BSI

Bundesamt für Sicherheit in der Informationstechnik

SPEZIFIKATION ZUR ENTWICKLUNG INTEROPERABLER VERFAHREN UND KOMPONENTEN NACH SIGG/SIGV

SIGNATUR-INTEROPERABILITÄTSSPEZIFIKATION SIGI

ABSCHNITT A1 ZERTIFIKATE

STAND: 30.04.99
VERSION 4.0

Godesberger Allee 183, 53175 Bonn - Postfach 20 03 63, 53133 Bonn
Telefon: (0228) 9582 - 0, Telefax: (0228) 9582 - 400
Internet: www.bsi.bund.de

ABSCHNITT A1
ZERTIFIKATE

Andreas Berger, Alfred Giessler, Petra Glöckner, Wolfgang Schneider
GMD – Forschungszentrum Informationstechnik GmbH
Institut für Telekooperationstechnik
Dolivostr. 15, 64293 Darmstadt

INHALTSVERZEICHNIS

Hinweis

Die Anhänge zu diesem Dokument befinden sich in dem separaten Dokument bsicer02.doc.

1 EINLEITUNG

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 Signatur-

gesetz 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 grundliegende 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 Zertifikatserstellung. 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.

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 Zertifikatserweiterungen 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 wei-

tere 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 Entwickung eines entsprechenden PKI-Profils (PKI, public key infrastructure) [PKIX PRO 97] 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 Zertifkatsformaten 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 Zertifkaten 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 SigI-Spezifikation auch nur Profilfestlegungen für die Strukturen und Formate von Signaturschlüssel- und ihnen zugehörige Attributzertifkate 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 Verwechselungsmöglichkeit 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 Zertifzierungsstelle 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 Zertifzierungsstelle 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 Zertifikats- bzw. Attributzertifkatskomponenten realisiert werden. Außerdem werden im Rahmen dieses SigI-Profils (Signaturgesetz-Interoperabilität) noch weitere spezielle Angaben benötigt, wie z.B. das Erstellungsdatum eines Zertifikates oder eine Schlüsselnutzungsart für Verzeichnisdienste. Hierfür wurden SigI-spezifischen Objektbezeichner unter dem Objektbezeichnerzweig *id-sigi* festgelegt, der seinerseits ein Zweig des TeleTrust-Vereins ist. Unter diesem Zweig wurden die Zweige *cp* (1) für Zertifizierungsrichtlinien, *kp* (2) für Schlüsselnutzungsarten, *at* (3) für Attribute und private Erweiterungen, sowie on (4) für OTHER-NAME-Definitionen wie beispielsweise für Personendaten 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- und Attributzertifikate

#	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 Schlüssels des Signaturschlüsselinhabers sowie der öffentliche Schlüssels 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

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 implemtierungsnähere 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 “SigI-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 Zertifkatfeldes gemäß der X.509v3-Empfehlung. Für SigI-spezifische Zertifkatsfelder, 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 [PKIX PRO 97], [DIN SigG/V 98] oder [MTRUST 96] getroffen wurden, zu beachten.

SigI-Konformitätsanforderungen

Dieser Unterabschnitt enhält Informationen über Einschränkungen und Anwendung der allgemeinen Konformitätsanforderungen hinsichtlich der durch X.509v3 möglichen Optionen. Im Rahmen der Signatur-Interoperabilitätsspezifikation SigI 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. *policy constraints* oder name *constraints* verboten.

Desweiteren enthält dieser Unterpunkt implementations-technische Informationen über einzelne Zertifikatsfelder und deren Unterstrukturen in einer taballelarischen Übersicht (Muster siehe Tabelle 2). Die erste Spalte enthält den ASN.1-Bezeichner des betreffenden Zertifikatfeldes. 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 Zertifikatfeldes. 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 Zertifikatfeldes 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 beschrieben 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 EINZELWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ			KLASSIFIKATION			
	(BEISPIELE)	(BEISPIELE)	[BYTES]	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
		Tag Länge *Wert*												

2.1 Signaturalgorithmus

Zweck

Das Signaturfeld *signatureAlgorithm* vom Typ *AlgorithmIdentifier* 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    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 *tbsCertificate*-Struktur enthalten.

SigI-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 Komponete *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. Weitere Einzelheiten zu dem Thema “Signaturalgorithmen” sind in der Teilspezifikation “A2 Signatur” [A2 99, 6] angegeben.

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}
```

Tabelle 3: Implementations-technische Informationen über *signatureAlgorithm*

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	ZERTIFIKATSTYP				RELEVANZ		
	(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	v	v	v	v	v		
Signature-Algorithm Algorithm Parameters	rsaSign.Withripemd160 SEQUENCE { {1 3 36 3 3 1 2 }, NULL }	 **30** 0A **06** 06 *2B 24 03 03 01 02* **05** 00	11	v	v	v	v	v		
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	v	v	v	v	v		

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").

SigI-Konformitätsanforderungen

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

Tabelle 4: Implementations-technische Informationen über *signature*

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

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.

SigI-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	optional
version	v	v	v	v	v			Subject	v	v	v	v	v		
serialNumber	v	v	v	v	v			SubjectPublicKeyInfo	v	v	v	v	v		
signature	v	v	v	v	v			IssuerUniqueID	v	v	v	v		v	
issuer	v	v	v	v	v			SubjectUniqueID	v	v	v	v		v	
validity	v	v	v	v	v			Extensions	v	v	v	v	v		

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.

SigI-Konformitätsanforderungen

Bei der Erzeugung von Zertifikaten ist dieVerwendung 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 EINZELWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	ZERTIFIKATSTYP				RELEVANZ		
	(BEISPIELE)	(BEISPIELE)	3	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional
version	2	**02** 01 *02*	3	v	v	v	v	v		

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. Zertifkate 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.

SigI-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]	ZERTIFIKATSTYP				RELEVANZ		
	(BEISPIELE)	(BEISPIELE)	17	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional
serialNumber	0	**02** 01 *00*	3	v	v	v	v	v		
	1	**02** 01 *01*	3							
	...	**02**	3-17							
	2^{8*15-1}-1	**02** 0F *7F FF FF FF FF FF FF FF FF FF FF FF FF FF FF*	17							

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" 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.

SigI-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 EINZELWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	ZERTIFIKATSTYP				RELEVANZ		
	(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	v	v	v	v	v		
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	v	v	v	v	v		
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	v	v	v	v	v		

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 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.

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

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-Zeittypen *UTCTime* (coordinated universal time, Weltzeit) oder *GeneralizedTime* (allgemeines Datums- und Zeitformat) repräsentiert werden, die Datums- und Zeitangaben bis auf Sekun-

dengenauigkeit 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 Zeitpunken 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 YYYYMMDD HHMMSSZ 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 *GeneralizedTime*	HH	Stunde 00, 01, ..., 23
YY	letzte zwei Ziffern der Jahreszahl, nur bei *UTCTime*	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.

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

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.

SigI-Konformitätsanforderungen

Bei der Erzeugung von Zertifikaten ist die Verwendung des allgemeinen Datums- und Zeitformates *GeneralizedTime* zu verwenden, bei dessen Kodierung das Format YYYYMMDD HH MMSSZ genommen werden soll.

Tabelle 12: Implementations-technische Informationen über *validity*

BEZEICHNER	WERTEBEREICH EINZELWERTE	HEXADEZIMALCODE	LÄNGE [BYTES]	ZERTIFIKATSTYP				RELEVANZ		
	(BEISPIELE)	(BEISPIELE)	36	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional
Validity	SEQUENCE {	**30** 22	36	v	v	v	v	v		
NotBefore	GeneralizedTime	**18** 0F	17							
	"199801010000000Z",	*31 39 39 38 30 31 30 31 30 30 30 30 30 30 5A*								
NotAfter	GeneralizedTime	**18** 0F	17	v	v	v	v	v		
	"200301010000000Z" }	*32 30 30 33 30 31 30 31 30 30 30 30 30 30 5A*								

2.3.6 TECHNISCHE NAMEN VON ZERTIFIKATSINHABERN

Zweck

Das *subject*-Namensfeld dient zur technischen 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 *PrintablString*, *TeletexString*, *UniversalString* und *BMP*-

String ist. Eine Übersicht der möglichen Objektbezeichner für *AttributeType* ist in der Tabelle 9 im Abschnitt 2.3.4 gegeben.

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)) }
```

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 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.

SigI-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 technischen Benennung des Zertifikatsinhabers. Weitere Namen eines Zertifikatsinhabers können danach nur zusätzlich zum *subject*-Feld im *extensions*-Feld unter *subjectAltName* (siehe Abschnitt 2.3.9.5) enthalten sein. Hier 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 soll für

Endbenutzer mit dem technischen Namen des Zertifikatsinhabers belegt werden; es muß mindestens die Attribute *commonName* und *countryName* enthalten. 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 ein weiteres Attribut, wie beispielsweise eine Seriennummer, 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]	ZERTIFIKATSTYP				RELEVANZ		
	(BEISPIELE)	(BEISPIELE)	513	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional
subject	SEQUENCE OF {	**30** 63	101	v	v	v	v	v		
countryName	SET OF { SEQUENCE { { 2 5 4 6 },	**31** 0B **30** 09 **06** 03 *55 04 06*	13							
value	"DE" } }	**13** 02 *44 45*								
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	v	v	v	v			v
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	v	v	v	v			v
SerialNumber	SET OF { SEQUENCE { { 2 5 4 5 }, "1" } }	**31** 0A **30** 08 **06** 03 *55 04 05* **13** 01 *31*	12	v	v	v	v			v
Title	SET OF { SEQUENCE { { 2 5 4 12 }, "Dr." } },	**31** 0C **30** 0A **06** 03 *55 04 0C* **13** 03 *44 72 2E*	14	v	v	v	v			v
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	v	v	v	v	v		

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 gegebenfalls die Personalnummer:
`CN=Vorname Name, SN=Personalnummer, OU=Organisationseinheit, O=Firma, C=DE`

(4) Technischer Name einer Person als Mitglied der Kassenärztlichen Vereinung über Mitgliedsnummer und Namen der Vereinigung:
`CN=Vorname Name, T=Titel, SN=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`

Die Länge der *AttributeValue*-Stringtypen ist durch den Systemparameter *maxSize* festgelegt, der für die einzelnen Attribute gemäß der Tabelle 11 begrent ist. Hieraus ergeben sich die in der Längenspalte angegenen maximalen Längen der Attribute inklusive der ASN.1-Kontrollinformationen, die eine Länge von 11 Bytes haben. Für das *subject*-Feld bestehend aus nur obligatorischen Attributen ergibt sich aus dem Längenfeld der Tabelle 9 die Maximallänge von 90 Bytes und für das *subject* -Feld bestehend aus allen Attributen der Tabelle 9 von 513 Bytes.

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 Blockauffüllregeln. 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 ellyptischen 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 Objketbezeichner *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 Algortihmus 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-ecPublic-Key* können Parameter im Zertifikat durch die Struktur *ECParameters* explizit spezifiziert werden. Weitere Informationen zu der Parameterstruktur *ECParameters* 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.

SigI-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 policie*s identifiziert wird, die signaturgesetzkonformen Algorithmen beschreiben (siehe Abschnitt 2.3.9.4).

Tabelle 14: Implementations-technische Informationen über *subjectPublicKeyInfo*

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

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.

SigI-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]	ZERTIFIKATSTYP				RELEVANZ		
	(BEISPIELE)	(BEISPIELE)		Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional
IsssuerUniqueID				v	v	v	v		v	
SubjectUniqueID				v	v	v	v		v	

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 Sperrrlisten 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 alternativ 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.

SigI-Konformitätsanforderungen

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

SigI-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 SigI-spezifische proprietäre Erweiterungen stets als *non-critical* markiert. Im Rahmen dieses Profils werden SigI-spezifische proprietäre Erweiterungen für die Anforderungen (13)-(17) der Tabelle 1 definiert.

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

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

Tabelle 16: Implementations-technische Informationen über Erweiterungen

ERWEITERUNG	ZERTIFIKATSTYP				RELEVANZ			KLASSIFIKATION			
	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
BasicConstraints	v	v	v	v	v			v		v	
KeyUsage	v	v	v	v	v			v		v	
ExtKeyUsage		v	v		v			v		v	
CertificatePolicies	v	v	v	v	v			v			v
SubjectAltName	v	v	v	v	v			v			v
IssuerAltName	v	v	v	v			v	v			v
AuthorityKeyIdentifier	v	v	v	v	v			v			v
SubjectKeyIdentifier	v	v	v	v			v	v			v
CRLDistributionPoints	v	v	v	v			v	v			v
PolicyMappings	v	v	v	v			v	v			v
PolicyConstraints	v	v	v	v		v		v		v	
SubjectDirectoryAttributes	v	v	v	v			v	v			v
NameConstraints	v	v	v	v		v		v		v	
PrivateKeyUsagePeriod	v	v	v	v			v	v		v	
AuthorityInfoAccess	v	v	v	v		v			v		v
LiabilityLimitationFlag	v	v	v	v	v				v		v
DateOfCertGen	v	v	v	v			v		v		v
Procuration	v	v	v	v			v		v		v
Admission	v	v	v	v			v		v		v
MonetaryLimit	v	v	v	v			v		v		v
DeclarationOfMajority	v	v	v	v			v		v		v
ICCSN	v	v	v	v			v		v		v
PKReference	v	v	v	v			v		v		v
Restriction	v	v	v	v			v		v		v

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 Zertifkatssignaturen 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.

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

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 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.

SigI-Konformitätsanforderungen

Im Rahmen des SigI-Profils 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 Authenisierungszwecke 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		KOMBINATION		
BITNAME		BIT#	1	2	3
Digital Signature		0			
Non Repudiation	Anwender, Zeitstempeldienst, Verzeichnisdienst	1	v		v
Key Encipherment		2			
Data Encipherment		3			
Key Agreement		4			
Key CertSign	Zertifizierungsstellen	5		v	
CRL Sign	Verzeichnisdienst	6			v
Encipher Only		7			
Decipher Only		8			

Tabelle 20: Implementations-technische Informationen über *keyUsage*

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ			KLASSIFIKATION			
	(BEISPIELE)	(BEISPIELE)	[BYTES] 16	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
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	v				v			v		v	
keyUsage extnId critical extnValue cRLSign+ nonRepud.	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			v		v			v		v	
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		v		v	v			v		v	

2.3.9.3 Anwendungsabhängige Verwendungszwecke des Schlüsselpaares

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.

SigI-spezifische Objektbezeichner für Signaturgesetz-Interoperabilität sind unter dem Objektbezeichnerzweig *id-sigi* festgelegt, der seinerseits ein Zweig von TeleTrust ist Das Verfahren für die Vergabe von neuen Objektbezeichnern, zum Beispiel für Schlüsselnutzungsarten, ist in Kapitel 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-sigi OBJECT IDENTIFIER              ::=  { 1 3 36 8 }
id-sigi-kp OBJECT IDENTIFIER           ::=  { 1 3 36 8 2 }
id-sigi-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 Hinweisfeld 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.

SigI-Konformitätsanforderungen

Die Benutzung des Objektbezeichners *id-kp-time-Stamping* ist bei der Generierung von Zertifikaten der Zeitstempeldienste obligatorisch. Die Benutzung des Objektbezeichners *id-sigi-kp-directoryService* ist bei der Generierung von Zertifikaten der Verzeichnisdienste obligatorisch. In diesen Zertifikaten muß die *extKeyUsage*-Erweiterung stets als *critical* markiert werden.

Tabelle 21: Implementations-technische Informationen über *extKeyUsage*

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

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 sognannter *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 }
-- SigI-Definitionen
id-sigi                  OBJECT IDENTIFIER   ::= { 1 3 36 8 }
id-sigi-cp               OBJECT IDENTIFIER   ::= { 1 3 36 8 1 }
id-sigi-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 dringenst 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-unotice* 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.

SigI-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. Signaturgesetzkonforme 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-sigi-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- und/oder Attributzertifikate 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 Attributinhalte. 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 2.3.9.15.11 beschrieben.

Tabelle 22: Implementations-technische Informationen über *certificatePolicies*

BEZEICHNER	WERTEBEREICH EINZELWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ			KLASSIFIKATION			
	(BEISPIELE)	(BEISPIELE)	[BYTES] 60	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
Certific.Policies	SEQUENCE {	**30** 37	57	v	v	v	v	v			v			v
ExtnId	{ 2 5 29 32 },	**06** 03 *55 1D 20*												
Critical	FALSE,													
ExtnValue	OCTET STRING	**04** 31												
	SEQ. OF {SEQ. {	**30** 2E **30** 2C												
PolicyIdentifier	{ 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.de/Fachinfo/Digitalsign/neu/policy.htm" } } } }	**16** 37 *68 74 74 70 3A 2F 2F 77 77 77 ...*								v				

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 (siehe Kapitel 3).

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 Domänennamen 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-Identifier	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]

Im Rahmen der SigI-Spezifikation wird der Datentyp *PersonalData* als Instanz der *otherName*-Informationsobjektklasse definiert, der in Abgrenzung zu den technischen Namen im *subject*-Feld, identitätsgebundene Merkmale wie den gesetzlichen Personennamen, ein Pseudonym, eine Seriennummer, das Geburtsdatum, den Geburtsort, das Geschlecht oder die Wohnanschrift enthalten kann. Durch diese Struktur werden die in der Tabelle 1 in den Punkten (1) und (3-5) genannten Anforderungen des Signaturgesetzes erfüllt. Prinzipiell können durch die Struktur *PersonalData* alle Informationen, die in einem Personalausweis enthalten sind, dargestellt werden. An dieser Stelle sei auf die neue PKIX-Arbeitsgruppe "Qualified Certificates" [PKIX CP 98] hingewiesen, die ebenfalls die Definition von Strukturen für Personendaten behandelt. Die Struktur *PersonalData* sollte aus Gründen der internationalen Interoperabilität an die entsprechenden PKIX-Definitionen angepaßt werden, sobald diese als hinreichend stabil betrachtet werden können.

Im Rahmen der SigI-Spezifikation wird somit eine klare Unterscheidung zwischen technischen und gesetzlichen Namen getroffen. Ein Signaturschlüsselinhaber kann danach mehrere technische Namen, aber nur einen gesetzlichen Namen haben

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 }
  id-sigi-on OBJECT IDENTIFIER ::= { 1 3 36 8 4 }
  id-sigi-on-personalData     OBJECT IDENTIFIER
                          ::=  { 1 3 36 8 4 1 }
  PersonalData        ::= SEQUENCE {
    nameOrPseudonym         NameOrPseudonym,
    nameDistinguisher       [0]  INTEGER              OPTIONAL,
    dateOfBirth             [1]  GeneralizedTime      OPTIONAL,
    placeOfBirth            [2]  DirectoryString      OPTIONAL,
    gender                  [3]  PrintableString      OPTIONAL,
    postalAddress           [4]  DirectoryString      OPTIONAL }
  NameOrPseudonym     ::= CHOICE {
    surAndGivenName         SEQUENCE {
      surName               DirectoryString,
      givenName             SEQUENCE OF DirectoryString },
    pseudoNym               DirectoryString }
```

Statische Semantik

Bei Namensgleichheit ist die Benutzung der *nameDistinguisher*-Komponente obligatorisch. Das Geburtsdatum *dateOfBirth* darf nur Datumsangaben gemäß YYYYMMDD enthalten. Für die Komponente *gender* sind die zwei Werte "F" für female (weiblich) und "M" für male (männlich) vordefiniert.

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 Zertifkates 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, anderfalls 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.

SigI-Konformitätsanforderungen

Die obligatorische *subjectAltName*-Erweiterung muß bei der Erstellung von Zertifikaten benutzt und als *non-critical* gekennzeichnet werden. Identitätsgebundene Merkmale müssen in der Datenstruktur *personalData* eingetragen werden. Hierbei ist nur der Personenname bzw. das Pseudonym obligatorisch. Alle anderen Angaben wie die Seriennummer, der Geburtsort, das Geburtsdatum, das Geschlecht oder die Wohnanschrift sind optional, wobei die Seriennummer ggf. zur Vermeidung von Mehrdeutigkeiten zu verwenden, d.h. obligatorisch ist. 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. Zertifizierungsstellen müssen bei der Ausstellung von Zertifikaten alle hierfür erforderlichen Informationen prüfen. Zertifikatsinhaber sollten jegliche Änderungen dieser Informationen unverzüglich der zugehörigen Zertifizierungsstelle mitteilen. Weitere Informationen über Namenskonventionen sind im Anhang I zu finden.

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
```

(3) OtherName als Personenname mit Geburtsdatum:

```
person-1 PersonalData {
  nameOrPseudonym {
    surAndGivenName {surname "Name", givenName "Vorname" },
  dateOfBirth "1900.01.01" } }
```

(4) OtherName als Pseudonym mit Namenunterscheidungskennung:

```
person-2 PersonalData {
  nameOrPseudonym {
  pseudoNym "Rumpelstilzchen" }
  nameDistinguisher 1 } }
```

Tabelle 24: Implementations-technische Informationen über *subjectAltName*

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ			KLASSIFIKATION			
	(BEISPIELE)	(BEISPIELE)	[BYTES] 500	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
subjectAltName	SEQUENCE {	**30** 43	69	v	v	v	v	v			v			v
extnId	{ 2 5 29 17 },	**06** 03 *55 1D 11*												
critical	FALSE,													
extnValue	OCTET STRING	**04** 3C												
otherName	SEQ OF {[0] SEQ {	**30** 3A **80** 2C **30** 2A												
type-id	{ 1 3 36 8 4 1 },	**06** 05 *2B 24 08 04 01*												
value	[0] EXPLICIT SEQ. {	**A0** 21 **30** 1F												
surAndGivenName	SEQUENCE {	**30** 11												
surName	"Name",	**13** 04 *4E 61 6D 65*												
givenName	SEQ OF{"Vorname" } },	**30** 09 **13** 07 *56 6F 73 6E 61 6D 65*												
dateOfBirth	[1] "19000101" } }	**81** 0A **18** 08 *31 39*								v				
	} },	*30 30 30 31 30 31*												
rfc822Name	[1] "ca@cert.de" } }	**81** 0A *63 61 40 63 65 72 74 2E 64 65*												

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 Zertifzierungsstelle 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 *distingushed names* [RFC 2247 98]).

Der im Rahmen der SigI-Spezifikation definierte Datentyp *PersonalData* kann auch als *issuerAltName* verwendet werden, um den gesetzlichen Namen bzw. das Pseudonym der Zertifizierungsstelle in den von ihr ausgestellten Zertifikaten einzutragen.

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 }

id-sigi-on          OBJECT IDENTIFIER   ::=  { 1 3 36 8 4 }
```

```
id-sigi-on-personalData     OBJECT IDENTIFIER
                        ::= { 1 3 36 8 4 1 }

PersonalData        ::= SEQUENCE {
    nameOrPseudonym         NameOrPseudonym,
    nameDistinguisher       [0]  INTEGER             OPTIONAL,
    dateOfBirth             [1]  GeneralizedTime     OPTIONAL,
    placeOfBirth            [2]  DirectoryString     OPTIONAL,
    gender                  [3]  PrintableString     OPTIONAL,
    postalAddress           [4]  DirectoryString     OPTIONAL }

NameOrPseudonym     ::= CHOICE {
    surAndGivenName         SEQUENCE {
      surName               DirectoryString(,
      givenName             SEQUENCE OF DirectoryString,
    pseudoNym               DirectoryString }
```

Statische Semantik

Bei Namensgleichheit ist die Benutzung der *nameDistinguisher*-Komponente obligatorisch. Das Geburtsdatum *dateOfBirth* darf nur Datumsangaben gemäß YYYYMMDD enthalten. Für die Komponente *gender* sind die zwei Werte "F" für female (weiblich) und "M" für male (männlich) vordefiniert.

Allgemeine Konformitätsanforderungen

Es gelten die gleichen Aussagen wie für *subjectAltName.*

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

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 gegebenfalls 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 Zertifzierungspfaden unterstützen.

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

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 mehrereren 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.

SigI-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 *subjectyKeyIdentifier*

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ			KLASSIFIKATION			
	(BEISPIELE)	(BEISPIELE)	[BYTES] 31	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
subjectKeyId	SEQUENCE {	**30** 1D	31	v	v	v	v			v	v			v
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*												

2.3.9.9 Informationen zur Beschaffung von Sperrlisten

Zweck

Die *cRLDistribututionPoints*-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.

SigI-Konformitätsanforderungen

Die optionale *cRLDistribututionPoints*-Erweiterung muß als *non-critical* markiert werden, so daß statt der Benutzung von Sperrlisten auch andere Mechanismen wie z.B. On-line Prüfdienste, 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.

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 EINZELWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ			KLASSIFIKATION			
	(BEISPIELE)	(BEISPIELE)	[BYTES] 100	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
CRLDist.Point	SEQUENCE {	**30** 29	*43*	v	v	v	v			v	v			v
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 ...*												

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 *policyMappings*-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

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

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 (siehe Abschnitt 3.9) angegeben werden.

SigI-Konformitätsanforderungen

SigI-konforme Zertifizierungsstellen können die optionale Erweiterung *subjectDirectory-Attributes* bei der Erstellung von Zertifikaten unterstützen.

Beschränkungen, Zulassungen oder andere Zusatzinformationnen 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 Signaturschlüsselzertifikat selbst, sondern in einem Attributzertifikat enthalten ist, so muß das betreffende Attributzertifikat Teil des signierten Dokuments sein.

Im Rahmen des SigI-Profils werden die zusätzlichen Attribute *atProcuration* (Vertretungsmacht), *atAdmission* (Zulassungsinformation), *atMonetaryLimit* (monetäre Beschränkung) und *atRestriction* (sonstige Einschränkungen) für die Themenbereiche (13) bis (17) der Tabelle 1 festgelegt. Diese Attribute sind im Abschnitt 3.9 des Kapitels "Attributzertifikate" definiert.

Tabelle 30: Implementations-technische Informationen über *subjectDirectoryAttributes*

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

2.3.9.12 Beschränkungen von Zertifizierungsrichtlinen

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.

SigI-Konformitätsanforderungen

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

Tabelle 31: Implementations-technische Informationen über *policyConstraints*

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

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.

SigI-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			
	(BEISPIELE)	(BEISPIELE)	[BYTES]	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
name Constraints				v	v	v	v		v		v		v	

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 *critical* gekennzeichnet werden.

SigI-Konformitätsanforderungen

Die Benutzung der *privateKeyUsagePeriod*-Erweiterung ist bei der Erstellung von Zertifikaten optional.

Tabelle 33: Implementations-technische Informationen über *privateKeyUsagePeriod*

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

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 SigI-Profils 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. SigI-konforme Anwendungen sollten alle hier definierten Erweiterungen erkennen und verarbeiten können, auch wenn sie aus internationalen Interoperabilitätsgründen als *non-critical* gekennzeichnet sind.

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 *AuthorityInfoAccess-Syntax 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.

SigI-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.

Tabelle 34: Implementations-technische Informationen über *authorityInfoAccess*

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ			KLASSIFIKATION			
	(BEISPIELE)	(BEISPIELE)	[BYTES] 128	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
authorityInfoAccess	SEQUENCE {	**30** 5D	95	v	v	v	v			v		v		v
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 ...*												

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 }
```

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

2.3.9.15.3 ERSTELLUNGSDATUM EINES ZERTIFIKATES

Zweck

Die SigI-spezifische private Erweiterung *dateOfCertGen* dient zur Anzeige des Erstellungsdatum eines Zertifikates. Sie wird durch den Objektbezeichner *id-sigi-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-sigi-at-dateOfCertGen }

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

id-sigi-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 2.3.5 beschriebene Format YYYYMMDDHHSSZ zu beachten.

SigI-Konformitätsanforderungen

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

Tabelle 36: Implementations-technische Informationen über *dateOfCertGen*

BEZEICHNER	WERTEBEREICH EINZELWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ			KLASSIFIKATION			
	(BEISPIELE)	(BEISPIELE)	[BYTES] 28	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
dateOfCert G extnID critical extnValue	SEQUENCE { { 1 3 36 8 3 1 }, FALSE, OCT. STRING "19980101000000Z" }	**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	v	v	v	v	v		v		v		v

2.3.9.15.4 VERTRETUNGSMACHT

Zweck

Die SigI-spezifische, private Erweiterung *procuration* dient zur Anzeige der Vertretungsmacht für eine dritte Person. Sie wird durch den Objektbezeichner *id-sigi-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-sigi-at-procuration }

id-sigi-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                CertifcateSerialNumber }

CertificateSerialNumber ::= INTEGER

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

SigI-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.

SigI-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			
	(BEISPIELE)	(BEISPIELE)	[BYTES] 256	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
procuration	SEQUENCE {	**30** 2D	47	v	v	v	v			v		v		v
extnID	{ 1 3 36 8 3 2 },	**06** 05 *2B 24 08 03 02*												
critical	FALSE,													
extnValue	OCTET STRING {	**04** 24												
ProcurationSyn	SEQUENCE OF {	**30** 22												
signingFor	[0] SEQUENCE {	**80** 20 **30** 1E												
type-id	{1 3 36 8 4 1 },	**06** 05 *2B 24 08 04 01*												
value	[0] EXPLICIT SEQ {	**A0** 15 **30** 13												
surAndGivenN	SEQUENCE {	**30** 11												
surName	"Name",	**13** 04 *4E 61 6D 65*												
givenName	SEQ OF {"Vorname"} } } } } }	**30** 09 **13** 07 *56 6F 73 6E ...*												

2.3.9.15.5 ZULASSUNG

Zweck

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

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

id-sigi-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.

SigI-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. SigI-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			
	(BEISPIELE)	(BEISPIELE)	[BYTES] 256	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
admission	SEQUENCE {	**30** 21	35	v	v	v	v			v		v		v
extnID	{ 1 3 36 8 3 3},	**06** 05 *2B 24 08 03 03*												
critical	FALSE,													
extnValue	OCTET STRING	**04** 18												
AdmissionSynt	SEQUENCE {	**30** 16												
contentsOfAdm	SEQUENCE OF {	**30** 14												
Admissions	SEQUENCE {	**30** 12												
ProfessionInfos	SEQUENCE OF {	**30** 10												
ProfessionInfo	SEQUENCE {	**30** 0E												
professionItems	SEQ OF {"Dipl.-Phys" } } } } } }	**30** 0C **13** 0A *44 69 70 6C 2E 2D 50 68 79 73*												

2.3.9.15.6 MONETÄRE BESCHRÄNKUNG

Zweck

Die SigI-spezifische, private Erweiterung *monetaryLimit* dient zur Anzeige einer monetären Beschränkung. Sie wird durch den Objektbezeichner *id-sigi-at-monetaryLimit* referenziert. Außer den Komponenten *amount* und *exponent*, aus denen sich der Beschränkungswert gemäß $amount \cdot 10^{exponent}$ 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-sigi-at-monetaryLimit }

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

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

SigI-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. SigI-konforme Systeme und Anwendungen müssen die private Erweiterung *monetaryLimit* erkennen und verarbeiten können.

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 EINZELWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ			KLASSIFIKATION			
	(BEISPIELE)	(BEISPIELE)	[BYTES] 24	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
MonetaryLimit	SEQUENCE {	**30** 16	24	v	v	v	v			v		v		v
ExtnID	{ 1 3 36 8 3 4 },	**06** 05 *2B 24 08*												
		03 04												
Critical	FALSE													
ExtnValue	OCTET STRING	**04** 0D												
	SEQUENCE {	**30** 0B												
Currency	"DEM",	**13** 03 *44 45 4D*												
Amount	1,	**02** 01 *01*												
Exponent	4 } }	**02** 01 *04*												

2.3.9.15.7 VOLLJÄHRIGKEIT

Zweck

Die SigI-spezifische, private Erweiterung *declarationOfMajority* dient zur Anzeige der Volljährigkeit eines Teilnehmers. Sie wird durch den Objektbezeichner *id-sigi-at-declarationOfMajority* 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 Betreffenden). 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-sigi-at-declarationOfMajority }
id-sigi-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 2.3.9.5 beschriebene Format YYYYMMDD zu beachten.

SigI-Konformitätsanforderungen

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

Tabelle 41: Implementations-technische Informationen über *declarationOfMajority*

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

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 SigI-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-sigi-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-sigi-at-ICCSN }
id-sigi-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.

SigI-Konformitätsanforderungen

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

Tabelle 42: Implementations-technische Informationen über *iCCSN*

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

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-sigi-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-sigi-at-pKReference }
id-sigi-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

SigI-Konformitätsanforderungen

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

Tabelle 43: Implementations-technische Informationen über *pKReference*

BEZEICHNER	WERTEBEREICH EINZELWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ			KLASSIFIKATION			
	(BEISPIELE)	(BEISPIELE)	[BYTES] 33	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional	Standarderweiterung	private Erweiterung	critical-Markierung	non-critical-Markierung
pKReference	SEQUENCE {	**30** 11	*19*	v	v	v	v			v		v		v
extnID	{ 1 3 36 8 3 7 },	**06** 05 *2B 24 08 03 07*												
critical	FALSE,													
extnValue	OCT. OCT.STR.	**04** 08 **04** 06												
	"DEPCA"1 }	*44 45 50 43 41 01*												

2.3.9.15.10 SONSTIGE EINSCHRÄNKUNGEN

Zweck

Die SigI-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-sigi-at-restriction }

id-sigi-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)) }
```

SigI-Konformitätsanforderungen

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

Tabelle 44: Implementations-technische Informationen über *restriction*

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

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. SigI-spezifische Objektbezeichner für Signaturgesetz-Interoperabilität sind unter dem Objektbezeichnerzweig *id-sigi* (1.3.36.8) festgelegt. Unterhalb von *id-sigi* gibt es bisher die Zweige *id-sigi-cp* (1.3.36.8.1) für Zertifizierungsrichtlinien, *id-sigi-kp* (1.3.36.8.2) für Schlüsselnutzungsarten, *id-sigi-at* (1.3.36.8.3) für private Erweiterungen und Attribute, sowie *id-sigi-on* (1.3.36.8.4) für OTHER-NAME-Typdefinitionen wie beispielsweise für Personendaten.

3 ATTRIBUTZERTIFIKATE

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

Nach dem Signaturgesetz [SigG 97, §2] ist ein Attribut-Zertifikat eine gesonderte digitale Bescheinigung, die unter eindeutiger Bezugnahme auf ein Signaturschlüssel-Zertifikat weitere Angaben enthält (Attribut-Zertifikat).

Die Gültigkeit eines Attribut-Zertifikates wird spätestens mit dem Ende der Gültigkeit des zugehörigen Signaturschlüssel-Zertifikates beendet. Da die Gültigkeit eines Zertifikates nach der Signaturverordnung [SigV, §7] auf maximal fünf Jahre begrenzt ist, ist auch die Gültigkeit eines Attribut-Zertifikates auf maximal fünf Jahre begrenzt. Ein Attributzertifikat kann unabhängig vom zugehörigen Signaturschlüssel-Zertifikat gesperrt werden.

Normen für Attributzertifikate

Die allgemeine Struktur für Attributzertifikate wurde in einer ersten Version von der ITU-T in der Empfehlung [ITU-T X.509 97 | ISO/IEC 9594-8 97] festgelegt. Im PKIX-Profil, das ein Internet-Profil für das erweiterte X.509v3-Zertifikatsformat beschreibt, werden keine Aussagen über Attributzertifikate gemacht.

Bedeutung von Attributzertifikaten

Die Authentifikation und Identifikation von Teilnehmern und Systemen in einer Sicherheitsinfrastruktur beruht auf der Bindung von Identitäten an deren öffentliche Schlüsselzertifikate. Dabei enthält jedes Schlüsselzertifikat die erforderliche Information zur Erbringung bestimmter kryptographischer Funktionen. Attribute, die einem Inhaber zugeordnet sind, können in einer separaten Struktur festgelegt werden, die von einer Zertifizierungsstelle zu signieren ist und technisch als Attributzertifikat bezeichnet wird. Ein Attributzertifikat ist somit eine von einer Zertifizierungstelle digitale Bescheinigung, die unter eindeutiger Bezugnahme auf ein Signaturschlüsselzertifikat weitere Angaben enthält.

3.1 Struktur von Attributzertifikaten

Jedes Attributzertifikat enthält die Namen *subject* des Inhabers und *issuer* des Erstellers, den Signaturalgorithmusbezeichner des Erstellers *signature*, den Gültigkeitszeitraum des Attributzertifikates *attrCertValidityPeriod*, die Versionsnummer *version* und die Seriennummer des Attributzertifikates *serialNumber*, sowie die eigentlichen Attribute *attributes*. Darüberhinaus können Attributzertifikate optionale Felder für Namensbezeichner *issuerUniqueID* und Attributzertifikatserweiterungen *extensions* enthalten.

ASN.1-Definitionen

```
AttributeCertificate   ::=  SEQUENCE {
   tbsAttributeCertificate      TBSAttributeCertificate,
```

```
    signatureAlgorithm          AlgorithmIdentifier,
    signature                   BIT STRING }

TBSAttributeCertificate ::= 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 }

AttCertValidityPeriod ::=   SEQUENCE {
    notBeforeTime               GeneralizedTime,
    notAfterTime                GeneralizedTime }

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 }
```

SigI-Konformitätsanforderungen

Der Inhaber eines Zertifikates kann mehrere Attributzertifikate besitzen, die mit einem Schlüsselzertifikate verbunden sind. Schlüssel- und Attributzertifikate eines Teilnehmers müssen nach [ITU-T X.509 97] nicht zwangsläufig von derselben Zertifizierungsstelle ausgestellt werden. Allerdings wird in der X.509-Empfehlung auch auf die möglichen Probleme im Zusammenhang mit der Überprüfung paralleler Zertifikatspfade hingewiesen.

Hinweis

Aus Gründen der Redundanzvermeidung werden in diesem Kapitel nur diejenigen Attribut-Zertifikatsfelder genauer beschrieben, die im Vergleich zu den Signaturschlüssel-Zertifikatsfeldern eine unterschiedliche ASN.1-Syntax besitzen. Hierzu gehören die Attribut-Zertifikatsfeldes *subject* und *attributes*. Bei allen anderen Feldern wird auf die Definitionen in den entsprechenden Abschnitten des Kapitels 2 verwiesen.

3.2 Versionsnummer

Zweck

Das Versionsfeld *version* dient zur Unterscheidung verschiedener Versionen eines Attributzertifikates. Gegenwärtig ist nur die Version v1 von der ITU-T festgelegt.

ASN.1-Definitionen

```
TBSAttributeCertificate     ::=  SEQUENCE {
   Version                       Version DEFAULT v1,
   ... }

Version                     ::=  INTEGER { v1(0) }
```

SigI-Konformitätsanforderungen

Bei der Erzeugung von Attributzertifikaten ist die Benutzung der Version X.509v1 obligatorisch.

3.3 Identität eines Zertifikatsinhabers

Zweck

Das Inhaberfeld *subject* eines Attributzertifikates enthält die Identität des Zertifikatsinhabers, wobei die Kopplung zwischen dem Attributzertifikat und dem Inhaber des zugehörigen Zertifikates auf zwei Arten erfolgen kann. Zum einen kann das Inhaberfeld durch den Typnamen *subjectName* unmittelbar den Namen des Zertifikatsinhabers spezifizieren. Bei der anderen Alternative wird durch den Typnamen *baseCertificateID* indirekt der Name des Zertifikatsinhabers über den Namen der Zertifizierungsstelle, die das Zertifikat ausgestellt hat, und die zugehörige Seriennummer des Zertifikates angeben. Die zweite Alternative ist deswegen möglich, weil ein einzelnes Zertifikat und damit dessen Inhaber eindeutig durch die Seriennummer und den Namen der ausstellenden Zertifizierungsstelle identifiziert werden.

ASN.1-Definitionen

```
TBSAttributeCertificate     ::=  SEQUENCE {
   ...,
   subject                CHOICE {
     baseCertificateID    [0]  IssuerSerial,
     subjectName          [1]  GeneralNames },
   ... }

IssuerSerial          ::= SEQUENCE {
   issuer                 GeneralNames,
   serial                 CertificateSerialNumber,
   issuerUID              UniqueIdentifier OPTIONAL }
```

SigI-Konformitätsanforderungen

SigI-konforme Zertifizierungstellen müssen bei der Erstellung von Attributzertifikaten das *subject*-Feld durch die Alternative *baseCertificateID* vom Typ *IssuerSerial* spezifizieren. Hierdurch wird die Verwendung eines eindeutigen Verweises auf das zugehörige Basiszertifikat erzwungen. Die Benutzung des optionalen Teilfeldes *issuerUID* ist verboten. Das Teilfeld *issuer* vom Typ *GeneralNames* muß genau ein Element vom Typ *DirectoryName* enthalten, das den Anforderungen des *issuer*-Feldes aus dem Signaturzertifikat genügt.

3.4 Name des Erstellers eines Attributzertifikates

Zweck

Das *issuer*-Namensfeld identifiziert die Instanz bzw. Zertifizierungsstelle, die das betreffende Attributzertifikat erstellt und signiert hat.

ASN.1-Definitionen

```
TBSAttributeCertificate     ::=  SEQUENCE {
    ...,
    issuer                GeneralNames,
    ... }
```

SigI-Konformitätsanforderungen

Das Teilfeld *issuer* vom Typ *GeneralNames* muß zumindest ein Element vom Typ *DirectoryName* enthalten, das den Anforderungen des issuer Feldes aus dem Signaturzertifikat genügt. Weitere Elemente unterliegen denselben Beschränkungen wie das *issuerAltName*-Erweiterungsfeld (siehe Abschnitt 2.3.9.6).

3.5 Signatur

Zweck

Das Signaturfeld *signature* vom Typ *AlgorithmIdentifier* enthält den Bezeichner des Algorithmus, der von der Zertifizierungsstelle zum Signieren des Attributzertifikates benutzt wird. Hierbei ist zu beachten, daß Signaturalgorithmen immer in Kombination mit Einweg-Hash-Funktionen und digitalen Signaturformaten 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. 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" beschrieben.

ASN.1-Definitionen

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

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

SigI-Konformitätsanforderungen

Es gelten dieselben Festlegungen wie sie im Abschnitt 2.3.3 beschrieben sind. Das Signaturfeld *signature* der *tbsAttributeCertificate*-Struktur muß denselben Algorithmusbezeichner wie im *signatureAlgorithm*-Feld der *AttributeCertificate*-Struktur enthalten.

3.6 Seriennummer

Zweck

Die Seriennummer ist eine natürliche Zahl, die von der Zertifizierungsstelle jedem Attributzertifikat zugewiesen wird und die dieses dadurch innerhalb der Zertifizierungsstelle eindeutig identifiziert.

ASN.1-Definitionen

```
CertificateSerialNumber     ::=  INTEGER
```

Allgemeine Konformitätsanforderungen

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

SigI-Konformitätsanforderungen

Es gelten dieselben Festlegungen wie sie im Abschnitt 2.3.2 beschrieben sind.

3.7 Gültigkeitsdauer

Zweck

Die Gültigkeitsdauer eines Attributzertifikates wird im Feld *attCertValidityPeriod* durch die zwei Zeitpunkte *notBeforeTime* und *notAfterTime* angegeben. Beide Zeitpunkte werden durch das ASN.1-Datums- und Zeitformat *GeneralizedTime* dargestellt.

SigI-Konformitätsanforderungen

Nach der Signaturverordnung [SigV 97, §7] endet die Gültigkeit eines Attributzertifikates mit der Gültigkeit des zugehörigen Signaturschlüsselzertifikates.

Bei der Kodierung der Datums- und Zeitangaben ist für *GeneralizedTime* das Format YYYYMMDDHHMMSSZ (siehe Abschnitt 2.3.5) zu beachten.

3.8 Eindeutige Bezeichner

Zweck

Das optionale *issuerUniqueID-Feld* dient zur eindeutigen Identifizierung des Erstellers eines Attributzertifikates, falls der im *issuer*-Feld angegebene Name der ausstellenden Zertifizierungsstelle hierzu nicht ausreicht.

SigI-Konformitätsanforderungen

Die Benutzung des optionalen *issuerUniqueID*-Feldes ist bei der Generierung von Attributzertifikaten verboten, da die ausstellende Zertifizierungsstelle bereits durch das *issuer*-Feld eindeutig benamt sein muß.

Es gelten dieselben Festlegungen wie sie im Abschnitt 2.3.8 beschrieben sind.

3.9 Attribute

Das Attributfeld *attributes* enthält die eigentlichen Nutzdaten eines Attributzertifikates, die syntaktisch in der Form von X.500-Verzeichnisdienstattributen aufgebaut sind.

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.

Allgemeine Konformitätsanforderungen

Die einzelnen Attribute eines Attributzertifikates müssen nicht auf Attribute beschränkt sein, die in einem X.500-Verzeichnisdienst gespeichert sind.

SigI-Konformitätsanforderungen

Im Rahmen des SigI-Profils werden die zusätzlichen Attribute *atProcuration* (Vertretungsmacht), *atAdmission* (Zulassungsinformation), *atMonetaryLimit* (monetäre Beschränkung), *atDeclarationOfMajority (*Volljährigkeitserklärung eines Teilnehmers*)* und *atRestriction* (sonstige Einschränkungen) festgelegt.

Außer dieser Möglichkeit, Attribute in Attributzertifikaten zu halten, können Attribute auch in der *subjectDirectoryAttributes*-Erweiterung eines Signaturschlüssel- oder Attributzertifikats angegeben werden (siehe Abschnitt 2.3.9.11).

SigI-konforme Zertifizierungsstellen sollen bei der Erzeugung von Zertifikaten die SigI-spezifischen Attribute unterstützen können.

SigI-konforme Systeme und Anwendungen sollen die SigI-spezifischen Attribute erkennen und verarbeiten können.

Die SigI-spezifischen Objektbezeichner für Attribute werden unter dem Objektbezeichnerzweig *id-sigi-at* festgelegt.

ASN.1-Definitionen

```
TBSAttributeCertificate ::= SEQUENCE {
    ...,
    attributes SEQUENCE 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 }
id-sig-at OBJECT IDENTIFIER ::= { 1 3 36 8 3 }
```

3.9.1 ERSTELLUNGSDATUM EINES ZERTIFIKATS

Zweck

Das SigI-spezifische Attribut *atDateOfCertGen* dient zur Anzeige des Erstellungsdatum eines Zertifikates. Es wird durch den Objektbezeichner *id-sigi-at-dateOfCertGen* referenziert.

ASN.1-Definitionen

```
atDateOfCertGen ATTRIBUTE ::=    {
WITH SYNTAX           DateOfCertGenSyntax
   SINGLE VALUE
   ID                 id-sigi-at-dateOfCertGen }
id-sigi-at OBJECT IDENTIFIER              ::= { 1 3 36 8 3 }
id-sigi-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 2.3.5 beschriebene Format YYYYMMDDHHSSZ zu beachten.

SigI-Konformitätsanforderungen

Die Benutzung des Attributes *atDateOfCertGen* durch Zertifizierungsstellen bei der Erzeugung von Zertifikaten ist optional und dabei als *non-critical* zu markieren. SigI-konforme Systeme und Anwendungen müssen dieses Attribute erkennen können.

Tabelle 36: Implementations-technische Informationen über *atDateOfCertGen*

BEZEICHNER	WERTEBEREICH EINZELWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ		
	(BEISPIELE)	(BEISPIELE)	[BYTES] 28	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional
AtDateOfCertGen	SEQUENCE { { 1 3 36 8 3 1 }, SET OF {"19980101000000Z"} }	**30** 1A **06** 05 *2B 24 08 03 01* **31** 11 **18** 0F *31 39 39 38 30 31* *30 31 30 30 30 ...*	28	v	v	v	v			v

3.9.2 VERTRETUNGSMACHT

Zweck

Das SigI-spezifische Attribut *atProcuration* dient zur Anzeige der Vertretungsmacht für eine dritte Person. Es wird durch den Objektbezeichner *id-sigi-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, nach dessen Gesetz die Vertretungsmacht zu interpretieren ist, sowie die Art der Vertretung. Durch das SEQUENCE-OF-Konstrukt kann in dem Attribut die Vertretungsmacht für mehrere dritte Personen angegeben werden.

ASN.1-Definitionen

```
atProcuration ATTRIBUTE ::= {
WITH SYNTAX             ProcurationSyntax
   SINGLE VALUE
   ID                   id-sigi-at-procuration }

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

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

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

CertificateSerialNumber ::= INTEGER

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

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

SigI-Konformitätsanforderungen

Die Benutzung des Attributes *atProcuration* durch Zertifizierungsstellen bei der Erzeugung von Zertifikaten ist optional. Die Länge des Systemparameters *maxSize* von ProcurationSyntax ist auf 128 Bytes begrenzt.

Tabelle 45: Implementations-technische Informationen über *atProcuration*

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ		
	(BEISPIELE)	(BEISPIELE)	[BYTES] 256	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional
atProcuration	SEQUENCE {	**30** 2D	47	v	v	v	v			v
type	{ 1 3 36 8 3 2 },	**06** 05 *2B 24 08 03 02*								
values	SET OF { SEQ OF {	**31** 24 **30** 22								
signingFor	[0] SEQUENCE {	**80** 20 **30** 1E								
type-id	{ 1 3 36 8 4 1 },	**06** 05 *2B 24 08 04 01*								
value	[0] EXPLICIT SEQ {	**A0** 15 **30** 13								
surAGivenN	SEQUENCE {	**30** 11								
surName	"Name",	**13** 04 *4E 61 6D 65*								
givenName	SEQ OF {"Vorname" }}}}}}	**30** 09 **13** 07 *56 6F 73 ..*								

3.9.3 ZULASSUNG

Zweck

Das SigI-spezifische Attribut *atAdmission* dient zur Anzeige von Zulassungen, wie beispielsweise einer berufsrechtlichen Zulassung.

Das SigI-spezifische, Attribut *admission* mit der Syntax *AdmissionSyntax* dient zur Anzeige von Zulassungen, wie beispielsweise einer berufsrechtlichen Zulassung. Es wird durch den Objektbezeichner *id-sigi-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 Identifzierung der verantwortlichen Namensinstanz, eine URL zur Lokalisierung der Codeliste und ein Textstring benutzt, der beispielsweise die Stelle, das Land und den Codelistennamen enthalten kann. Objektbezeichner für die Komponente *namingAuthorityId* werden unter dem Zweig id-sigi-at-namingAuthorities angeordnet und müssen beim TeleTrust-Verein beantragt werden.
- 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

```
atAdmission ATTRIBUTE ::=   {
WITH SYNTAX                 AdmissionSyntax
   SINGLE VALUE
   ID                       id-sigi-at-admission }

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

id-sigi-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.

SigI-Konformitätsanforderungen

Die Benutzung des Attributes *atAdmission* durch Zertifizierungsstellen bei der Erzeugung von Zertifikaten ist optional. Die Länge des Systemparameters *maxSize* von *AdmissionSyntax* ist auf 128 Bytes begrenzt.

Tabelle 46: Implementations-technische Informationen über *atAdmission*

BEZEICHNER	WERTEBEREICH EINZELWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ		
	(BEISPIELE)	(BEISPIELE)	[BYTES] 256	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional
AtAdmission	SEQUENCE {	**30** 29	43	v	v	v	v			v
type	{ 1 3 36 8 3 3},	**06** 05 *2B 24 08 03 03*								
values	SET OF {	**31** 20								
AdmissionSynt	SEQUENCE {	**04** 18								
ContentsOfAdm	SEQUENCE OF {	**30** 16								
Admissions	SEQUENCE {	**30** 14								
ProfessionInfos	SEQUENCE OF {	**30** 12								
ProfessionInfo	SEQUENCE {	**30** 10								
ProfessionItems	SEQ OF {"Dipl.-Phys" } } } } } } }	**30** 0E **30** 0C **13** 0A *44 69 70 6C 2E 2D 50 68 79 73*								

3.9.4 MONETÄRE BESCHRÄNKUNG

Zweck

Das SigI-spezifische Attribut *atMonetaryLimit* dient zur Anzeige einer monetären Beschränkung. Es wird durch den Objektbezeichner *id-sigi-at-monetaryLimit* referenziert. Außer den Komponenten *amount* und *exponent*, aus denen sich der Beschränkungswert gemäß $amount \cdot 10^{exponent}$ ergibt, muß die Währung im Teilfeld *currency* angegeben werden. Die Tabelle 39 im Abschnitt 2.3.9.15.6 enthält eine Übersicht der internationalen Währungen und deren zugehörige Abkürzungen.

ASN.1-Definitionen

```
atMonetaryLimit   ATTRIBUTE ::=  {
WITH SYNTAX                      MonetaryLimitSyntax
   SINGLE VALUE
   ID                            id-sigi-at-monetaryLimit }
MonetaryLimitSyntax   ::=   SEQUENCE {
   currency                      PrintableString (SIZE(3)),
   amount                        INTEGER,
   exponent                      INTEGER }
```

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

SigI-Konformitätsanforderungen

Die Benutzung des Attributs *atMonetaryLimit* durch Zertifizierungsstellen bei der Erzeugung von Zertifikaten ist optional. Die Länge der Wertefelder von *amount* und *exponent* ist auf 1 Byte begrenzt. Beispiele für zulässige Werte für *currency* können aus der Tabelle 39 entnommen werden. SigI-konforme Systeme und Anwendungen müssen das Attribut *atMonetaryLimit* erkennen und verarbeiten können.

Tabelle 47: Implementations-technische Informationen über *atMonetaryLimit*

BEZEICHNER	WERTEBEREICH EINZELWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ		
	(BEISPIELE)	(BEISPIELE)	[BYTES] 24	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional
atMonetaryLimit	SEQUENCE {	**30** 16	24	v	v	v	v			v
type	{ 1 3 36 8 3 4 },	**06** 05 *2B 24 08 03 04*								
values	SET OF { SEQUENCE {	**31** 0D **30** 0B								
currency	"DEM",	**13** 03 *44 45 4D*								
amount exponent	1, 4 } } }	**02** 01 *01* **02** 01 *04*								

3.9.5 VOLLJÄHRIGKEIT

Zweck

Das SigI-spezifische Attribut *atDeclarationOfMajority* dient zur Anzeige der Volljährigkeit eines Teilnehmers. Es wird durch den Objektbezeichner *id-sigi-at-declarationOfMajority* 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. Die dritte Variante *dateOfBirth* enthält das Geburtsdatum des Teilnehmers.

ASN.1-Definitionen

```
atDeclarationOfMajority     ATTRIBUTE ::=  {
WITH SYNTAX                 DeclarationOfMajoritySyntax
   SINGLE VALUE
   ID                       id-sigi-at-declarationOfMajority }

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

id-sigi-at-atDeclarationOfMajority OBJECT IDENTIFIER
                            ::=  { 1 3 36 8 4 5 }
```

Statische Semantik

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

SigI-Konformitätsanforderungen

Die Benutzung des SigI-spezifischen Attributs *atDeclarationOfMajority* durch Zertifizierungsstellen bei der Erzeugung von Zertifikaten ist optional. SigI-konforme Systeme und Anwendungen müssen das SigI-spezifische Attribut *atDeclarationOfMajority* erkennen und verarbeiten können.

Tabelle 48: Implementations-technische Informationen über *atDeclarationOfMajority*

BEZEICHNER	WERTEBEREICH EINZELWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ		
	(BEISPIELE)	(BEISPIELE)	[BYTES] 23	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional
AtDeclaration OfMajority notYoungerThan	SEQUENCE { { 1 3 36 8 3 5 }, SET OF { [0] IMP. 18 } }	**30** 0C **06** 05 *2B 24 08 03 05* **31** 03 **80** 01 *12*	14	v	v	v	v			v

3.9.6 SONSTIGE EINSCHRÄNKUNGEN

Zweck

Das SigI-spezifische Attribut *atRestriction* dient zur Anzeige von sonstigen Einschränkungen.

ASN.1-Definitionen

```
AtRestriction ATTRIBUTE     ::=  {
WITH SYNTAX                      RestrictionSyntax
   SINGLE VALUE
   ID                            id-sigi-at-restriction }
RestrictionSyntax       ::=  DirectoryString(SIZE(1..maxSize))
id-atRestriction OBJECT IDENTIFIER ::= { 1 3 36 8 4 8 }
```

SigI-Konformitätsanforderungen

Die Benutzung des Attributes *atRestriction* durch Zertifizierungsstellen bei der Erzeugung von Zertifikaten ist optional. Die Länge des Systemparameters *maxSize* von *RestrictionSyntax* ist auf 128 Bytes begrenzt.

Tabelle 49: Implementations-technische Informationen über *atRestriction*

BEZEICHNER	WERTEBEREICH EINZEILWERTE	HEXADEZIMALCODE	LÄNGE	ZERTIFIKATSTYP				RELEVANZ		
	(Beispiele)	(Beispiele)	[Bytes] 141	Zertifizierungsstellen	Zeitstempeldienst	Verzeichnisdienst	Teilnehmer	obligatorisch	verboten	optional

atRestriction	SEQUENCE {
	{ 1 3 36 8 4 8 },
	SET OF {
	"Sonstige Zulassungs. ..." } }
		30 29
		06 05 *2B 24 08 04 08*
		31 20
		13 14 *53 6F 6E ...*
			43	v	v	v	v			v

3.10 Erweiterungen

Das optionale *extensions-Feld* gestattet die Aufnahme weiterer Felder in Attributzertifikate. Die Definition von Erweiterungen für Attributzertifikate ist identisch mit der für Zertifikate (siehe Unterpunkt 2.3.9). Gegenwärtig existieren noch keine speziellen Erweiterungen für Attributzertifikate.