optio-

nen Teilfeld *crlEntryExtensions* als CRL-Eintragserweiterungen spezifiziert werden. Diese zertifikatsspezifischen CRL-Erweiterungen [ITU-T X.509 97, ANSI X9.55 95], für die auch private Erweiterungen definiert werden können, gestatten das Hinzufügen von zusätzlichen Attributen zu CRL-Einträgen.

Derzeit sind für das Internet die als *non-critical* eingestuften optionalen CRL-Eintragserweiterungen *reason code*, *hold instruction code*, *invalidity date* und *certificateIssuer* definiert, die in den folgenden Unterpunkten beschrieben werden.

### ASN.1-Definitionen

```

TBSCertList ::= SEQUENCE {
    ...,
    revokedCertificates . ...
    ... }

revokedCertificates SEQUENCE OF SEQUENCE {
    userCertificate CertificateSerialNumber,
    revocationDate Time,
    crlEntryExtensions Extensions OPTIONAL } OPTIONAL,

CertificateSerialNumber INTEGER

Time ::= CHOICE {
    utcTime UTCTime,
    generalizedTime GeneralizedTime }

Extensions ::= SEQUENCE (1..MAX) OF Extension

Extension ::= SEQUENCE {
    extnId OBJECT IDENTIFIER,
    critical BOOLEAN DEFAULT FALSE,
    extnValue OCTET STRING }

EXTENSION ::= CLASS {
    &id OBJECT IDENTIFIER UNIQUE,
    &ExtType }

WITH SYNTAX {
    SYNTAX &ExtnType
    IDENTIFIED BY &id }

certificateExtension OBJECT IDENTIFIER ::= { 2 5 29 }

id-ce OBJECT IDENTIFIER ::= certificateExtension

```

### Allgemeine Konformitätsanforderungen

Erweiterungen in Sperrlisteneinträgen können als *critical* oder als *non-critical* gekennzeichnet werden. Der CRL-Verifikationsprozeß soll zu einem *fail*-Ergebnis führen, falls er eine als *critical* markierte CRL-Erweiterung nicht verarbeiten kann. Unbekannte und als *non-critical* spezifizierte Erweiterungen dürfen bei der Validierung ignoriert werden.

Die Unterstützung der optionalen und als *non-critical* festgelegten Erweiterungen für das Internet ist für konforme Zertifizierungsstellen und Anwendungen freiwillig. Zertifizierungsstellen sollten jedoch bei der Erstellung von Sperrlisten die *reason code*-Erweiterung benutzen, sofern ihnen diese Information vorliegt.

ISIS-Konformitätsanforderungen

Bei der Erstellung von Sperrlisten ist die Benutzung der *holdInstructionCode*-, und der *invalidityDate*-CRL-Eintragserweiterung verboten. Die Benutzung der *certificateIssuer*- und der *reasonCode*-CRL-Eintragserweiterung ist optional.

**Tabelle 14: Implementations-technische Informationen über CRL-Eintragserweiterung**

ERWEITERUNG	RELEVANZ			KLASSIFIKATION			
	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
reasonCode							
holdInstructionCode							
InvalidityDate							
Certificate issuer							

V 3.1.3.6.1.

ERWEITERUNG DER SPERRLISTEN-EINTRÄGE

## SPERRGRÜNDE

Zweck

Das *reasonCode*-Feld ist eine *non-critical* CRL-Eintragserweiterung, die den Grund für die Sperrung eines Zertifikates anzeigt. Sperrgründe werden syntaktisch durch einen ASN.1-Aufzähltyp beschrieben, der aus einer Menge von benannten Integerwerten besteht. Sperrgründe können von Anwendungen dazu benutzt werden, um zu entscheiden, wie sie auf die Anzeige eines gesperrten Zertifikates in Abhängigkeit von ihren lokalen Sicherheitsrichtlinien reagieren sollen.

ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }
id-ce OBJECT IDENTIFIER ::= { 2 5 29 }
id-ce-cRLReason OBJECT IDENTIFIER ::= { 2 5 29 21 }
cRLReason EXTENSION ::= {
    SYNTAX          CRLReason
    IDENTIFIED BY   id-ce-cRLReason }
CRLReason ::= ENUMERATED {
    unspecified     (0),
    keyCompromise  (1),

```

```

cACompromise          ( 2 ) ,
affiliationChanged    ( 3 ) ,
superseded            ( 4 ) ,
cessationOfOperation ( 5 ) ,
certificateHold       ( 6 ) ,
removeFromCRL        ( 7 ) }

```

### Allgemeine Konformitätsanforderungen

Die Verwendung der *reason codes*-Erweiterung im *crlEntryExtension*-Feld durch Zertifizierungsstellen wird dringend empfohlen. Sie sollte nur dann unterbleiben, falls der Grund für die Sperrung eines Zertifikates nicht bekannt ist, d.h. der Sperrgrund *unspecified* (nicht spezifiziert) sollte nicht benutzt werden. Die Bedeutung der einzelnen Sperrgründe ist in der folgenden Tabelle zusammengefaßt.

**Tabelle 15: Bedeutung von Sperrgründen**

Bezeichner	Wert	Sperrgründe
unspecified	0	nicht-spezifizierter Sperrgrund
keyCompromise	1	Kompromittierung eines privaten Teilnehmerschlüssels
CACompromise	2	Kompromittierung eines privaten Schlüssels einer Zertifizierungsstelle
affiliationChanged	3	Änderung der Namensinformationen eines Zertifikates, ohne daß eine Kompromittierung des privaten Schlüssels vorliegt
superseded	4	Ablauf der Gültigkeit eines Zertifikates, ohne daß eine Kompromittierung des privaten Schlüssels vorliegt
cessationOfOperation	5	Zertifikat wird vor Ablauf seiner Gültigkeit nicht mehr benötigt, ohne daß eine Kompromittierung des privaten Schlüssels vorliegt
certificateHold	6	vorübergehende Sperrung eines Zertifikates
removeFromCRL	7	wird in Verbindung mit delta-CRLs benutzt

ISIS-Konformitätsanforderungen

Die Benutzung des *reasons*-Teilfeld ist bei der Generierung von Sperrlisten optional. Hierbei sind nur die Sperrgründe *keyComromise*, *CACompromise*, *affiliationChanged* und *cessationOfOperation* zulässig. Die Benutzung aller anderen Sperrgründe ist verboten.

**Tabelle 16: Implementations-technische Informationen über *cRLReason***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	RELEVANZ			KLASSIFIKATION		
				obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 12						
cRLReason extnId critical extnValue CACompromise	SEQUENCE { { 2 5 29 21 }, FALSE, OCTET STRING ENUMERATED }	30 0A 06 03 55 1D 15 04 03 0A 01 02	12						

## SPERRINSTRUKTIONEN

Zweck

Das *holdInstructionCode*-Feld ist eine *non-critical* CRL-Erweiterung, die einen registrierten Instruktionsbezeichner enthält, der die auszuführenden Aktionen festlegt, die für temporär gesperrte Zertifikate (*certificate hold*) gelten. Gegenwärtig sind für das Internet die drei Instruktionscodes *holdinstruction-none*, *holdinstruction-callissuer* und *holdinstruction-reject* festgelegt.

ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue      OCTET STRING }

id-ce OBJECT IDENTIFIER ::= { 2 5 29 }

id-ce-holdInstructionCode OBJECT IDENTIFIER ::= { 2 5 29 23 }

holdInstructionCode EXTENSION ::= {
    SYNTAX          HoldInstructionCode
    IDENTIFIED BY  id-ce-holdInstructionCode }

HoldInstructionCode ::= CHOICE {
    id-holdinstruction-none    OBJECT IDENTIFIER,
    id-holdinstruction-callissuer OBJECT IDENTIFIER,

```

<code>id-holdinstruction-reject</code>	OBJECT IDENTIFIER }	
<b><code>id-holdInstruction</code></b>	OBJECT IDENTIFIER ::=	
{ 1 2 840 10040 2 }		
<b><code>id-holdinstruction-none</code></b>	OBJECT IDENTIFIER ::=	
{1 2 840 10040 2 1}		
<b><code>id-holdinstruction-callissuer</code></b>	OBJECT IDENTIFIER ::=	
{1 2 840 10040 2 2}		
<b><code>id-holdinstruction-reject</code></b>	OBJECT IDENTIFIER ::=	
{1 2 840 10040 2 3}		

#### Allgemeine Konformitätsanforderungen

Konforme Anwendungen, die diese Erweiterung verarbeiten, sollen beim Empfang einer *callissuer*-Instruktion entweder den Zertifikatsersteller kontaktieren oder das Zertifikat zurückweisen. Desweiteren sollen sie beim Empfang einer *reject*-Instruktion das Zertifikat zurückweisen. Die *none*-Instruktion ist semantisch äquivalent zu einem fehlenden *holdInstructionCode*-Feld und wird für die Internet-PKI verworfen.

#### ISIS-Konformitätsanforderungen

Die Benutzung der *holdInstructionCode*-Eintragsweiterung ist bei der Generierung von Sperrlisten verboten, da auch die Benutzung des *certificateHold*-Sperrgrundes und damit eine zeitweilige Sperrung von Zertifikaten verboten ist. Zum einen darf nach [SigG 97 §8(1)] und [SigV 97 §9(3)] eine Sperrung nicht rückgängig gemacht werden, d.h. ein Zertifikat darf auch nicht vorübergehend gesperrt und später wieder reaktiviert werden. Zum anderen werden zur Suspendierung zeitlich überlappende CRLs benötigt, um eine kurzfristige Suspendierung zu ermöglichen. Zeitlich überlappende CRLs auszustellen ist jedoch verboten (siehe 3.1.3.5). Wenn es zu einem Zeitpunkt mehrere gültige Sperrlisten gibt, so kann nicht sichergestellt werden, daß der Verifizierer einer digitalen Signatur die neueste CRL verwendet

**Tabelle 17: Implementations-technische Informationen über *holdInstructionCode***

BEZEICHNER	WERTEBEREICH EINZELWERTE	HEXADEZIMALCODE	LÄNGE	RELEVANZ			KLASSIFIKATION				
				obligatorisch	verboten	optional	Standardweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung	
	(BEISPIELE)	(BEISPIELE)	[BYTES]								
<code>holdInstructionCode</code>											

## SPERRDATUM

Zweck

Das *invalidityDate*-Feld ist eine *non-critical* CRL-Eintragsweiterung, die den Zeitpunkt und das Datum des Bekanntwerdens eines kompromittierten privaten Schlüssels oder eines anderweitigen Ungültigwerdens eines Zertifikates anzeigt. Dieses Feld ist nicht zu verwechseln mit dem CRL-Feld *revocationDate* (siehe Abschnitt V 3.1.3.6), das für jedes widerrufenes Zertifikat den Sperrzeitpunkt angibt.

ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

id-ce OBJECT IDENTIFIER ::= { 2 5 29 }

id-ce-invalidityDate OBJECT IDENTIFIER ::= { 2 5 29 24 }

invalidityDate EXTENSION ::= {
    SYNTAX          GeneralizedTime
    IDENTIFIED BY  id-ce-invalidityDate }

```

Allgemeine Konformitätsanforderungen

Zeitlich betrachtet kann der Wert *invalidityDate*-Feldes vor dem in *revocationDate* (siehe Abschnitt V3.1.3.6) angegebenen Zeitpunkt liegen. Bei der Kodierung des Feldes im *GeneralizedTime* Zeit- und Datumsformat sind die im Abschnitt V 3.1.3.4 angegebenen Konformitätsanforderungen zu beachten.

ISIS-Konformitätsanforderungen

Bei der Erzeugung von Sperrlisten ist die Benutzung des *invalidityDate*-Feldes verboten.

**Tabelle 18: Implementations-technische Informationen über *invalidityDate***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	RELEVANZ			KLASSIFIKATION			
				obligatorisch	verboten	optional	Standardweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES]							
invalidityDate										

## IDENTIFIZIERUNG DES ZERTIFIKATERSTELLERS

Zweck

Das *certificateIssuer*-Feld ist eine CRL-Eintragserweiterung, die den Ersteller eines gesperrten Zertifikates in einer indirekten CRL identifiziert. *IndirectCRLs* sind Sperrlisten, die nicht nur gesperrte Zertifikate einer Zertifizierungsstelle enthalten, sondern Sperrinformationen verschiedener Zertifizierungsstellen zusammenfassen. Zur Kenntlichmachung einer *indirectCRL* muß das Teilfeld *indirectCRL* in der *issuingDistributionPoint* CRL-Erweiterung gesetzt werden. Falls das *certificateIssuer*-Feld nicht im ersten Eintrag einer indirekten CRL gesetzt ist, so nimmt es per Voreinstellung den durch das *issuer*-Feld gegebenen Wert für den CRL-Ersteller an. Fehlt dieses Feld in den nachfolgenden Einträgen einer indirekten CRL, so wird der Wert für den CRL-Ersteller durch den Wert des vorangegangenen Eintrages bestimmt.

ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

certificateIssuer EXTENSION ::= {
    SYNTAX          CertificateIssuer
    IDENTIFIED BY   id-ce-certificateIssuer }

id-ce-certificateIssuer OBJECT IDENTIFIER ::= { 2 5 29 29 }

CertificateIssuer ::= GeneralNames

```

Allgemeine Konformitätsanforderungen

Konforme Zertifizierungsstellen, die diese Erweiterung benutzen, sollten sie stets als *critical* kennzeichnen.

Konforme Anwendungen sollten diese Erweiterung nicht ignorieren, weil sie ansonsten die CRL-Einträge nicht eindeutig an bestimmte Zertifikate zuordnen können.

ISIS-Konformitätsanforderungen

Die *certificateIssuer*-Erweiterung ist optional. Das *certificateIssuer*-Feld wird im Falle der Übernahme von Zertifikaten durch eine andere Zertifizierungsstelle benötigt [SigG 97, §13 (4)], da ansonsten die Gefahr besteht, daß Seriennummern doppelt vorkommen.

**Tabelle 19: Implementations-technische Informationen über *certificateIssuer***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	RELEVANZ			KLASSIFIKATION		
				obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES]						
CertificateIssuer									

### V 3.1.3.7. Sperrlistenerweiterungen

#### Zweck

Das CRL-Erweiterungsfeld *crlExtensions* besteht aus einer Folge von CRL-Erweiterungen [ITU-T X.509 97, ANSI X9.55 95], die auch als private Erweiterungen definiert werden können und die das Hinzufügen von zusätzlichen Attributen zu Sperrlisten gestatten.

In [ITU-T X.509 97, RFC 2459 99] sind die als non-critical eingestuft optionalen CRL-Erweiterungen *authority key identifier*, *issuer alternative name*, *CRL number*, *issuing distribution point* und die als critical eingestufte CRL-Erweiterung *delta CRL indicator* definiert, die in den folgenden Unterpunkten beschrieben werden.

OpenCDP definiert darüber hinaus die CRL-Erweiterung *cRLScope*, die im Abschnitt V 3.2.2.1 beschrieben ist.

#### Allgemeine Konformitätsanforderungen

Das *crlExtensions*-Teilfeld darf nur in Verbindung mit Sperrlisten der Version v2 benutzt werden.

Erweiterungen in Sperrlisten können als *critical* oder als *non-critical* gekennzeichnet werden. Der CRL-Verifikationsprozeß soll zu einem *fail*-Ergebnis führen, falls er eine als *critical* markierte CRL-Erweiterung nicht verarbeiten kann. Unbekannte und als *non-critical* spezifizierte Erweiterungen dürfen bei der Validierung ignoriert werden.

Die Unterstützung der optionalen und als *non-critical* festgelegten Erweiterungen für das Internet ist für konforme Zertifizierungsstellen und Anwendungen freiwillig. Konforme Zertifizierungsstellen sollten jedoch bei der Erstellung von Sperrlisten die *crlNumber*-Erweiterung benutzen und in allen erstellten Sperrlisten einsetzen. Konforme Anwendungen sollten die als *critical* gekennzeichneten Erweiterungen unterstützen können.

#### ISIS-Konformitätsanforderungen

Bei der Erstellung von Sperrlisten müssen die als *non-critical* eingestuft optionalen CRL-Erweiterungen *authority key identifier* und *CRL number* verwendet werden. Die Benutzung von *issuingDistributionPoint* ist verboten, während *deltaCRLIndicator* und *issuerAlternativeName* optional sind.

**Tabelle 20: Implementations-technische Informationen über Erweiterungen**

ERWEITERUNG	RELEVANZ			KLASSIFIKATION			
	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
authority key identifier							
issuer alternative name							
CRL number							
issuing distribution point							
delta CRL indicator							

### V 3.1.3.7.1. IDENTIFIZIERUNG VON SIGNATURSCHLÜSSELN VON ZERTIFIZIERUNGSSTELLEN

#### Zweck

Das *authorityKeyIdentifier*-Erweiterungsfeld dient zur Identifizierung eines bestimmten öffentlichen Schlüssels einer Zertifizierungsstelle, der zum Signieren einer Sperrliste verwendet wurde. Die Erweiterung wird dann verwendet, wenn eine Zertifizierungsstelle mehrere Signaturschlüssel – sei es als gleichzeitig aktive Schlüssel oder zum Schlüsselwechsel – besitzt. Die Identifizierung kann entweder durch den Schlüsselnamen im *keyIdentifier*-Teilfeld oder durch den Namen der Zertifizierungsstelle im *authorityCertIssuer*-Teilfeld und die Seriennummer im *authorityCertSerialNumber*-Teilfeld erfolgen.

Die Kombination *authorityCertIssuer* und *authorityCertSerialNumber* identifiziert eindeutig ein bestimmtes Zertifikat einer Zertifizierungsstelle. Der *keyIdentifier* kennzeichnet den öffentlichen Schlüssel der Zertifizierungsstelle und somit nicht nur eine, sondern gegebenenfalls eine Reihe von Zertifikaten. Damit erlaubt die Verwendung des *keyIdentifier* eine größere Flexibilität zum Auffinden eines Zertifizierungspfades. Außerdem müssen die Zertifikate, die den *keyIdentifier* im *authorityKeyIdentifier*-Erweiterungsfeld benutzen, nicht zurückgezogen werden, wenn die Zertifizierungsstelle sich bei gleichbleibendem Schlüssel ein neues Zertifikat ausstellen läßt.

Andererseits ist die Flexibilität zum Auffinden eines Zertifizierungspfades nicht immer gewünscht. Verfügt eine Zertifizierungsstelle über mehrere Zertifikate für den gleichen Schlüs-

sel, die aber beispielsweise verschiedene Haftungsgrenzen beinhalten, so ist es erforderlich, nicht nur den öffentlichen Schlüssel sondern genau dasjenige Zertifikat der Zertifizierungsstelle zu referenzieren, das für den jeweiligen Teilnehmer gültig ist.

### ASN.1-Definitionen

```
Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

authorityKeyIdentifier EXTENSION ::= {
    SYNTAX          AuthorityKeyIdentifier
    IDENTIFIED BY   id-ce-authorityKeyIdentifier }

id-ce-authorityKeyIdentifier OBJECT IDENTIFIER ::= { 2 5 29 35 }

AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier    [0] KeyIdentifier OPTIONAL,
    authorityCertIssuer [1] GeneralNames OPTIONAL,
    authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL }

KeyIdentifier ::= OCTET STRING
```

### Allgemeine Konformitätsanforderungen

Entweder sind die beiden *authorityCertIssuer*- und *authorityCertSerialNumber*-Teilfelder in Zertifikate zu integrieren oder beide wegzulassen. Im zweiten Fall muß stattdessen das *keyIdentifier*-Teilfeld eingebaut werden.

Falls beide Identifizierungsmethoden benutzt werden, sollte die Zertifizierungsstelle deren Konsistenz sicherstellen. Ein Schlüsselbezeichner soll bezüglich aller Schlüsselbezeichner, die eine Zertifizierungsstelle für einen Zertifikatsinhaber benutzt, eindeutig sein.

Systeme sollten die Fähigkeit besitzen, Zertifizierungspfade finden und validieren zu können, wenn die ausstellende Zertifizierungsstelle mehrere Signaturschlüssel besitzt. Sie sollten eine der beiden Identifikationsmethoden zum Auffinden von Zertifizierungspfaden unterstützen.

ISIS-Konformitätsanforderungen

Die Benutzung dieser Erweiterung ist in allen Sperrlisten obligatorisch und sie muß als *non-critical* gekennzeichnet werden. Außerdem soll als Schlüsselidentifizierungsmethode die Verwendung der beiden *authorityCertIssuer*- und *authorityCertSerialNumber*-Teilfelder unterstützt werden, die ein bestimmtes Zertifikat der Zertifizierungsstelle eindeutig identifiziert. Dabei muß im *authorityCertIssuer*-Teilfeld zumindest der *issuer*-Name des Zertifikaterstellers vom Typ *directoryName* angegeben werden.

**Tabelle 21: Implementations-technische Informationen über authorityKeyIdentifier**

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	RELEVANZ			KLASSIFIKATION		
				obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 280						
authorityKey Identifier	SEQUENCE { ExtnId { 2 5 29 35 }, Critical FALSE, ExtnValue OCTET STRING	30 68 06 03 55 1D 23	106						
AuthCertIssuer	SEQUENCE { [1] SEQUENCE OF {	80 57							
DirectoryName	[4] SEQUENCE OF {	84 55 30 53							
CountryName	SET OF SEQUENCE { { 2 5 4 6 }, "DE" }	31 0B 30 09 06 03 55 04 06 13 02 44 45							
Organization Name value	SET OF SEQUENCE { { 2 5 4 10 }, "REGTP" }	31 0E 30 0C 06 03 55 04 0A 13 05 52 45 47							
Organizational Unit Value	SET OF SEQUENCE { { 2 5 4 11 }, "Wurzelzertifizierungsstelle" }	54 50 31 24 30 22 06 03 55 04 0B 13 1B 57 75 72 7A 65 6C 7A 65 72 74 69 66 69 7A 69 65 72 75 6E 67 73 73 74 65 6C 6C 65							
CommonName Value	SET OF SEQUENCE { { 2 5 4 3 }, "DEPCA" } }	31 0E 30 0C 06 03 55 04 03 13 05 44 45 50 43 41							
AuthCertSerNum	[2] 1 }	82 03 02 01 01							

## V 3.1.3.7.2. ALTERNATIVE NAMEN VON SPERRLISTENERSTELLERN

Zweck

Das *issuerAltName*-Erweiterungsfeld enthält einen oder mehrere alternative Namen für den Ersteller einer Sperrliste, durch die zusätzliche Identitäten an den Sperrlistenersteller gebunden werden.

Neben dem *distinguished name* des Sperrlistenerstellers können im alternativen Namensfeld des Ausstellers zusätzliche Adreßinformationen zur Erreichbarkeit im Internet abgelegt werden. Dazu gehören insbesondere die Angabe einer Internetadresse für elektronische Post, sowie Angaben über den DNS-Namen des Sperrlistenerstellers (DNS, domain name system).

Die Adresse für elektronische Post (rfc822) sollte eine symbolische Mailadresse sein, die es einem Teilnehmer ermöglicht, Kontakt zur Zertifizierungsstelle aufzunehmen. Es sollen an dieser Stelle keine persönlichen Mailadressen von Mitarbeitern verwendet werden.

Der DNS-Name der Zertifizierungsstelle sollte der registrierte Domain-Name der Zertifizierungsstelle sein. Über diesen Namen können Anwendungen die Adressen zusätzlicher Dienste und Protokolle der Zertifizierungsstelle ermitteln, wie beispielsweise die Adresse eines X.500-Verzeichnisdienstes. Diese Vorgehensweise ist eine Alternative zur Verwendung des globalen X.500 Verzeichnisdienst. Mit etablierten Verfahren (beispielsweise [RFC 2052 96]) können die Adressen der gewünschten Dienste aus dem angegebenen DNS-Namen abgeleitet werden. Zu beachten ist, daß Informationen über den DNS-Namen einer Zertifizierungsstelle auch im *distinguished name* der Zertifizierungsstelle angegeben sein können. Dies geschieht durch die Definition des sog. *DC*-Bezeichners (DC, domain component, Teilname eines Domännennamens) für *distinguished names* [RFC 2247 98]).

ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue      OCTET STRING }

issuerAltName EXTENSION ::= {
    SYNTAX          IssuerAltName
    IDENTIFIED BY  id-ce-issuerAltName }

id-ce-issuerAltName OBJECT IDENTIFIER ::= { 2 5 29 18 }

IssuerAltName ::= GeneralNames

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
    otherName      [0] OTHER-NAME,
    rfc822Name     [1] IA5String,
    dNSName        [2] IA5String,
    x400Address    [3] ORAddress,
    directoryName  [4] Name,
    ediPartyName   [5] EDIPartyName,
    uniformResourceId [6] IA5String,
  }

```

```

    ipAddress          [7] OCTET STRING,
    registeredID       [8] OBJECT IDENTIFIER }

OTHER-NAME ::= SEQUENCE {
    type-id            OBJECT IDENTIFIER,
    value              [0] EXPLICIT ANY DEFINED BY type-id }

EDIPartyName ::= SEQUENCE {
    nameAssigner       [0] DirectoryString OPTIONAL,
    partyName          [1] DirectoryString }

```

### Allgemeine Konformitätsanforderungen

Die Benutzung des String-Platzhaltersymbols "\*" in Namenstypen des *issuerAltName*-Erweiterungsfeldes ist verboten.

### ISIS-Konformitätsanforderungen

Die optionale *issuerAltName*-Erweiterung kann bei der Erstellung von Sperrlisten benutzt werden und ist dabei als *non-critical* zu kennzeichnen. Diese Erweiterung besitzt jedoch keine Bedeutung für die technische Identifikation des Erstellers der Sperrliste, sondern sie bindet lediglich weitere Merkmale (z.B. E-Mail-Adressen) an ihn.

BEISPIELE FÜR DIE BENUTZUNG DER *ISSUERALTNAME*-ERWEITERUNG:

- (1) Mailadresse: *rfc822Name*: rootca@regtp.de
- (2) X.500-Verzeichnisdienstname mit Angabe der E-Mail Adresse der Zertifizierungsstelle:  
*directoryName*: CN=Verzeichnisdienst, EMAIL=ca@cert.de, O=ZS1, C=DE

**Tabelle 22: Implementations-technische Informationen über *issuerAltName***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	RELEVANZ						
				obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 500							
<i>issuerAltName</i> <i>extnId</i> <i>critical</i> <i>extnValue</i> <i>rfc822Name</i>	SEQUENCE { { 2 5 29 18 }, FALSE, OCTET STRING SEQUENCE OF{[1] "ca@cert.de" } }	30 15 06 03 55 1D 12  04 0E 30 0C 81 0A 63 61 40 63 65 72 74 2E 64 65	23							

## V 3.1.3.7.3. SPERRLISTENNUMMERN

Zweck

Das *cRLNumber*-Feld ist eine *non-critical* CRL-Erweiterung, die eine um den Wert 1 anwachsende Folgenummer für jede von einer bestimmtem Zertifizierungsstelle erzeugte Sperrliste enthält. Sperrlisten können entweder über X.500-Dienste, mit Hilfe der *crlDistributionPoints*-Erweiterung oder über den OpenCDP-Mechanismus (siehe Abschnitt 3.2.2) den Teilnehmern zur Verfügung gestellt werden. Diese Erweiterung gestattet den Teilnehmern eine einfache Unterscheidungsmöglichkeit darüber, ob eine bestimmte CRL eine andere CRL ersetzt.

ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

cRLNumber EXTENSION ::= {
    SYNTAX          CRLNumber
    IDENTIFIED BY   id-ce-cRLNumber }

id-ce-cRLNumber OBJECT IDENTIFIER ::= { 2 5 29 20 }

CRLNumber ::= INTEGER (0..MAX)
  
```

Allgemeine Konformitätsanforderungen

Konforme Zertifizierungsstellen sollen diese Erweiterung in allen Sperrlisten verwenden.

ISIS-Konformitätsanforderungen

Bei der Erstellung von Sperrlisten muß die *cRLNumber*-Erweiterung benutzt und dabei als *non-critical* gekennzeichnet werden. Der Systemparameter MAX ist auf den Wert  $2^{8*15-1}-1$  begrenzt.

**Tabelle 23: Implementations-technische Informationen über *cRLNumber***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	RELEVANZ			KLASSIFIKATION		
				obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 26						
cRLNumber extnId critical extnValue	SEQUENCE { { 2 5 29 20 }, FALSE, OCTET STRING 1}	30 0A 06 03 55 1D 14 04 03 02 01 01	12						

## V 3.1.3.7.4. IDENTIFIKATION DER QUELLEN VON SPERRLISTEN

Zweck

Das *issuingDistributionPoint*-Feld ist eine *critical* CRL-Erweiterung, die den Verteilungspunkt (CRL distribution point) einer bestimmten Sperrliste identifiziert und anzeigt, ob die Sperrliste nur Endanwenderzertifikate, oder nur CA-Zertifikate oder nur Zertifikate für eine begrenzte Menge von Sperrgründen enthält.

Sperrlisten werden mit dem privaten Signaturschlüssel der betreffenden Zertifizierungsstelle signiert. *CRL-distribution points* besitzen keine eigenen Schlüsselpaare. Falls CRLs in einem X.500-Verzeichnisdienst gespeichert werden, so beinhaltet das *distributionPoint*-Feld im *issuingDistributionPoint* die entsprechende Adresse, die nicht mit dem X.500-Eintrag der Zertifizierungsstelle übereinstimmen muß. Zertifizierungsstellen können das *issuingDistributionPoint*-Feld zur Partitionierung von Sperrlisten einsetzen.

ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

issuingDistributionPoint EXTENSION ::= {
    SYNTAX          IssuingDistributionPoint
    IDENTIFIED BY  id-ce-issuingDistributionPoint }

id-ce-issuingDistributionPoint OBJECT IDENTIFIER ::=
    { 2 5 29 28 }

IssuingDistributionPoint ::= SEQUENCE {
    distributionPoint      [0] DistributionPointName OPTIONAL,
    onlyContainsUserCerts [1] BOOLEAN DEFAULT FALSE,
    onlyContainsCACerts   [2] BOOLEAN DEFAULT FALSE,
    onlySomeReasons       [3] ReasonFlags OPTIONAL,
    indirectCRL           [4] BOOLEAN DEFAULT FALSE }

DistributionPointName ::= CHOICE {
    fullName             [0] GeneralNames,
    nameRelativeToCRLIssuer [1] RelativeDistinguishedName }

ReasonFlags ::= Bit String {
    unused                (0),
    keyCompromise         (1),
    cACompromise          (2),
    affiliationChanged    (3),
    superseded            (4),
    cessationOfOperation  (5),
    certificateHold       (6) }

```

Allgemeine Konformitätsanforderungen

Obwohl diese Erweiterung als *critical* gekennzeichnet ist, muß sie nicht von Anwenderinfrastrukturen unterstützt werden.

Zertifizierungsstellen können das *issuingDistributionPoint*-Feld zur Partitionierung von Sperrlisten einsetzen. In diesem Fall müssen z.B. alle mit dem Sperrgrund *keyCompromise* gesperrten Zertifikate in einem bestimmten *distributionPoint* und alle mit einem anderen Sperrgrund gesperrten Zertifikate in einem anderen *distributionPoint* abgelegt sein.

Die mit *distributionPoints* verknüpften Sperrgründe müssen bei segmentierten Sperrlisten im *onlySomeReasons*-, *onlyContainUserCerts*- oder *onlyContainCACerts*-Teilfeld spezifiziert werden. Ansonsten, d.h. falls eine vollständige Sperrliste für alle Sperrgründe vorliegt, können diese Teilfelder weggelassen werden.

Falls das *issuingDistributionPoint*-Feld eine URL enthält, so soll diese als ein Verweis auf die aktuelle CRL der ausstellenden Zertifizierungsstelle betrachtet werden. Als Namensformate dienen hierbei die URI-Formate ftp, http, mailto und ldap, die über einen absoluten Pfadnamen den betreffenden Rechner angeben müssen.

### ISIS-Konformitätsanforderungen

Bei der Erstellung von Sperrlisten ist die Benutzung der *issuingDistributionPoint*-Erweiterung verboten.

**Tabelle 24: Implementations-technische Informationen über *issuingDistributionPoint***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	RELEVANZ				KLASSIFIKATION			
				obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung	
	(BEISPIELE)	(BEISPIELE)	[BYTES]								
IssuingDistributionPoint											

#### V 3.1.3.7.5. INDIKATOR VON SPERRLISTENÄNDERUNGEN

##### Zweck

Das *deltaCRLIndicator*-Feld ist eine *critical* CRL-Erweiterung, die eine Sperrlistenänderung (delta-CRL) identifiziert. Durch die Benutzung dieser Erweiterung kann die Effektivität von Anwendungen erhöht werden, die Sperrinformationen in einem von der CRL-Struktur abweichendem Format speichern, weil hierbei nur die tatsächlichen Änderungen in der lokalen Datenbank durchgeführt und die unveränderten und lokal bereits gespeicherten Informationen ignoriert werden können.

Durch das Feld *BaseCRLNumber* vom Typ *CRLNumber* wird die Folgenummer der CRL identifiziert, die als Ausgangspunkt für die Erzeugung dieser *delta-CRL* genommen wurde.

Die *delta-CRL* enthält die Änderungen zwischen der Ausgangs-CRL und der aktuellen CRL, die zusammen mit der *delta-CRL* erstellt wird.

Der delta-CRL-Mechanismus ist in der Abbildung 1 veranschaulicht.

### ASN.1-Definitionen

```

Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

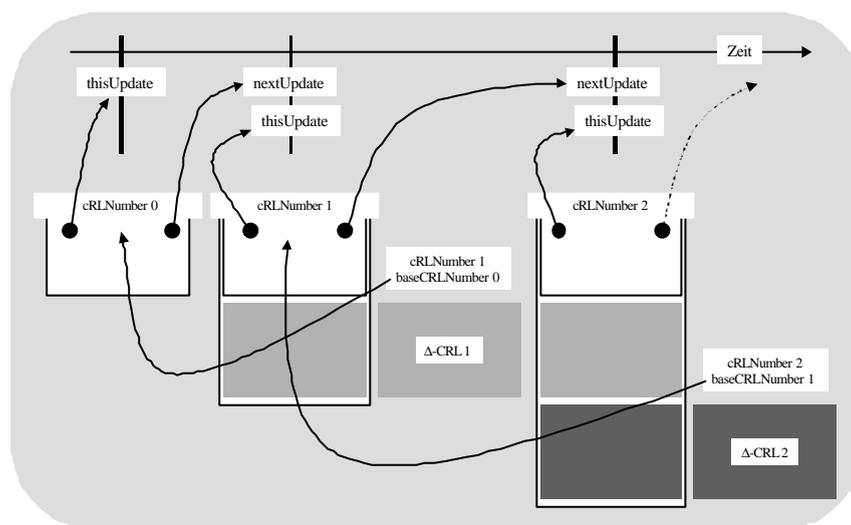
deltaCRLIndicator EXTENSION ::= {
    SYNTAX          BaseCRLNumber
    IDENTIFIED BY  id-ce-deltaCRLIndicator }

id-ce-deltaCRLIndicator OBJECT IDENTIFIER ::=
    { 2 5 29 27 }

BaseCRLNumber ::= CRLNumber

CRLNumber ::= INTEGER (0..MAX)
  
```

Abbildung 1: *deltaCRL*-Mechanismus



### Allgemeine Konformitätsanforderungen

Zertifizierungsstellen dürfen selbst darüber entscheiden, ob sie *delta-CRLs* erstellen oder nicht. Wenn eine *delta-CRL* ausgestellt wird, so muß die Zertifizierungsstelle auch eine vollständige CRL erstellen. Der Wert der CRL-Folgenummern *cRLNumber* (siehe Abschnitt 3.1.3.7.3) in CRLs und *delta-CRLs* muß identisch sein.

CRL-Benutzer, die eine lokal gespeicherte CRL mit Hilfe von *delta-CRLs* aktualisieren, sollen eine konstruierte CRL als unvollständig betrachten, falls sie eine *delta-CRL* empfangen, deren Folgenummer um mehr als 1 größer ist als die der zuletzt empfangenen *delta-CRL*.

ISIS-Konformitätsanforderungen

Die Benutzung der *deltaCRLIndicator*-CRL-Erweiterung bei der Erstellung von Sperrlisten ist optional. ISIS-konforme Systeme und Anwendungen müssen diese Erweiterung erkennen und verarbeiten können, sofern sie den deltaCRL-Mechanismus benutzen.

**Tabelle 25: Implementations-technische Informationen über *deltaCRLIndicator***

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	RELEVANZ			KLASSIFIKATION		
				obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung
	(BEISPIELE)	(BEISPIELE)	[BYTES] 29						
deltaCRLIndicator extnId critical cRLNumber	SEQUENCE { { 2 5 29 27 }, TRUE, OCTET STRING 1 }	30 0D 06 03 55 1D 1B 01 01 FF 04 03 02 01 01	15						

## V 3.2. Verwaltung und Bereitstellung von Sperrlisten

Zertifizierungsstellen müssen Sperrlisten on-line zur Verfügung stellen und diese in regelmäßigen Abständen aktualisieren. Endanwender können diese Sperrlisten entweder lokal in ihrer Umgebung speichern und periodisch aktualisieren oder bei jeder Verifikation on-line auf die aktuelle Sperrliste zugreifen.

Entscheidet sich ein Anwender, Sperrlisten lokal zu speichern, so ist er selber dafür verantwortlich, stets im Besitz der für ihn hinreichend aktuellen Sperrlisten zu sein. In diesem Fall sollten lokal Sperrlisten für alle oder jedenfalls einige der Zertifizierungsstellen gespeichert werden. Auf diese Weise ist es möglich, auch bei Rechnern, die nicht permanent mit dem öffentlichen Netz verbunden sind, Signaturen mit hinreichend hoher Sicherheit zu verifizieren.

Um bei der Verifikation einer digitalen Signatur eine aktuelle Sperrliste abrufen zu können, gibt es folgende Varianten, die derzeit in der PKIX Arbeitsgruppe für die Internet "X.509 Public Key Infrastructure" vorgeschlagen sind:

- CDP (Certificate Distribution Point)
- OpenCDP (Open CRL Distribution Process)
- OCSP (Online Certificate Status Protocol) unter Verwendung von CRLs

Alle drei Varianten basieren auf CRLs, d.h. sie geben nur Informationen über zurückgezogene Zertifikate. Damit können sie nicht den nach dem Signaturgesetz geforderten Verzeichnisdienst ersetzen, der über alle ausgestellten Zertifikate Auskunft geben können muß.

Mit der Verwendung von Sperrlisten geht der Anwender stets das Risiko ein, daß bei einer Verifikation eine Unterschrift irrtümlich als gültig anerkannt wird, sofern der Sperrzeitpunkt innerhalb des Gültigkeitszeitraumes der Sperrliste liegt. Diese Unsicherheit ist durch die Anwenderinfrastruktur dem Benutzer entsprechend kenntlich zu machen. Geeignete Maßnahmen sind in diesem Zusammenhang die Unterrichtung des Teilnehmers, sowie die Fixierung des Problems in den zugehörigen Sicherheitsrichtlinien.

### V 3.2.1. CDP (CERTIFICATE DISTRIBUTION POINT)

Der Certificate Distribution Point (CDP) wurde schon in Abschnitt Z 2.3.9.9 erläutert. Hierbei handelt es sich um die X.509v3 Zertifikatserweiterung *cRLDistributionPoints*, die zur Beschaffung von Sperrlisten dient. Im Zertifikat ist bei dieser Variante direkt der Verweis auf die zugehörige Sperrliste enthalten, in der das Zertifikat eingetragen wird, wenn es zurückgezogen werden sollte. Diese Lösung ist insofern inflexibel, als die Information statisch in das Zertifikat eingebunden ist und somit das Zertifikat zurückgezogen werden muß, wenn sich die Adresse ändert, von der die Sperrliste verfügbar ist.

Gedacht war diese Zertifikatserweiterung außerdem für eine Segmentierung von CRLs, jedoch beinhaltet CDP kein Segmentierkriterium und ist somit zur Segmentierung ungeeignet.

An dieser Stelle sei darauf hingewiesen, daß zu dem Thema “cRLDistributionPoints” das US-Patent 5,699,431 von Entrust Technologies Inc. existiert, das aber weltweit und gebührenfrei benutzt werden darf.

### ASN.1-Definitionen

```
Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

cRLDistributionPoints EXTENSION ::= {
    SYNTAX          CRLDistPointsSyntax
    IDENTIFIED BY  id-ce-cRLDistributionPoints }

id-ce-cRLDistributionPoints OBJECT IDENTIFIER ::= { 2 5 29 31 }

CRLDistPointsSyntax ::= SEQUENCE SIZE (1..MAX) OF
    DistributionPoint

DistributionPoint ::= SEQUENCE {
    distributionPoint [0] DistributionPointName OPTIONAL,
    reasons           [1] ReasonFlags OPTIONAL,
    cRLIssuer         [2] GeneralNames OPTIONAL }

DistributionPointName ::= CHOICE {
    fullName          [0] GeneralNames,
    nameRelativeToCRLIssuer [1] RelativeDistinguishedName }

ReasonFlags ::= BIT STRING {
    unused            (0),
    keyCompromise    (1),
    cACompromise     (2),
    affiliationChanged (3),
    superseded       (4),
    cessationOfOperation (5),
    certificateHold   (6) }
```

### Allgemeine Konformitätsanforderungen

Falls der *DistributionPointName*-Name einer CRL-Verteilungsstelle im *URI*-Format angegeben wird, so ist die *URI* als ein Pointer auf die aktuelle Sperrliste anzusehen, deren zugehörigen Sperrgründe durch das Feld *reasons* und deren Ersteller durch das Feld *cRLIssuer* gekennzeichnet werden können. Die Werte im *URI*-Format (http, ldap, ftp) unterliegen denselben Einschränkungen wie für *subjectAltName*-Erweiterungen. Falls das optionale *reasons*-Teilfeld in der Erweiterung nicht verwendet wird, so soll die Sperrliste gesperrte Zertifikate für alle Sperrgründe enthalten. Falls das optionale *cRLIssuer*-Teilfeld nicht benutzt wird, so soll die Sperrliste von derjenigen Zertifizierungsstelle erstellt werden, die das Zertifikat erzeugt hat.

### ISIS-Konformitätsanforderungen

Die optionale *cRLDistributionPoints*-Erweiterung muß als *non-critical* markiert werden, so daß statt der Benutzung von Sperrlisten auch andere Mechanismen wie z.B. On-line Prüfdienste zur Verifikation herangezogen werden können. Diese Methode zur Beschaffung von Sperrlisten kann durch Zertifizierungsstellen und Anwendungen unterstützt werden. Die Benutzung des *reasons*-Teilfeld ist bei der Generierung von Zertifikaten verboten. Eine Segmentierung der Sperrliste über die *cRLDistributionPoints*-Erweiterung ist verboten, sofern die

Sperrliste nicht selbst das Segmentierkriterium enthält (siehe Abschnitt 3.2.2.1). Verwendet eine Zertifizierungsstelle mehrere verschiedene URIs, so müssen diese alle auf die gleiche Information zeigen. Wenn der CRL-Herausgeber nicht die Zertifizierungsstelle ist, die dieses Zertifikat ausgestellt hat, so muß der Name des CRL-Herausgebers im *cRLIssuer*-Teilfeld angegeben werden.

### V 3.2.2. OPENCDP (OPEN CRL DISTRIBUTION PROCESS)

Eine Alternative zu CDP bietet der Open CRL Distribution Process (OpenCDP), der eine Segmentierung von CRLs nach verschiedenen Kriterien ermöglicht. Die Aufteilung einer CRL in mehrere Teil-CRLs erfordert zwei Funktionen:

- eine Funktion zur Lokalisierung der entsprechenden CRL
- eine Funktion, die die Verbindung zwischen Zertifikat und Teil-CRL herstellt

Zur Lokalisierung der entsprechenden CRL wurde in OpenCDP das X.500 Attribut *Revocation Information Attribute* eingeführt. Es beinhaltet keine sicherheits-kritischen Informationen und muß deshalb nicht im Zertifikat integriert und von der Zertifizierungsstelle signiert werden. Darüber hinaus bietet das *Revocation Information Attribute* die Möglichkeit, weitere Informationen zum Status eines Zertifikats abzulegen, wie beispielsweise die Adresse eine OCSP Dienstes.

Eine weitere Möglichkeit, CRLs zu lokalisieren, bietet das *CRL List Attribute*. Dieses X.500 Attribut einer Zertifizierungsstelle beinhaltet eine Liste aller (Teil)-CRLs.

Die Verbindung zwischen Zertifikat und CRL wird im Gegensatz zu CDP nicht statisch, sondern dynamisch hergestellt, indem eine Zertifizierungsstelle ein Segmentierkriterium festlegt und als X.509v2 CRL-Erweiterung *cRLScope* in die CRLs einträgt. Dieses Kriterium kann sie aber auch jederzeit wieder ändern ohne daß das Einfluß auf bereits ausgestellte Zertifikate oder CRLs hat.

Darüber hinaus definiert OpenCDP den *Revocation Issuer* als eine weitere X.509v3 Zertifikatserweiterung, mit der eine Zertifizierungsstelle im Fall von *indirectCRLs* anzeigen kann, daß sie das Ausstellen und Pflegen von Sperrlisten oder die Aufgabe des Auskunftdienstes (OCSP) an einen oder mehrere Dritte delegiert hat.

#### V 3.2.2.1. CRL-Erweiterung *cRLScope*

##### Zweck

Die optionale X.509v2 CRL-Erweiterung *cRLScope* ermöglicht eine Segmentierung von CRLs, da hiermit jede CRL das Segmentierungs-Kriterium beinhaltet. Mögliche Segmentierkriterien sind beispielsweise die Seriennummern oder der Gültigkeitszeitraum, mit denen eine CRL in mehrere Teil-CRLs zerlegt werden kann.

## ASN.1-Definitionen

```
Extension ::= SEQUENCE {
    extnId          OBJECT IDENTIFIER,
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING }

cRLScope     EXTENSION ::= {
    SYNTAX          CRLScopeSyntax
    IDENTIFIED BY   { <oid tbd> } }

CRLScopeSyntax ::= SEQUENCE {
    serialNumberRange [0] NumberRange OPTIONAL,
    subjectKeyIdRange [1] NumberRange OPTIONAL,
    nameSubtrees      [2] GeneralNames OPTIONAL,
    notBeforeRange    [3] NotBeforeRange OPTIONAL,
    onlyContainsUserCerts [4] BOOLEAN DEFAULT FALSE,
    onlyContainsCACerts [5] BOOLEAN DEFAULT FALSE,
    onlySomeReasons   [6] ReasonFlags OPTIONAL,
    indirectCRL       [7] BOOLEAN DEFAULT FALSE }

NumberRange ::= SEQUENCE {
    startingNumber    INTEGER,
    endingNumber      INTEGER,
    modulus           INTEGER OPTIONAL }

NotBeforeRange ::= SEQUENCE {
    startingNotBeforeTime GeneralizedTime,
    endingNotBeforeTime  GeneralizedTime }
```

### Allgemeine Konformitätsanforderungen:

Wenn das Feld *serialNumberRange* oder *subjectKeyIdRange* verwendet wird und ein Modulus angegeben ist, so muß die Seriennummer bzw. der Identifier zuerst modulo dieses Wertes *modulus* genommen werden, bevor überprüft wird, ob die Zahl im Intervall zwischen *startingNumber* (inklusive) und *endingNumber* liegt.

Für das Feld *nameSubtrees* gelten die gleichen Konventionen wie für die X.509v3 Zertifikatserweiterung *NameConstraints* in Z 2.3.9.13.

Wird das Feld *notBeforeRange* als Segmentierkriterium verwendet, so gehören alle Zertifikate zu dieser CRL, deren Gültigkeitsbeginn ab *startingNotBeforeTime* (inklusive) und deren Gültigkeitsende vor *endingNotBeforeTime* liegt.

Für die Felder *onlyContainsUserCerts*, *onlyContainsCACerts*, *onlySomeReasons* und *indirectCRL* gelten die gleichen Konventionen wie bei der X.509v2 CRL Erweiterung *issuingDistributionPoint* (siehe Abschnitt 3.1.3.7.4).

Die CRL Erweiterung *cRLScope* steht im Widerspruch zur X.509v2 CRL Erweiterung *issuingDistributionPoint* und darf nicht gleichzeitig mit ihr verwendet werden.

### ISIS-Konformitätsanforderungen:

Die Benutzung der *cRLScope*-CRL-Erweiterung bei der Erstellung von Sperrlisten ist optional und als *critical* zu markieren. **ISIS**-konforme Systeme und Anwendungen müssen

diese Erweiterung erkennen und verarbeiten können, sofern sie den OpenCDP-Mechanismus benutzen.

### V 3.2.2.2. X.500-Attribut revocation information attribute

Um die Inflexibilität von CDP zu vermeiden, ist bei OpenCDP der Verweis auf die entsprechende CRL nicht im Zertifikat integriert, sondern OpenCDP definiert ein *Revocation Information Attribute*, das den Verweis auf die CRL und weitere Informationen zum Status eines Zertifikats enthält. Dieses Attribut *revocInfo* kann entweder in einem X.500 Verzeichnis abgelegt werden oder zusätzlich zu dem Zertifikat mitgeschickt werden, z.B. bietet [PKCS7 93] die Möglichkeit den Daten noch weitere Attribute hinzuzufügen, die nicht signiert werden.

#### ASN.1-Definitionen

```

revocInfo          ATTRIBUTE ::= {
    WITH SYNTAX      RevocInfo
    ID                <oid tbd> }

RevocInfo         ::= SEQUENCE {
    certIssuer        GeneralNames,
    certSerialNumber INTEGER,
    infoLocations     [0] SEQUENCE SIZE (1..MAX) OF InfoLocation
                       OPTIONAL,
    extensions        [1] SEQUENCE OF INSTANCE OF
                       TYPE-IDENTIFIER OPTIONAL }

InfoLocation     ::= SEQUENCE {
    locator           GeneralNames,
    infoType          InfoType DEFAULT crl,
    reasons           ReasonFlags OPTIONAL }

InfoType         ::= ENUMERATED {
    crl                (0),
    oCSPServer         (1) }

ReasonFlags     ::= BIT STRING {
    unused              (0),
    keyCompromise       (1),
    cACompromise        (2),
    affiliationChanged  (3),
    superseded           (4),
    cessationOfOperation (5),
    certificateHold      (6) }

```

#### ISIS-Konformitätsanforderungen

Die Bedeutung der einzelnen Sperrgründe ist in der Tabelle 15 angegeben. Zertifizierungsstellen, die zu OpenCDP konform sind, dürfen das *revocation information* Attribut nicht in Zertifikate einbauen. Die Teilkomponente *locator* der *InfoLocation*-Struktur enthält für CRLs entweder den zugehörigen Verzeichnisdienstnamen oder die zugehörige URL und für den OCSP-Dienst den zugehörigen DNS Namen. Zur Zeit werden keine Einschränkungen hinsichtlich der einzelnen Teilkomponenten von *revocInfo* gemacht.

### V 3.2.2.3. X.500-Attribut CRL list attribute

Außerdem definiert OpenCDP mit *CRL List Attribute* ein Attribut für X.500 konforme Directory Services, welches in der Objektklasse *certificationAuthority* verwendet werden kann. Es enthält eine signierte Liste von CRLs, wobei zu jeder CRL der Zeitpunkt des letzten Updates angegeben ist. Dieses Attribut kann ebenfalls im X.500-Verzeichniseintrag der Zertifizierungsstelle gehalten werden, um eine Liste aller ausgestellten Teil-CRLs zur Verfügung zu stellen. Der Vorteil dieses Attributs *cRLList* ist, daß nur diese Liste häufig neu herausgegeben werden muß, nicht aber alle Teil-CRLs, sofern sie sich nicht geändert haben. Der Anwender muß nur diese Liste überprüfen, ob seine lokal gespeicherten CRLs noch aktuell sind, anstatt alle CRLs laden zu müssen. Darüberhinaus kann der Anwender über diesen Mechanismus erkennen, wenn vor dem Ablaufdatum einer CRL eine neue CRL herausgegeben wurde.

#### ASN.1-Definitionen

```
CRLList                ATTRIBUTE ::= {
    WITH SYNTAX          CRLList
    ID                   <oid tbd> }

CRLList                ::= SIGNED { SEQUENCE {
    signature            AlgorithmIdentifier,
    issuer               GeneralNames,
    thisUpdate           GeneralizedTime,
    nextUpdate           GeneralizedTime OPTIONAL,
    cRLLocators         CRLLocators } }

CRLLocators           ::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
    locator              GeneralName,
    cRLScope             CRLScopeSyntax,
    lastUpdate           GeneralizedTime OPTIONAL }
```

#### ISIS-Konformitätsanforderungen

Die Teilkomponente *locator* der *CRLLocators*-Struktur enthält für CRLs entweder den zugehörigen Verzeichnisdienstnamen oder die zugehörige URL und für den OCSP-Dienst den zugehörigen DNS Namen. Zur Zeit werden keine Einschränkungen hinsichtlich der einzelnen Teilkomponenten von *CRLList* gemacht.

### V 3.2.2.4. Erweiterung revocation issuer

Wenn *indirectCRLs* verwendet werden, kann eine Zertifizierungsstelle über die X.509v3-Zertifikatserweiterung *revocation issuer* in von ihr ausgestellten Zertifikat kenntlich machen, daß sie das Ausstellen und Pflegen von Sperrlisten oder die Aufgabe des Auskunftdienstes (OCSP) an einen oder mehrere Dritte übertragen hat. Diese Zertifikatserweiterung ersetzt das Feld *cRLIssuer* der *cRLDistributionPoint*-Erweiterung, die bei Verwendung von OpenCDP nicht verwendet werden darf.

### ASN.1-Definitionen

```

RevocationIssuer EXTENSION ::= {
  SYNTAX          GeneralNames
  IDENTIFIED BY   { <oid tbd> } }

```

### Allgemeine Konformitätsanforderungen

OpenCDP konforme Zertifizierungsstellen dürfen keine Zertifikate ausstellen, die die Sperrlisteninformationen *cRLDistributionPoint* beeinhalteten.

### ISIS-Konformitätsanforderungen

Zertifizierungsstellen können OpenCDP unterstützen. Hierfür müssen sie zumindest das *Revocation Information Attribute* erzeugen und zur Verfügung stellen. Darüber hinaus kann eine Segmentierung der Sperrliste vorgenommen werden. Für diesen Fall müssen Zertifizierungsstellen die X.509v2 CRL-Erweiterung *cRLScope* in ihren Sperrlisten verwenden.

Eine Zertifizierungstelle darf entweder nur CDP oder nur OpenCDP unterstützen, auf gar keinen Fall dürfen beide Mechanismen parallel verwendet werden.

## V 3.2.3. OCSP (ONLINE CERTIFICATE STATUS PROTOCOL) AUF DER BASIS VON CRLS

Das Online Certificate Status Protocol (OCSP) ermöglicht aktuellere Sperrinformationen als dies mit periodisch herausgegebenen Sperrlisten möglich ist. Dieses Protokoll basiert von der Grundidee her nicht auf CRLs, kann aber auch CRLs als Informationsquelle verwenden, d.h. die OCSP-Antworten werden auf der Basis von CRLs gegeben. Sind die Informationen nicht auf Auskünfte aus Sperrlisten beschränkt, sondern basieren auf einer Positivliste aller jemals ausgestellten Zertifikate, so erfüllt OCSP die Anforderungen eines Verzeichnisdienstes gemäß des Signaturgesetzes. Dieser Fall wurde in Kapitel V 2 ausführlich behandelt.

Das Protokoll OCSP kann auch auf der Basis von CRLs arbeiten. Damit geht zwar der Vorteil der Aktualität der Auskünfte verloren, aber es bietet für Anwenderinfrastrukturen eine zusätzliche Möglichkeit zur Verwaltung und Bereitstellung von Sperrlisten. Es können mittels OCSP allerdings keine CRLs an den Anwender zurückgeliefert werden.

Die Verifikation einer Signatur verläuft on-line, d.h. es wird nicht gegen eine lokale CRL geprüft, sondern es wird ein Verzeichnisdienst befragt, der auf der Basis einer ihm vorliegenden CRL Auskünfte erteilt. Mögliche Antworten sind in diesem Fall entweder *notRevoked*, *revoked* oder *unknown*.

Dieser OCSP Dienst muß nicht von der Zertifizierungsstelle betrieben werden, sondern kann von anderen Organisationen übernommen werden, die dafür verantwortlich sind, sich stets die aktuelle Sperrliste von der Zertifizierungsstelle zu laden.

Beispielsweise könnte ein Kaufhaus diese Variante des Sperrlisten-Managements nutzen, die eine Prüfung über den signaturgesetz-konformen Verzeichnisdienst nur ab einer bestimm-

ten Höhe des Rechnungsbetrags durchführen will. In den übrigen Fällen genügt es ihnen, ihren internen auf CRLs basierenden OCSP Dienst zu befragen, der die Sperrlisteninformationen in regelmäßigen Abständen oder unregelmäßig beim Auftreten von Sperrereignissen von der Zertifizierungsstelle erhält.

Die Verwendung eines solchen internen OCSP-Dienstes bei der Verifikation von Signaturen ist nicht gesetzeskonform, die erzielten Verifikationsergebnisse können von dem Ergebnis abweichen, welches eine gesetzeskonforme Verifikation liefert. Die Einschätzung des Risikos sowie die Abwägung dieses Risikos gegen die niedrigeren Kommunikationskosten und Wartezeiten obliegt dem Anwender.

#### V 3.2.4. ABFRAGE VON SPERRLISTEN

Für den Abruf von Sperrlisten werden keine gesonderten Protokolle definiert. Für diesen Zweck sei auf die Dokumente der IETF PKIX Working Group verwiesen [PKIX OP-LDAP 98, PKIX OP-FTPHTTP 98], die diese Funktionen beschreiben. In diesen Dokumenten werden die Protokolle definiert, über die Sperrlisten von Servern abgerufen werden können. Diese Dokumente adressieren das Problem des Abrufens von Zertifikaten und Sperrlisten.

Als Protokolle werden LDAPv2 sowie FTP und HTTP definiert. Nicht definiert wird, wie die Anwenderinfrastruktur die Adresse des für eine Zertifizierungsstelle zuständigen Servers erhält. Üblicherweise wird die Anwenderinfrastruktur den technischen Namen der Zertifizierungsstelle verwenden, um im globalen X.500 Directory den entsprechenden Eintrag abzurufen. Hier können dann Attribute abgelegt werden, die eine oder mehrere URLs für den Abruf beschreiben (siehe Abschnitt 2.3.2). Möglich ist auch, daß im alternativen Namen der Zertifizierungsstelle eine Internet-Adresse direkt angegeben wird. In diesem Fall ist kein globales X.500 notwendig.

##### Beispiele für URLs aus [PKIX OP-FTPHTTP]:

```
ftp://ftp.netcom.com/sp/spyrus/housley.cer
ftp://ftp.your.org/pki/id48.cer
ftp://ftp.your.org/pki/id48.no42.crl
http://www.netcom.com/sp/spyrus/housley.cer
http://www.your.org/pki/id48.cer
http://www.your.org/pki/id48.no42.crl
```

In [PKIX OP-LDAP 98] werden die minimalen Protokollelemente beschrieben, die ein LDAP-Server zum Abruf von Daten im Rahmen der PKIX unterstützen muß. Daneben wird ebenfalls definiert, welche Protokollelemente ein LDAP-Server unterstützen muß, wenn die in ihm gespeicherten Daten verändert werden sollen. Die in diesen Dokumenten spezifizierten Protokolle decken diesen Themenbereich hinreichend ab.

## V 4. ANHANG <sup>2</sup> OBJEKTBEZEICHNER

Die folgende Tabelle enthält eine Übersicht über alle Objektbezeichner, die im Abschnitt V benutzt wurden.

**Tabelle 26: Objektbezeichner**

OBJEKTBEZEICHNERNUMMERN										OBJEKTBEZEICHNERNAMEN	REFERENZ
1										iso	
	2									member-body	
		840								data country code, USA	
			10040							X9-57	
				2						holdInstruction	
					1					none	3.2.3.6.1
					2					callissuer	3.2.3.6.1
					3					reject	3.2.3.6.1
	3									identified-organization	
		6								dod	
			1							internet	
				5						security	
					5					mechanisms	
						7				pkix	
							48			id-ad, access description	
								1		ocsp	
									1	ocsp-basic	3.2.2
									2	caIssuers	
		36								teletrust	
			8							id-isis	
				3						id-isis-at	
					9					id-isis-at-retrieveIfAllowed	2.2.1
					10					id-isis-at-requestedCertificate	2.2.2
					12					id-isis-at-CertInDirSince	2.2.2
					13					id-isis-at-certHash	2.2.1

## Fortsetzung von Tabelle 26

OBJEKTBEZEICHNERNUMMERN										OBJEKTBEZEICHNERNAMEN	REFERENZ
2										<b>joint-iso-ccitt</b>	
	5									ds	
		4								attributeType	
			3							commonName	3.2.3.3
			4							surName	3.2.3.3
			5							serialNumber	3.2.3.3
			6							countryName	3.2.3.3
			7							localityName	3.2.3.3
			8							stateOrProvinceName	3.2.3.3
			10							organizationName	3.2.3.3
			11							organizationalUnit	3.2.3.3
			12							title	3.2.3.3
			15							businessCategory	3.2.3.3
			17							postalCode	2.2.3.3
			47							givenName	3.2.3.3
		29								id-ce, certificate extensions	
			18							issuerAltName	3.2.3.7.2
			20							cRLNumber	3.2.3.7.3
			21							cRLReason	3.2.3.6.1
			23							holdInstructionCode	3.2.3.6.1
			24							invalidityDate	3.2.3.6.1
			27							deltaCRLIndicator	3.2.3.7.5
			28							issuingDistributionPoint	3.2.3.7.4
			29							certificateIssuer	3.2.3.6.1
			31							cRLDistributionPoints	3.3.1
			35							authorityKeyIdentifier	3.2.3.7.1

## V 5. ANHANG <sup>2 2</sup> ASN.1 DEFINITIONEN

Dieser Abschnitt enthält eine Zusammenfassung aller ASN.1-Definitionen in alphabetischer Reihenfolge, die im Abschnitt V benutzt werden.

<b>AlgorithmIdentifier</b>	::=	SEQUENCE { Algorithm parameters	OBJECT IDENTIFIER, ANY DEFINED BY algorithm OPTIONAL }
<b>AttributeType</b>	::=	OBJECT IDENTIFIER	
<b>AttributeTypeAndValue</b>	::=	SEQUENCE { type value	AttributeType, AttributeValue }
<b>AttributeValue</b>	::=	ANY DEFINED BY AttributeType	
<b>authorityKeyIdentifier</b>	EXTENSION ::= {	SYNTAX IDENTIFIED BY	AuthorityKeyIdentifier id-ce-authorityKeyIdentifier }
<b>AuthorityKeyIdentifier</b>	::=	SEQUENCE { keyIdentifier authorityCertIssuer authorityCertSerialNumber	[0] KeyIdentifier OPTIONAL, [1] GeneralNames OPTIONAL, [2] CertificateSerialNumber OPTIONAL }
<b>BaseCRLNumber</b>	::=	CRLNumber	
<b>BasicOCSPResponse</b>	::=	SEQUENCE { tbsResponseData signatureAlgorithm signature certs	ResponseData, AlgorithmIdentifier, BIT STRING, [1] EXPLICIT SEQUENCE OF Certificate OPTIONAL }
<b>certHash</b>	EXTENSION ::= {	SYNTAX IDENTIFIED BY	CertHashSyntax id-isis-at-certHash }
<b>CertHashSyntax</b>	::=	SEQUENCE { hashAlgorithm certificateHash	AlgorithmIdentifier, OCTET STRING }
<b>CertID</b>	::=	SEQUENCE { hashAlgorithm issuerNameHash issuerKeyHash serialNumber	AlgorithmIdentifier, OCTET STRING, OCTET STRING, CertificateSerialNumber }
<b>certificateIssuer</b>	EXTENSION ::= {	SYNTAX IDENTIFIED BY	CertificateIssuer id-ce-certificateIssuer }
<b>CertificateIssuer</b>	::=	GeneralNames	
<b>CertificateList</b>	::=	SEQUENCE { tbsCertList signatureAlgorithm signature	TBSCertList, AlgorithmIdentifier, BIT STRING }

<b>CertificateSerialNumber</b>		INTEGER
<b>CertInDirSince</b> EXTENSION		::= {
SYNTAX		CertInDirSinceSyntax
IDENTIFIED BY		id-ce-authorityKeyIdentifier }
<b>CertInDirSinceSyntax</b>		GeneralizedTime
<b>CertStatus</b>	::=	CHOICE {
good		[0] IMPLICIT NULL,
revoked		[1] IMPLICIT RevokedInfo,
unknown		[2] IMPLICIT UnknownInfo }
<b>CRLDistPointsSyntax</b>	::=	SEQUENCE SIZE(1..MAX) OF DistributionPoint
<b>cRLDistributionPoints</b> EXTENSION	::=	{
SYNTAX		CRLDistPointsSyntax
IDENTIFIED BY		id-ce- cRLDistributionPoints }
<b>CRLList</b> ATTRIBUTE	::=	{
WITH SYNTAX		CRLList
ID		<oid tbd> }
<b>CRLList</b>	::=	SIGNED { SEQUENCE {
signature		AlgorithmIdentifier,
issuer		GeneralNames,
thisUpdate		GeneralizedTime,
nextUpdate		GeneralizedTime OPTIONAL,
cRLLocators		CRLLocators } }
<b>CRLLocators</b>	::=	SEQUENCE SIZE (1..MAX) OF SEQUENCE {
locator		GeneralName,
cRLScope		CRLScopeSyntax,
lastUpdate		GeneralizedTime OPTIONAL )
<b>cRLNumber</b> EXTENSION	::=	{
SYNTAX		CRLNumber
IDENTIFIED BY		id-ce-cRLNumber }
<b>CRLNumber</b>	::=	INTEGER (0..MAX)
<b>CRLReason</b> EXTENSION	::=	{
SYNTAX		CRLReason
IDENTIFIED BY		id-ce-cRLReason }
<b>CRLReason</b>	::=	ENUMERATED {
unspecified		(0),
keyCompromise		(1),
cACompromise		(2),
affiliationChanged		(3),
superseded		(4),
cessationOfOperation		(5),
certificateHold		(6),
removeFromCRL		(7) }
<b>CRLScope</b> EXTENSION	::=	{
SYNTAX		CRLScopeSyntax
IDENTIFIED BY		{ <oid tbd > } }

---

<b>CRLScopeSyntax</b>	::=	SEQUENCE {
serialNumberRange		[0] NumberRange OPTIONAL,
subjectKeyIdRange		[1] NumberRange OPTIONAL,
nameSubtrees		[2] GeneralNames OPTIONAL,
notBeforeRange		[3] NotBeforeRange OPTIONAL,
onlyContainsUserCerts		[4] BOOLEAN DEFAULT FALSE,
onlyContainsCACerts		[5] BOOLEAN DEFAULT FALSE,
onlySomeReasons		[6] ReasonFlags OPTIONAL,
indirectCRL		[7] BOOLEAN DEFAULT FALSE }

---

<b>deltaCRLIndicator</b>	EXTENSION ::=	{
SYNTAX		BaseCRLNumber
IDENTIFIED BY		id-ce-deltaCRLIndicator }

---

<b>DigestInfo</b>	::=	SEQUENCE {
digestAlgorithm		AlgorithmIdentifier,
digest		OCTET STRING }

---

<b>DirectoryString</b>	::=	CHOICE {
printableString		PrintableString (SIZE (1..maxSize))
teletexString		TeletexString (SIZE (1..maxSize))
bmpString		BMPString (SIZE (1..maxSize))
universalString		UniversalString (SIZE (1..maxSize)) }

---

<b>DistributionPoint</b>	::=	SEQUENCE {
distributionPoint		[0] DistributionPointName OPTIONAL,
reasons		[1] ReasonFlags OPTIONAL,
cRLIssuer		[2] GeneralNames OPTIONAL }

---

<b>DistributionPointName</b>	::=	CHOICE {
fullName		[0] GeneralNames,
nameRelativeToCRLIssuer		[1] RelativeDistinguishedName }

---

<b>EDIPartyName</b>	::=	SEQUENCE {
nameAssigner		[0] DirectoryString OPTIONAL,
partyName		[1] DirectoryString }

---

<b>EXTENSION</b>	::=	CLASS {
&id		OBJECT IDENTIFIER UNIQUE,
&ExtType }		
WITH SYNTAX {		
SYNTAX		&ExtnType
IDENTIFIED BY		&id }

---

<b>Extensions</b>	::=	SEQUENCE (1..MAX) OF Extension
-------------------	-----	--------------------------------

---

<b>GeneralName</b>	::=	CHOICE {
otherName		[0] OTHER-NAME,
rfc822Name		[1] IA5String,
dNSName		[2] IA5String,
x400Address		[3] ORAddress,
directoryName		[4] Name,
ediPartyName		[5] EDIPartyName,
uniformResourceIdentifier		[6] IA5String,
iPAddress		[7] OCTET STRING,
registeredID		[8] OBJECT IDENTIFIER }

---

<b>GeneralNames</b>	::=	SEQUENCE SIZE (1..MAX) OF GeneralName
---------------------	-----	---------------------------------------

---

---

<b>HoldInstructionCode</b> EXTENSION	::=	{
SYNTAX		HoldInstructionCode
IDENTIFIED BY		id-ce-holdInstructionCode }

---

<b>HoldInstructionCode</b>	::=	CHOICE {
id-holdinstruction-none		OBJECT IDENTIFIER,
id-holdinstruction-callissuer		OBJECT IDENTIFIER,
id-holdinstruction-reject		OBJECT IDENTIFIER }

---

<b>InfoLocation</b>	::=	SEQUENCE {
locator		GeneralNames,
infoType		InfoType DEFAULT crl,
reasons		ReasonFlags OPTIONAL }

---

<b>InfoType</b>	::=	ENUMERATED {
crl		(0),
oCSPServer		(1) }

---

<b>invalidityDate</b> EXTENSION	::=	{
SYNTAX		GeneralizedTime
IDENTIFIED BY		id-ce-invalidityDate }

---

<b>issuerAltName</b> EXTENSION	::=	{
SYNTAX		IssuerAltName
IDENTIFIED BY		id-ce-issuerAltName }

---

<b>IssuerAltName</b>	::=	GeneralNames
----------------------	-----	--------------

---

<b>issuingDistributionPoint</b> EXTENSION	::=	{
SYNTAX		IssuingDistributionPoint
IDENTIFIED BY		id-ce-issuingDistributionPoint }

---

<b>IssuingDistributionPoint</b>	::=	SEQUENCE {
distributionPoint	[0]	DistributionPointName OPTIONAL,
onlyContainsUserCerts	[1]	BOOLEAN DEFAULT FALSE,
onlyContainsCACerts	[2]	BOOLEAN DEFAULT FALSE,
onlySomeReasons	[3]	ReasonFlags OPTIONAL,
indirectCRL	[4]	BOOLEAN DEFAULT FALSE }

---

<b>KeyHash</b>	::=	OCTET STRING
----------------	-----	--------------

---

<b>KeyIdentifier</b>	::=	OCTET STRING
----------------------	-----	--------------

---

<b>Name</b>	::=	CHOICE { RDNSequence }
-------------	-----	------------------------

---

<b>NotBeforeRange</b>	::=	SEQUENCE {
startingNotBeforeTime		GeneralizedTime,
endingNotBeforeTime		GeneralizedTime }

---

<b>NumberRange</b>	::=	SEQUENCE {
startingNumber		INTEGER,
endingNumber		INTEGER,
modulus		INTEGER OPTIONAL }

---

---

<b>OCSRequest</b>	::=	SEQUENCE {
tbsRequest		TBSRequest,
optionalSignature		[0] EXPLICIT Signature OPTIONAL }

---

<b>OCSResponse</b>	::=	SEQUENCE {
responseStatus		OCSResponseStatus,
responseBytes		[0] EXPLICIT ResponseBytes OPTIONAL }

---

<b>OCSResponseStatus</b>	::=	ENUMERATED {
successful		(0),
malformedRequest		(1),
internalError		(2),
tryLater		(3),
certRequired		(4), -- not used
sigRequired		(5),
unauthorized		(6) }

---

<b>OTHER-NAME</b>	::=	SEQUENCE {
type-id		OBJECT IDENTIFIER,
value		[0] EXPLICIT ANY DEFINED BY type-id }

---

<b>RDNSSequence</b>	::=	SEQUENCE OF RelativeDistinguishedName
---------------------	-----	---------------------------------------

---

<b>ReasonFlags</b>	::=	Bit String {
unused		(0),
keyCompromise		(1),
cACompromise		(2),
affiliationChanged		(3),
superseded		(4),
cessationOfOperation		(5),
certificateHold		(6) }

---

<b>RelativeDistinguishedName</b>	::=	SET OF AttributeTypeAndValue
----------------------------------	-----	------------------------------

---

<b>Request</b>	::=	SEQUENCE {
reqCert		CertID,
singleRequestExtensions		[0] EXPLICIT Extensions OPTIONAL }

---

<b>requestedCertificate</b> EXTENSION	::=	{
SYNTAX		RequestedCertificateSyntax
IDENTIFIED BY		id-isis-at-requestedCertificate }

---

<b>RequestedCertificateSyntax</b>	::=	Certificate
-----------------------------------	-----	-------------

---

<b>ResponderID</b>	::=	CHOICE {
byName		[0] Name,
byKey		[1] KeyHash }

---

---

<b>ResponseBytes</b>	::=	SEQUENCE { responseType response
<b>ResponseData</b>	::=	SEQUENCE { version responderID producedAt responses responseExtensions
<b>retrieveIfAllowed</b> EXTENSION	::=	{ SYNTAX IDENTIFIED BY
<b>RevocationIssuer</b> EXTENSION	::=	{ SYNTAX IDENTIFIED BY
<b>revocInfo</b> ATTRIBUT	::=	{ WITH SYNTAX ID
<b>RevocInfo</b>	::=	SEQUENCE { certIssuer certSerialNumber infoLocations extensions
<b>revokedCertificates</b>		SEQUENCE OF SEQUENCE { userCertificate revocationDate crlEntryExtensions
<b>RevokedInfo</b>	::=	SEQUENCE { revocationTime revocationReason
<b>Signature</b>	::=	SEQUENCE { signatureAlgorith signature certs
<b>SingleResponse</b>	::=	SEQUENCE { certID certStatus thisUpdate nextUpdate singleExtensions

---

---

<b>TBSCertList</b>	::=	SEQUENCE { version signature issuer thisUpdate nextUpdate revokedCertificates userCertificate revocationDate crlEntryExtensions crlExtensions	Version OPTIONAL, AlgorithmIdentifier, Name, Time, Time OPTIONAL, SEQUENCE OF SEQUENCE { CertificateSerialNumber, Time, Extensions OPTIONAL } OPTIONAL, [0] EXPLICIT Extensions OPTIONAL }
<b>TBSRequest</b>	::=	SEQUENCE { version requestorName requestList requestExtensions	[0] EXPLICIT Version OPTIONAL, [1] EXPLICIT GeneralName OPTIONAL, SEQUENCE OF Request, [2] EXPLICIT Extensions OPTIONAL }
<b>Time</b>	::=	CHOICE { utcTime generalizedTime	UTCTime, GeneralizedTime }
<b>UnknownInfo</b>	::=	NULL	
<b>CertInDirSince</b> EXTENSION	::=	{ SYNTAX IDENTIFIED BY	CertInDirSinceSyntax id-isis-at-CertInDirSince }
<b>VerifySyntax</b>	::=	GeneralizedTime	
<b>Version</b>	::=	INTEGER { v1(0), v2(1) }	

---

Objektbezeichner


---

<b>id-ad</b> OBJECT IDENTIFIER	::=	{ 1 3 6 1 5 5 7 48 }
<b>id-ce</b> OBJECT IDENTIFIER	::=	{ 2 5 29 }
<b>id-ce-authorityKeyIdentifier</b> OBJECT IDENTIFIER	::=	{ 2 5 29 35 }
<b>id-ce-certificateIssuer</b> OBJECT IDENTIFIER	::=	{ 2 5 29 29 }
<b>id-ce-cRLDistributionPoints</b> OBJECT IDENTIFIER	::=	{ 2 5 29 31 }
<b>id-ce-cRLNumber</b> OBJECT IDENTIFIER	::=	{ 2 5 29 20 }
<b>id-ce-cRLReason</b> OBJECT IDENTIFIER	::=	{ 2 5 29 21 }
<b>id-ce-deltaCRLIndicator</b> OBJECT IDENTIFIER	::=	{ 2 5 29 27 }
<b>id-ce-holdInstructionCode</b> OBJECT IDENTIFIER	::=	{ 2 5 29 23 }
<b>id-ce-invalidityDate</b> OBJECT IDENTIFIER	::=	{ 2 5 29 24 }

---

---

<b>id-ce-issuerAltName</b>	OBJECT IDENTIFIER ::=	{ 2 5 29 18 }
<b>id-ce-issuingDistributionPoint</b>	OBJECT IDENTIFIER ::=	{ 2 5 29 28 }
<b>id-holdInstruction</b>	OBJECT IDENTIFIER ::=	{ 1 2 840 10042 2 }
<b>id-holdinstruction-callissuer</b>	OBJECT IDENTIFIER ::=	{ 1 2 840 10042 2 2 }
<b>id-holdinstruction-none</b>	OBJECT IDENTIFIER ::=	{ 1 2 840 10042 2 1 }
<b>id-holdinstruction-reject</b>	OBJECT IDENTIFIER ::=	{ 1 2 840 10042 2 3 }
<b>id-pkix-ocsp</b>	OBJECT IDENTIFIER ::=	{ 1 3 6 1 5 5 7 48 1 }
<b>id-pkix-ocsp-basic</b>	OBJECT IDENTIFIER ::=	{ 1 3 6 1 5 5 7 48 1 1 }
<b>id-isis</b>	OBJECT IDENTIFIER ::=	{ 1 3 36 8 }
<b>id-isis-at</b>	OBJECT IDENTIFIER ::=	{ 1 3 36 8 3 }
<b>id-isis-at-certHash</b>	OBJECT IDENTIFIER ::=	{ 1 3 36 8 3 13 }
<b>id-isis-at-requestedCertificate</b>	OBJECT IDENTIFIER ::=	{ 1 3 36 8 3 10 }
<b>id-isis-at-retrieveIfAllowed</b>	OBJECT IDENTIFIER ::=	{ 1 3 36 8 3 9 }
<b>id-isis-at-CertInDirSince</b>	OBJECT IDENTIFIER ::=	{ 1 3 36 8 3 12 }

---

# Abschnitt III: Allgemeines

## A 1. ABKÜRZUNGEN UND BEGRIFFE

### Bemerkung

Begriffe und Definitionen aus dem Bereich der Sicherheitstechnik werden international in englischer Sprache formuliert. Die folgende Tabelle enthält deutsche Übersetzungen zu den wichtigsten englischen Fachausdrücken und eine Erläuterung von häufig benutzten Abkürzungen. Formale Namen von technischen Objekten, die für die Verarbeitung von Systemen benötigt werden und für die es eine eigene formale Syntax gibt, werden in der Tabelle durch Kursivschrift hervorgehoben.

**Tabelle 53: Abkürzungen und Begriffe**

ABKÜRZUNG	ENGLISCH	DEUTSCH
*	wild card	Platzhaltersymbol für Teilstrings
		Konkatenierung von Daten
	<i>admission</i>	spezifische Erweiterung für Zulassungsinformation
	algorithm identifier <i>AlgorithmIdentifier</i>	Eindeutiger Bezeichner des kryptographischen Algorithmus, der von einer Zertifizierungsstelle zum Signieren eines Zertifikates benutzt wird
	alternative subject name <i>subjectAltName</i>	Zertifikatserweiterung, die einen oder mehrere alternative Namen für Zertifikatsinhaber enthält, durch die zusätzliche Identitäten an den Zertifikatsinhaber gebunden werden
ANS	american national standard	US-Normen
ANSI	american national standards institute	US-Normungsgremium
ASN.1	abstract syntax notation one	abstrakte Notation zur Beschreibung von Datentypen und Datenwerten
	<i>atAdmission</i>	spezifisches Attribut für Zulassungsinformationen
	<i>atMonetaryLimit</i>	spezifisches Attribut für monetäre Beschränkungen
	<i>atProcuratont</i>	spezifisches Attribut für Vertretungsmacht
	attributes	Feld eines Attributzertifikates, das die eigentlichen Nutzdaten eines Attributzertifikates enthält, die syntaktisch in der Form von X.500-Verzeichnisdienstattributen aufgebaut sind
	authority key identifier <i>authorityKeyIdentifier</i>	Feld einer Zertifikatserweiterung, das zur Identifizierung eines bestimmten öffentlichen Schlüssels einer Zertifizierungsstelle dient
	base certificate identifier <i>baseCertificateID</i>	Feld eines Attribut-Zertifikates, durch das – alternativ zum subject-Feld – indirekt der Name des Zertifikatsinhabers über den Namen der Zertifizierungsstelle, die das Zertifikat ausgestellt hat, und die zugehörige Seriennummer des Zertifikates angegeben werden kann

ABKÜRZUNG	ENGLISCH	DEUTSCH
	basic constraints <i>basicConstraints</i>	Zertifikatserweiterung, die anzeigt, ob ein Zertifikatsinhaber in der Rolle als Zertifizierungsstelle auftreten kann und ob eine Beschränkung der Zertifizierungspfadlänge vorliegt
BER	basic encoding rules	Variante von ASN.1-Kodierungsvorschriften, mehrdeutig
BSI		Bundesamt für Sicherheit in der Informationstechnik
C	country	Länderbezeichnung nach ISO 3166, Attribut in <i>distinguished name</i> -Typen
CA	certification authority	Zertifizierungsstelle
CCITT	comité consultatif international pour télégraphique et téléphonique (international consultative committee for telephone and telegraph)	
	certificate	Zertifikat, gemäß X.509 digital signierte Datenstruktur, welche die Bindung der Identität eines Zertifikatsinhabers zu einem öffentlichen Schlüssel herstellt
	public key certificate	
	certificate and CRL repository	Ablage für Zertifikate und Zertifikats-Sperrlisten
	certificate and CRL retrieval	Suche von Zertifikaten und Zertifikats-Sperrlisten
	certificate policies <i>certificatePolicies</i>	Zertifikatserweiterung, die zur Anzeige der Verfahrensweisen bei der Erstellung eines Zertifikates durch eine Zertifizierungsstelle und der Zwecke, die mit dem Zertifikat verbunden sind, dient
	certificate renewal	Erneuerung von Zertifikaten
	certificate revocation	Sperren von Zertifikaten
	certificate user	Zertifikatsbenutzer: eine Person oder ein System, das Zertifikate benutzt
	certification path	Zertifizierungsweg: Eine geordnete Folge von Zertifikaten, beginnend mit dem Zertifikat, dessen öffentlichen Schlüssel ein Benutzer kennt, bis hin zu einem Zertifikat, dessen öffentlicher Schlüssel von einem Benutzer zu validieren ist
	certification request	Zertifikatsanforderung
CDP	certificate distribution point	Verteilungspunkt für Zertifikate
CHA	certificate holder authorization	Rechte von Zertifikatsinhabern
CHR	card holder reference	Referenzierung des öffentlichen Schlüssels einer Zertifizierungsstelle
	client	Kunde, Kommunikationspartner, Teilnehmer, Anwendungsprogramm
CN	common name	Personenname, Attribut in <i>distinguished name</i> -Typen
	confidentiality key management certificate	Zertifikat für vertrauliche Schlüsselverwaltung
CPI	certificate profile identifier	Kennzeichnung des Aufbaus von Authentisierungszertifikaten
CPS	certification practise	spezielles <i>PolicyQualifiers</i> -Merkmal, das veröffentlichte Aussagen einer Zertifizierungsstelle über die Erfahrungen

ABKÜRZUNG	ENGLISCH	DEUTSCH
	statement	enthält, die sie bei der Erstellung von Zertifikaten gemacht hat
	<i>critical</i>	Zertifikatsfeld, das die Wichtigkeit einer Zertifikatserweiterung anzeigt
CRL	certificate revocation list	Liste zurückgezogener Zertifikate, Sperrliste
	CRL distribution points	Zertifikatserweiterung, die Informationen enthält, die zur Beschaffung von Sperrlisten dienen
	<i>cRLDistributionPoints</i>	
	cross-certification	gegenseitige Zertifizierung basierend auf einem Netzwerk-Vertrauensmodell
CV	card verifiable certificates	Zertifikate zur Authentisierung von Terminal und Chipkarte
DAP	directory access protocol	Protokoll für den Zugriff auf ein X.500-Verzeichnis
	data encipherment	Bit im Feld der Zertifikatserweiterung <i>keyUsage</i> , das die Schlüsselnutzungsart "Verschlüsselung von Nutzdaten" anzeigt
	<i>dataEncipherment</i>	
	<i>dateOfCertGen</i>	spezifische Erweiterung für das Erstellungsdatum eines Zertifikates
DC	domain component	Teilname eines Domänennamens, Attribut in <i>distinguished name</i> -Typen
DD	day	zweistellige Tageszahl
DE	data element	Datenelement, Chipkarten
	decipher only	Bit im Feld der Zertifikatserweiterung <i>keyUsage</i> , das die Schlüsselnutzungsart: "Schlüsselaustauschverfahren zur alleinigen Entschlüsselung von Daten" anzeigt
	<i>decipherOnly</i>	
delta-CRL	deltacertificate revocation list	Änderungen einer Sperrliste
DER	distinguished encoding rules	Variante von ASN.1-Kodierungsvorschriften
	digital signature	Bit im Feld der Zertifikatserweiterung <i>keyUsage</i> , das die Schlüsselnutzungsart "allgemeine Prüfung digitaler Signaturen zur Authentifizierung" anzeigt
	<i>digitalSignature</i>	
	digital signature public key certificate	Signaturzertifikat basierend auf öffentlichen Schlüsseln
DIN		Deutsches Institut für Normung e.V.
DIR	directory service	Verzeichnisdienst
	directory attributes of a subject	Zertifikatserweiterung, die zur Bereitstellung von Verzeichnis-Attributwerten für einen Zertifikatsinhaber dient
	<i>subjectDirectoryAttributes</i>	
DN	distinguished name	eindeutiger Name, der nach X.500 definiert wird
	<i>distinguishedName</i>	
DNS	domain name system	Methode zur Konvertierung zwischen Namen und Adressen im Internet
DO	data object	Datenobjekt, Chipkarten
DSA	digital signature algorithm	asymmetrischer Verschlüsselungsalgorithmus zum Generieren digitaler Signaturen
DSI	digital signature input	Signatur-Verschlüsselungsformate
DSS	digital signature standard	von Nist entwickelter Standard für digitale Signaturen

ABKÜRZUNG	ENGLISCH	DEUTSCH
		bestehend aus DSA und SHA-1
ECDSA	elliptic curve digital signature algorithm	Signaturalgorithmus basierend auf elliptischen Kurven
	end entity	Endanwender: Person als Anwender von Zertifikaten oder Endanwendersystem, das der Inhaber eines Zertifikates ist.
	encipher only <i>encipherOnly</i>	Bit im Feld der Zertifikatserweiterung <i>keyUsage</i> , das die Schlüsselnutzungsart "Schlüsselaustauschverfahren zur alleinigen Verschlüsselung von Daten" anzeigt
	explicit text <i>explicitText</i>	Teilfeld des <i>notice</i> -Merkmalanzeigefeldes, das den Namen einer Organisation sowie einen Text enthält, der eine spezielle Aussage dieser Organisation darstellt
	extended key usage <i>extKeyUsage</i>	Zertifikatserweiterung, die zur Definition von anwendungsabhängigen Nutzungsarten von zertifizierten Schlüsseln, zusätzlich oder alternativ zum <i>keyUsage</i> -Erweiterungsfeld, benutzt werden kann
	extension value <i>extnValue</i>	Zertifikatsfeld, das den Wert einer Zertifikatserweiterung enthält.
	extensions	optionales Zertifikatsfeld, das weitere Zertifikatserweiterungen enthält
	extensions identifier <i>extId</i>	Zertifikatsfeld, das den Objektbezeichner einer Zertifikatserweiterung enthält
FTP	file transfer protocol	Filetransferprotokoll
	generalized time format <i>GeneralizedTime</i>	ASN.1-Typ für allgemeine Datums- und Zeitformate
GET		HTTP-Zugriffsmethode
GMT	Greenwich Mean Time	Greenwich-Zeit
HH	hour	zweistellige Stundenzahl
HTML	hyper text markup language	auf SGML basierende Sprache zur Beschreibung von Hypertext-Dokumenten
HTTP	hypertext transfer protocol	Protokoll zum Laden von Dokumenten und/oder beschreibenden Kopfinformationen des WWW
ICC	integrated circuit card	Chipkarte
ICCSN	integrated circuit serial number	Seriennummer von Chipkarten
IEC	international electrotechnical commission	internationales Normungsgremium auf dem Gebiet der Elektrik und Elektronik
IETF	internet engineering task force	verantwortliches Gremium zur Entwicklung von Internet-Standards
IFD	interface device	Schnittstelle, Chipkarte
	inhibition of policy mapping <i>inhibitPolicyMapping</i>	optionales Feld der <i>policyConstraints</i> -Zertifikatserweiterung, das die Anzahl von weiteren Zertifikaten enthält, die Zertifizierungspfad folgen können, ehe eine Anerkennung fremder Zertifizierungsrichtlinien verboten ist

ABKÜRZUNG	ENGLISCH	DEUTSCH
IP	internet protocol	Übertragungsprotokoll der Netzwerkebene
IPSEC	internet protocol security	internet protocol, das Authentizität, Vertraulichkeit und Integrität gewährleistet
ISIS	Industrial Signature Interoperability Specification	die vorliegende gemeinsame Spezifikation der Arbeitsgemeinschaft Trust-Center für digitale Signaturen
ISO	international organization for standardization	internationales Standardisierungsgremium
	issuer alternative name <i>issuertAltName</i>	Zertifikatserweiterung, die einen oder mehrere alternative Namen für den Ersteller eines Zertifikates oder einer Zertifikats-Sperrliste enthält, durch die zusätzliche Identitäten aus dem Internetbereich an die Zertifizierungsstelle gebunden werden
	issuer	Zertifikatsfeld, das einen eindeutigen Namen der ausstellenden Zertifizierungsstelle enthält
	issuer domain policy <i>issuerDomainPolicy</i>	Teilfeld der <i>PolicyMappings</i> -Zertifikatserweiterung, die einer Inhabertzertifizierungsstelle Informationen über die Sicherheitsrichtlinien der Erstellerzertifizierungsstelle liefert,
	issuer unique identifier <i>issuerUniqueIdentifier</i>	optionales Zertifikatsfeld, das einen eindeutigen Bezeichner für die ausstellende Zertifikatserstelle enthält
ITU	international telecommunication union	Standardisierungsbehörde der UN
ITU-T	telecommunication standardization sector of ITU	Teilbereich der ITU (früher als CCITT bezeichnet), der für die Standardisierung im Telekommunikationsbereich zuständig ist.
	key agreement <i>keyAgreement</i>	Bit im Feld der Zertifikatserweiterung <i>keyUsage</i> , das die Schlüsselnutzungsart "Schlüsselaustauschverfahren" anzeigt
	key encipherment <i>keyEncipherment</i>	Bit im Feld der Zertifikatserweiterung <i>keyUsage</i> , das die Schlüsselnutzungsart "Schlüsseltransport, Schlüsselverwaltung" anzeigt
	key identifier of a subject <i>subjectKeyIdentifier</i>	Feld einer Zertifikatserweiterung, das zur Identifizierung eines bestimmten öffentlichen Schlüssels eines Zertifikatsinhabers dient
	key management certificate	Zertifikat für Schlüsselverwaltung
	key usage <i>keyUsage</i>	Feld einer Zertifikatserweiterung, das zur Anzeige der Verwendungszwecke, des in einem Zertifikat enthaltenen Schlüssels, dient
L	locality	Angabe eines geographischen Ortes, Attribut in <i>distinguished name</i> -Typen
LDAP	lightweight directory access protocol	Zugriffsprotokoll für Klienten auf Verzeichnisse, Alternative zu X.500
LSB	least significant bit	niederwertigstes Bit
MIME	multipurpose internet mail extensions	Internet-Standardformat für erweiterte elektronische Post
MISPC	minimum interoperability specification for PKI components	Spezifikation zur Entwicklung zusammenarbeitsfähiger Verfahren und Komponenten in einer öffentlichen Sicherheitsinfrastruktur

ABKÜRZUNG	ENGLISCH	DEUTSCH
	<i>monetaryLimit</i>	spezifische Erweiterung für monetäre Beschränkungen
MSB	most significant bit	höchstwertiges Bit
	name constraints <i>nameConstraints</i>	Zertifikatserweiterung, die den Namensraum anzeigt in dem Namen von Zertifikatsinhabern in aufeinanderfolgenden Zertifikaten eines Zertifizierungspfades liegen müssen
MM	month, minute	zweistellige Monats- oder Minutenzahl
NIST	national institute of standards and technology	nationales US-Institut (früher als NBS, national bureau of standards bezeichnet), das für Normen und deren technische Anwendungen zuständig ist
	non-repudiation <i>nonRepudiation</i>	Bit im Feld der Zertifikatserweiterung keyUsage, das die Schlüsselnutzungsart "Prüfung digitaler Signaturen zur Sicherung der Verbindlichkeit von Dokumenten und/oder Aktionen" anzeigt
	not valid after <i>notAfter</i>	Zertifikatsfeld, das das Ende der Gültigkeit eines Zertifikates festlegt
	not valid before <i>notBefore</i>	Zertifikatsfeld, das den Beginn der Gültigkeit eines Zertifikates festlegt
O	organization	Bezeichnung einer Organisation, Attribut in <i>distinguished name</i> -Typen
OCSP	online certificate status protocol	Anwendungsprotokoll zur Bestimmung des aktuellen Zustandes eines digitalen Zertifikates ohne die Benutzung von Sperrlisten
OID	object identifier	Objektbezeichner
OIW	Open systems environment implementors workshop	Objektbezeichnerzweig unter dem Objektbezeichnerzweig iso(1)-identified-organization(3)-OIW(14)
OpenCDP	open CRL distribution process	Mechanismen zur Verteilung von Sperrlisten
ORA	organizational registration authority	Organisationsregistrierungsstelle
OU	organizational unit	Bezeichnung einer untergeordneten Organisation, Attribut in <i>distinguished name</i> -Typen
	out-of band transaction or communication	Transaktionen oder Kommunikation, die außerhalb der zugrundeliegenden Infrastruktur abläuft
PCA	policy certification authority	Wurzelzertifizierungsstelle
PCT	private communication technology	Protokoll entwickelt von Microsoft und Visa International für sichere Kommunikation im Internet
PEM	privacy enhanced mail	Internetstandard für sichere elektronische Post
	period of private key usage <i>privateKeyUsagePeriod</i>	Zertifikatserweiterung, die zur Festlegung von unterschiedlichen Gültigkeitsdauern von Zertifikaten und privaten Schlüsseln, die für digitale Signaturzwecke benutzt werden, dient
PK	public key	öffentlicher Schlüssel
PKCS	public key crypto systems	Kryptosysteme basierend auf öffentlichen Schlüsseln
	public key cryptographic standard	RSA-Standards im Sicherheitsbereich

ABKÜRZUNG	ENGLISCH	DEUTSCH
PKI	public key infrastructure	Sicherheitsinfrastruktur basierend auf öffentlichen Schlüsseln
PKIX	internet public key infrastructure	Internetprotokolle für die Sicherheitsinfrastruktur im Internet basierend auf öffentlichen Schlüsseln
PN	<i>pseudoNym</i>	spezifisches Attribut für Pseudonyme in <i>distinguished</i> name-Typen
	policy constraints <i>policyConstraints</i>	Zertifikatserweiterung, die zur Spezifikation von Beschränkungen dient, die zusätzlich bei der Überprüfung von Zertifizierungspfaden zu beachten sind
	policy domain of a subject <i>subjectDomainPolicy</i>	Teilfeld der <i>PolicyMappings</i> -Zertifikatserweiterung, die einer Erstellerzertifizierungsstelle Informationen über die Sicherheitsrichtlinien der Inhaberzertifizierungsstelle liefert, die mit den eigenen Sicherheitsrichtlinien vergleichbar und somit von ihr akzeptierbar sind.
	policy identifier <i>policyIdentifier</i>	Feld des Typs <i>PolicyInformation</i> , das einen Objektbezeichner einer bestimmten angewandten Verfahrensweise enthält
	policy information <i>PolicyInformation</i>	Feld der Zertifikatserweiterung <i>certificatePolicies</i> , das Informationen über eine bestimmte angewandte Verfahrensweise enthält
	policy mappings <i>PolicyMappings</i>	Zertifikatserweiterung, die in Zertifikaten für Zertifizierungsstellen zur Anzeige der Äquivalenz von Sicherheitsrichtlinien von unterschiedlichen Zertifizierungsstellen dient
	policy qualifiers <i>policyQualifiers</i>	optionales Feld des Typs <i>PolicyInformation</i> , das weitere Merkmale einer bestimmten angewandten Verfahrensweise enthält
	<i>Procuration</i>	spezifische Erweiterung für Vertretungsmacht
	public key information of a subject <i>subjectPublicKeyInfo</i>	Zertifikatsfeld, das den öffentlichen Schlüssel des Zertifikatsinhabers enthält
POST		HTTP-Zugriffsmethode
RA	registration authority	Registrierungsstelle, an die eine Zertifizierungsstelle bestimmte Verwaltungsaufgaben delegieren kann
RCA	root CA	Wurzelzertifizierungsinstanz, zuständige Behörde
	reference to notice <i>noticeRef</i>	Teilfeld des <i>unotice</i> -Merkmalanzeigefeldes, das den Namen einer Organisation sowie einen numerischen Textverweis enthält, der auf eine spezielle Aussage dieser Organisation hinweist, die als Text vorbereitet worden ist
RDN	relative distinguished name	Komponente eines nach X.500 eindeutigen Namen
RegTP		Regulierungsbehörde für Telekommunikation und Post
	required explicit policies <i>requireExplicitPolicy</i>	optionales Feld der <i>policyConstraints</i> -Zertifikatserweiterung, das die Anzahl von weiteren Zertifikaten enthält, die im Zertifizierungspfad folgen können, ehe bestimmte, explizite Sicherheitsrichtlinien benötigt werden
	repository	Ablage, Verwahrungsort, Verzeichnis  System oder ein Verbund verteilter Systeme, die Zertifikate und Zertifikats-Sperrlisten speichern und deren Verteilung an

ABKÜRZUNG	ENGLISCH	DEUTSCH
		Endanwender unterstützen
RFC	request for comment	Internet-Report
RIPEMD-160		Von H.Dobbertin, A. Bosselaers und B. Preneel entwickelte Hashfunktion, die einen 160 Bit langen Hashwert ergibt. Der RIPEMD-160 ist eine Verbesserung des RIPEMD.
RSA	Rivest Shamir Adleman Algorithm	asymmetrischer Verschlüsselungsalgorithmus zum Verschlüsseln und Signieren von Daten  US-Firma, die Kryptoprotokolle und -software entwickelt und die das Patent an dem Algorithmus besitzt
S	surname	Nachnahme einer Person, Attribut in <i>distinguished name</i> -Typen
S/MIME	secure/multipurpose internet mail extensions	Protokoll, das zusätzlich zu MIME digitale Signaturen und Verschlüsselung enthält
SECSIG	security special interest group of OIW	spezielle Arbeitsgruppe innerhalb von OIW, sie sich mit Sicherheitsfragen beschäftigt
SER	serial number <i>serialNumber</i>	Zertifikatsfeld, das die Seriennummer des Zertifikates enthält, die innerhalb der Zertifizierungsstelle eindeutig sein muß
SET	secure electronic transaction	Protokoll entwickelt von Visa und MasterCard für sichere elektronische Transaktionen
SGML	standard generalized markup language	Standard zur allgemeinen Beschreibung von Dokumenten, dient als Grundlage für HTML
SHA-1	secure hash algorithm 1	Hashfunktion, Weiterentwicklung von SHA
SigI	Signatur-Interoperabilitätspezifikation	Vom BSI herausgegebene Interoperabilitätsspezifikation, bei Erstellung der ISIS-Spezifikation als Grundlage verwendet, siehe Literaturstellen [A1 99] etc.
SigG		Gesetz zur digitalen Signatur
SigV		Verordnung zur digitalen Signatur
SP	state or province	Bezeichnung eines Bundeslandes, Attribut in <i>distinguished name</i> -Typen
SS	second	zweistellige Sekundenzahl
SSL	secure socket layer	Sicherung der Kommunikation auf der Transport/Sessionebene
ST	street address	Straße als Teil einer postalischen Adresse, Attribut in <i>distinguished name</i> -Typen
	subject	Zertifikatsfeld, das einen eindeutigen Namen des Zertifikatsinhabers enthält
T	title	Angabe eines Titels, Attribut in <i>distinguished name</i> -Typen
	to be signed certificate information <i>tbCertificate</i>	Bestandteile eines Zertifikats, die von einer Zertifizierungsstelle zu signieren sind
TLS	transport layer security	Aufbau einer sicheren Transportverbindung über einen unsicheren Kanal
TSS	time stamp service	Zeitstempeldienst
	unique identifier of a subject	optionales Zertifikatsfeld, das einen eindeutigen Bezeichner für

ABKÜRZUNG	ENGLISCH	DEUTSCH
	<i>subjectUniqueIdentifier</i>	einen Zertifikatsinhaber enthält
URI	universal resource identifier	weltweit eindeutiger Bezeichner für Betriebsmittel
URL	universal resource location	weltweit eindeutiger Name für den Ort eines Betriebsmittels
	user notice	spezielles <i>PolicyQualifiers</i> -Merkmal, das zur Anzeige für
	<i>unotice</i>	Endanwender dient, daß ein Zertifikat benutzt wurde
UTCTime	coordinated universal time	ASN.1-Typ für Datums- und Zeitformate, Weltzeit
	validation of certificate signature	Bit im Feld der Zertifikatserweiterung <i>keyUsage</i> , das die Schlüsselnutzungsart "Prüfung der Zertifikatssignatur einer Zertifizierungsstelle" anzeigt
	<i>keyCertSign</i>	
	validation of CRL signature	Bit im Feld der Zertifikatserweiterung <i>keyUsage</i> , das die Schlüsselnutzungsart "Prüfung der Sperrlistensignatur einer Zertifizierungsstelle" anzeigt
	<i>cRLSign</i>	
	validity	Zertifikatsfeld, das den Gültigkeitszeitraum des Zertifikates enthält
	version	Zertifikatsfeld, das die Versionsnummer des Zertifikats- formates enthält
WWW	world wide web	Dienst zum Laden von graphischen Informationen über das Internet
YY	year	zweistellige Jahreszahl
YYYY	year	vierstellige Jahreszahl
ZS		Zertifizierungsstelle

## A 2. LITERATUR

- [A1 99] BSI: Signatur-Interoperabilitätsspezifikation SigI, Abschnitt A1 Zertifikate, Version 4.0, Stand 30.04.99
- [A2 99] BSI: *Signatur-Interoperabilitätsspezifikation SigI, Abschnitt A2 Signatur*, Januar 1999
- [A5 99] BSI: Signatur-Interoperabilitätsspezifikation SigI, Abschnitt A5 Verzeichnisdienst, Version 3.0, Stand 30.04.99
- [A6 99] BSI: *Signatur-Interoperabilitätsspezifikation SigI, Abschnitt A6 Gültigkeitsmodell für digitale Signaturen*, Januar 1999
- [ANS X9.30] ANSI X9.30-199x: *Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry, Part 1: The Digital Signature Algorithm (DSA)*, 1992
- [ANS X9.31] ANSI X9.31-199x: *Public Key Cryptography Using Reversible Algorithms for the Financial Services Industry, Part 1: The RSA Signature Algorithm*, 199?
- [ANS X9.55] ANSI X9.55-1995: *Public Key Cryptography for the Financial Services Industry, Extensions to Public Key Certificates and Revocation Lists*, Dec. 1995
- [ANS X9.62] ANSI X9.62-199x: *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*, 1997
- [CCITT X.208 88] CCITT X.208: *Specification of Abstract Syntax Notation One (ASN.1)*, 1988
- [DIN SigG/V 98] DIN NI-17.4: *Spezifikation der Schnittstelle zu Chipkarten mit Digitaler Signatur-Anwendung/Funktion nach SigG und SigV, Version 0.98*, Juli 1998
- [ISO CD 15782] ISO CD 15782: *Security Management and General Banking Operations*, 1998
- [ISO/IEC 14888] ISO/IEC 14888: *IT-Security Techniques - Digital Signatures With Appendix- Part 3: Certificate-Based Mechanisms*, 1997

- 
- [ISO/IEC 9796-2] ISO/IEC 9796-2: *IT-Security Techniques - Digital Signatures Schemes Giving Message Recovery- Part 2: Mechanisms using a hash-function*, 1996
- [ITU-T X.411] ITU-T X.411: *Information Technology - Message Handling Systems - Message Transfer System Abstract service definition and procedures*, 19??
- [ITU-T X.500 97] ITU-T X.500: *Information Technology - Open Systems Interconnection - The Directory: Overview of concepts, models and services*, 1997
- [ITU-T X.501 97] ITU-T X.501: *Information Technology - Open Systems Interconnection - The Directory: Models*, 1997
- [ITU-T X.509 97] ITU-T X.509: *Information Technology - Open Systems Interconnection - The Directory: Authentication Framework*, 1997
- [ITU-T X.520 95] ITU-T X.520: *Information Technology - Open Systems Interconnection - The Directory: Selected Attribute Types*, 1995
- [ITU-T X.660 92] ITU-T X.660: *Information Technology - Open Systems Interconnection - Procedures for the operation of OSI Registration Authorities: General procedures*, 1992
- [ITU-T X.681 94] ITU-T X.681: *Information Technology - Abstract Syntax Notation One (ASN.1): Information object specification*, 1994
- [ITU-T X.690 94] ITU-T X.690: *Information Technology - ASN.1 Encoding Rules - Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*, 1994
- [MISPC 97] William Burr, Donna Dodson, Noel Nazario, W. Timothy Polk: *Minimum Interoperability Specification for PKI Components*, Version 1, June 1997
- [MKAT 97] *Regulierungsbehörde für Telekommunikation und Post: Maßnahmenkatalog für digitale Signaturen*, Version 1.0, November 1997
- [MTRUST 96] Fritz Bauspieß, *TelTrusT: MailTrusT Spezifikation*, Version 1.1, Dezember 1996
- [OIW 95] OIW, *Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 12 - OS Security*, June 1995
- [PKCS1 93] RSA Laboratories, Technical Note, *PKCS #1: RSA Encryption Standard*, Version 1.5, November 1993

- [PKCS1 98] RSA Laboratories, Technical Note, *PKCS #1: RSA Encryption Standard*, Version 2.0, September 1998
- [PKCS7 93] RSA Laboratories, Redwood City, California: *The Public-Key Cryptography Standards (PKCS)*, November 1993
- [PKIX QC 99] S. Santesson, W. Polk, P. Barzin, M. Nystrom: *Internet X.509 Public Key Infrastructure -Qualified Certificates Profile*, October 1999
- [PKIX ECDSA 97] L. Bassham, D. Johnson: *Internet Public Key Infrastructure -Representation of Elliptic Curve Digital Signature Algorithm (ECDSA) Keys and Signatures in Internet X.509 Public Key Infrastructure Certificates*, November 1997
- [PKIX OCDP 98] P. Hallam-Baker, and W. Ford: *Internet Public Key Infrastructure - Open CRL Distribution Process (OpenCDP)*, April 1998
- [PKIX OCSP 97] Michael Myers: *Internet Public Key Infrastructure -Online Certificate Status Protocol (OCSP)*, November 1997
- [PKIX OCSP 98] Michael Myers: *Internet Public Key Infrastructure -Online Certificate Status Protocol (OCSP)*, August 1998
- [PKIX OP-FTPHTTP 98], Russ Housley, Paul Hoffman *Internet Public Key Infrastructure - Operational Protocols - FTP and HTTP*, July 1998
- [PKIX OP-LDAP 98] Tim Howes, S. Boeyen, P. Richard: *Internet Public Key Infrastructure - Operational Protocols - LDAPv2*, September 1998
- [RFC 1035 87] P. Mockapetris: *Domain Names - Implementation and Specification*, 1987
- [RFC 1422 93] S. Kent: *Privacy Enhancement for Internet Electronic Mail - Part II: Certificate-Based Key Management*, February 1993
- [RFC 1630 94] T. Berners-Lee: *Universal Resource Identifiers in WWW*, 1994
- [RFC 1959 96] T. Howes, and M. Smith: *An LDAP URL Format*, June 1996
- [RFC 2052 96] A. Gulbrandsen, and M. Smith: *A DNS RR for specifying the location of services (DNS SRV)*, October 1996
- [RFC 2068 97] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee: *Hypertext Transfer Protocol - HTTP/1.1*, January 1997
- [RFC 2247 98] S. Kille, M. Wahl, A. Grimstad, R. Huber and S. Sataluri: *Using Domains in LDAP/X.500 Distinguished Name*, January 1998

- [RFC 2459 99] R. Housley, W. Ford, W. Polk, and D. Solo: *Internet X.509 Public Key Infrastructure - Certificate and CRL Profile*, 1999
- [RFC 2560 99] Myers, Ankney, Malpani, Galperin, Adams: *Internet X.509 Public Key Infrastructure - OCSP*, 1999
- [RFC 791 81] J. B. Postel: *Internet Protocol*, 1981
- [RFC 822 82] David H. Crocker: *Standard for the Format of ARPA Internet Text Messages: Message Encryption and Authentication*, August 1982
- [RIPEMD-160 96] H. Dobbertin, A. Bosselaers, and B. Preneel: *A strengthened version of RIPEMD*, April 1996
- [SEC 98] GMD, SECUDE-Tool: <http://www.darmstadt.gmd.de/secude/>, 1998
- [SigG 97] BRD: *Gesetz zur Regelung der Rahmenbedingungen für Informations- und Kommunikationsdienste (Informations- und Kommunikationsdienste-Gesetz - IuKDG), Artikel 3, Gesetz zur digitalen Signatur (Signaturgesetz - SigG)*, Juli 1997
- [SigV 97] BRD: *Verordnung zur digitalen Signatur (Signaturverordnung - SigV)*, Juli 1997
- [TS ZF 98] TeleSec, Deutsche Telekom AG: *Zertifikatsformate im Zertifizierungsbereich Signaturgesetz*, Mai 1998