

# Public Key Service 2007

## Profil der qualifizierten Signaturzertifikate

Version: 1.1  
Stand: 21.09.2007  
Status: Freigegeben

Öffentliches Dokument



## Impressum

### Herausgeber

---

T-Systems Enterprise Services

<b>Dateiname</b>	<b>Dokumentnummer</b>	<b>Dokumentenbezeichnung</b>
PKS Zertifikatsprofil 2048 qual Sign v1.1.doc		Spezifikation

<b>Version</b>	<b>Stand</b>	<b>Status</b>
1.1	21.09.2007	Freigegeben

<b>Autor</b>	<b>Inhaltlich geprüft von</b>	<b>Freigegeben von</b>
Peter Schmidt, Volker Mann	Thomas Hoof	Jörg Spitzensteder

<b>Ansprechpartner</b>	<b>Telefon / Fax</b>	<b>E-Mail</b>
[Hier Name eingeben]	[Hier Telefonnummer eingeben]	[Hier E-Mail eingeben]

### Kurzinfo

---

Zertifikatsprofil der qualifizierten PKS Signatur-Zertifikate

Copyright © 2007 by T-Systems Enterprise Services, Frankfurt (Main)

Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

## Änderungshistorie

Version	Stand	Bearbeiter	Änderungen / Kommentar
0.1	21.05.2007	PS	Initialer Draft
0.2	24.05.2007	PS, TH	Nachbesserungen und QS
1.0	27.07.2007	TH, JS	Korrekturen und Freigabe
1.1	21.09.2007	PS	Beschränkung der Länge für Restriction auf 100 Zeichen; Ergänzung der CPS-URL in policyQualifierInfo

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>5</b>
<b>2</b>	<b>Profil der Schlüsselzertifikate</b>	<b>6</b>
2.1	Übersicht über die Felder des Schlüsselzertifikates.....	7
2.2	Das Feld Version.....	9
2.3	Das Feld SerialNumber.....	9
2.4	Das Feld Signature.....	9
2.5	Das Feld Issuer.....	11
2.6	Das Feld Validity.....	12
2.7	Das Feld Subject.....	12
2.8	Das Feld SubjectPublicKeyInfo.....	14
2.9	Das Feld SignatureAlgorithm.....	14
2.10	Das Feld SignatureValue.....	14
2.11	Das Feld Extensions.....	15
2.11.1	Die Extension AuthorityKeyIdentifier (vorgeschrieben).....	15
2.11.2	Die Extension SubjectKeyIdentifier (vorgeschrieben).....	16
2.11.3	Die Extension KeyUsage (vorgeschrieben).....	17
2.11.4	Die Extension CertificatePolicies (vorgeschrieben).....	17
2.11.5	Die Extension SubjectAltNames (optional).....	18
2.11.6	Die Extension CRLDistributionPoints (vorgeschrieben).....	18
2.11.7	Die Extension AuthorityInfoAccess (vorgeschrieben).....	19
2.11.8	Die Extension QCStatements (vorgeschrieben).....	20
2.11.9	Die Extension LiabilityLimitationFlag (bedingt vorgeschrieben).....	21
2.11.10	Die Extension Procuration (optional).....	21
2.11.11	Die Extension Admission (optional).....	22
2.11.12	Die Extension Restriction (optional).....	23

# 1 Einleitung

Die qualifizierten Zertifikate mit Anbieterakkreditierung des Public Key Service sind konform zur ISIS-MTT Spezifikation [ISIS-MTT]. Damit ist eine Interoperabilität zu anderen Zertifizierungsstellen gegeben, die diesen Standard unterstützen, und die Zertifikate können von Standard-Clients verarbeitet werden.

Die ISIS-MTT-Spezifikation [ISIS-MTT] ist eine Profilierung der internationalen PKIX-Standards ist.

Dieses Dokument beschreibt den Aufbau von qualifizierten Signaturzertifikaten. Die in dem jeweiligen Zertifikattyp enthaltenen Daten werden ausgehend von der jeweils grundlegenden Struktur gemäß X.509 [X509] erläutert. Dabei werden die Felder und mögliche Inhalte, sowie die zu verwendenden Datentypen festgelegt.

## 2 Profil der Schlüsselzertifikate

Der generelle Aufbau des Schlüsselzertifikats entspricht der Norm X.509 und hat folgende Struktur:

```
Certificate ::= SEQUENCE
{
    toBeSigned      SEQUENCE {
        version      [0] Version DEFAULT v1
        serialNumber  INTEGER
        signature     AlgorithmIdentifier
        issuer        Name
        validity      Validity
        subject       Name
        subjectPublicKeyInfo  SubjectPublicKeyInfo
        issuerUniqueID  [1] IMPLICIT UniqueIdentifier
                                                                Optional
        subjectUniqueID  [2] IMPLICIT UniqueIdentifier
                                                                Optional
        extensions    [3] Extensions      Optional
    }
    signatureAlgorithm  AlgorithmIdentifier
    signatureValue      BIT STRING
}
```

Die Datentypen der verwendeten Felder des Zertifikates und deren mögliche Inhalte werden in den folgenden Kapiteln erläutert.

## 2.1 Übersicht über die Felder des Schlüsselzertifikates

Die folgende Tabelle zeigt alle möglichen Felder innerhalb des Zertifikates, die von TeleSec genutzt werden. Nicht verwendete Felder werden in der Tabelle nicht aufgeführt. Die Extensions werden zusätzlich mit P (Pflicht, immer vorhanden), B (bedingt Pflicht) oder O (optional) gekennzeichnet.

Name des Feldes	Bedeutung		Wert
Version	Versionsnummer für die verwendete Zertifikatsstruktur		2 (Version 3)
SerialNumber	Eindeutige Zertifikatsnummer innerhalb der CA		z.B. 1234
Signature	Algorithmus der Signatur des Zertifikates		sha-256WithRsaEncryption
Issuer	Name des Herausgebers des Zertifikates		z.B. C=DE / O=Deutsche Telekom AG / OU=T-TeleSec / CN=TeleSec SigG CA 1:PN
Validity	Gültigkeitszeitraum des Zertifikates		z.B. vom 02.04.2002 09:20:23 bis zum 02.04.2005 09:20:23
Subject	Name des Inhabers des Zertifikates		z.B. C=DE / CN=Herbert Mustermann / SER=1
SubjectPublicKeyInfo	Öffentlicher Schlüssel des Zertifikatsinhabers mit dem zugehörigen Algorithmus		rsaEncryption-Schlüssel (2048 bit)
SignatureAlgorithm	Algorithmus, mit dem das Zertifikat signiert worden ist		sha-256WithRsaEncryption
SignatureValue	Signatur des Zertifikates		BIT STRING
<b>Extensions</b>			
AuthorityKeyIdentifier	P	Informationen zur Identifizierung des öffentlichen Schlüssels des Ausstellerzertifikates	keyIdentifier = OCTET STRING mit einer Länge von 20 Bytes
SubjectKeyIdentifier	P	Informationen zur Identifizierung des öffentlichen Schlüssels des Zertifikates	OCTET STRING mit einer Länge von 20 Bytes

KeyUsage	P	Information, für welchen Zweck das Zertifikat benutzt werden darf	nur das Bit nonRepudiation ist gesetzt
CertificatePolicy	P	Information, unter welchen Bedingungen das Zertifikat erstellt wurde und unter welchen Bedingungen es genutzt werden darf, Information, wo das Certification Practice Statement zu finden ist	Wert isismtt-cp-sigconform und die URL der CPS
SubjectAltName	O	Zusätzlicher technischer Name für den Inhaber des Zertifikates.	z.B. E-Mail-Adresse = <a href="mailto:Herbert.Mustermann@xyz.de">Herbert.Mustermann@xyz.de</a>
CrlDistributionPoints	P	Liefert Informationen, wie Sperrinformationen (CRL) zu diesem Zertifikat bezogen werden können	z.B. ldap://ldap.telesec.de:389/o=Deutsche Telekom,c=de http: <a href="http://www.telesec.de/crl">http://www.telesec.de/crl</a>
AuthorityInfoAccess	P	Liefert die Adresse (URL) des OCSP-Servers für die Prüfung von Zertifikaten.	z.B. <a href="http://www.telesec.de/ocsp">http://www.telesec.de/ocsp</a>
QCStatement	P	Indikator dafür, dass es sich um ein qualifiziertes Zertifikat handelt. Eine Selbstbeschränkung (MonetaryLimit) kann mit diesem Feld vorgenommen werden.	ID = id_etsi_qcs_qcCompliance  z.B. zusätzliche Einschränkung: ID = id-etsi-qcs-QcLimitValue Wert: maximal 1000 Euro
LiabilityLimitation	B	Anzeige, dass es ein Attributzertifikat mit Einschränkung gibt.	z.B. Attributzertifikat mit Beschränkung vorhanden
Procuration	O	Informationen über Vertretungsvollmacht	Der Inhaber des Zertifikates darf für eine andere Person Unter-

			schriften leisten.
Admission	O	Informationen über die Berechtigung zur Ausübung von Tätigkeiten	Der Inhaber des Zertifikats ist ein anerkannter Steuerberater.
Restriction	O	Zusätzliche sonstige Einschränkungen	z.B. Der Inhaber des Zertifikats darf nur blaue Autos kaufen.
AdditionalInformation	O	Zusätzliche Informationen, die nicht in anderen Strukturen abgebildet werden können.	z.B. juristischer Text

## 2.2 Das Feld Version

In dem Feld `Version` wird die Version der Datenstruktur des Zertifikates kodiert. Alle Zertifikate müssen die Versionsnummer `v3` haben, da nur in dieser Version Extensions enthalten sein dürfen. Der Datentyp von `Version` ist `INTEGER` und hat immer den Wert `2` (entspricht `v3`).

## 2.3 Das Feld SerialNumber

In diesem Feld wird die Zertifikatsnummer abgelegt. Diese Zertifikatsnummer muss innerhalb des Zertifizierungsbereiches (gleicher Issuer) eindeutig sein. Der Datentyp der Zertifikatsnummer ist `INTEGER` und muss immer positiv sein. Die maximale Länge des kodierten Wertes darf 20 Bytes ( $1 \leq \text{Zertifikatsnummer} < 2^{159}$ ) nicht überschreiten.

## 2.4 Das Feld Signature

Das Feld `Signature` enthält den Bezeichner des Signaturalgorithmus, der von der CA für die Erstellung des Zertifikates benutzt wird. Der Inhalt dieses Feldes hat folgende Struktur:

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

Folgender Algorithmus wird verwendet:

- sha-256WithRsaEncryption {1.2.840.113549.1.1.11}

Das Feld `parameters` kann zusätzliche Parameter des Algorithmus enthalten. Für den verwendeten Algorithmus (RSA) werden sie jedoch nicht benötigt, deshalb ist dieses Feld Null (explizit NULL).

## 2.5 Das Feld Issuer

In dem Feld `Issuer` wird der Name des Herausgebers abgelegt. Der Inhalt dieses Feldes muss exakt mit dem Inhalt des Subject-Feldes des CA-Zertifikates übereinstimmen, von dem das Benutzerzertifikat unterschrieben worden ist.

Folgende Attribute werden verwendet:

- `countryName` (vorgeschrieben)
- `organizationName` (vorgeschrieben)
- `organizationalUnitName` (optional)
- `commonName` (optional)
- `pseudonym` (bedingt, muss verwendet werden, wenn sich im `commonName` ein `pseudonym` befindet).

Das Feld hat die folgende Datenstruktur:

```
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
```

Anmerkung:

Der Inhalt des Feldes `Issuer` wird aus dem Subject-Feld des CA-Zertifikates entnommen von dem das Zertifikat später unterschrieben werden soll.

## 2.6 Das Feld Validity

In diesem Feld wird der Gültigkeitszeitraum des Zertifikates eingetragen. Der Inhalt des Feldes hat folgende Syntax:

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

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

Die Uhrzeit wird bis zum Jahr 2049 als UTCTime kodiert werden. Ab dem Jahr 2050 wird die Kodierung GeneralizedTime verwendet.

## 2.7 Das Feld Subject

Der Name des Zertifikatsinhabers wird in dem Feld `Subject` gespeichert. Der Name für den Zertifikatsinhaber muss innerhalb der Zertifizierungsstelle für die komplette Lebensdauer der CA eindeutig sein.

Folgende Attribute werden verwendet:

- `countryName` (vorgeschrieben)
- `organizationName` (optional)
- `organizationalUnitName` (optional)
- `commonName` (vorgeschrieben)
- `serialNumber` (vorgeschrieben)
- `pseudonym` (bedingt vorgeschrieben s.u.)

Wenn der Zertifikatsinhaber ein Pseudonym als Name wünscht, wird zusätzlich das Attribut `Pseudonym` kodiert. Der Pseudonym-Name befindet sich immer in den Attri-

buten commonName und pseudonym. Hierbei wird ein Pseudonym mit der Endung „:PN“ gekennzeichnet.

Das Feld hat die gleiche Datenstruktur wie das Feld Issuer (siehe Kapitel 2.5).

## 2.8 Das Feld SubjectPublicKeyInfo

Der PublicKey des Zertifikatsinhabers wird in diesem Feld zusammen mit dem Algorithmus für den Gebrauch des Schlüssels gespeichert.

Als Algorithmus wird rsaEncryption {1 2 840 113549 1 1 1} verwendet.

Der Inhalt des Feldes hat folgende Syntax:

```
SubjectPublicKeyInfo ::= SEQUENCE
{
    algorithm      AlgorithmIdentifier
    subjectPublicKey BIT STRING
}
AlgorithmIdentifier ::= SEQUENCE
{
    algorithm      OBJECT IDENTIFIER
    parameters    ANY DEFINED BY algorithm OPTIONAL
}
```

Das Feld `parameters` kann zusätzliche Parameter des Algorithmus enthalten. Für den verwendeten Algorithmus (RSA) werden sie jedoch nicht benötigt, deshalb ist dieses Feld Null (explizit NULL).

## 2.9 Das Feld SignatureAlgorithm

Dieses Feld enthält den Algorithmus, mit dem das Zertifikat von der Zertifizierungsstelle unterschrieben worden ist. Der Inhalt und die Kodierung müssen identisch mit dem Feld `Signature` (siehe Kapitel 2.4) sein.

## 2.10 Das Feld SignatureValue

Das Feld `signatureValue` enthält die Signatur des Zertifikates, die von der Zertifizierungsstelle erzeugt worden ist.

Der Typ des Feldes `signatureValue` ist BIT STRING.

## 2.11 Das Feld Extensions

Die Extensions dienen zur Erweiterung der im Zertifikat enthaltenen Daten. Es gibt mehrere verschiedene Extensions, die in den folgenden Unterkapiteln aufgeführt werden.

Die Extensions haben folgende Syntax:

```
Extensions ::= SEQUENCE SIZE (1..MAX) OF Extension
Extension ::= SEQUENCE
{
  extnId      OBJECT IDENTIFIER
  critical    BOOLEAN DEFAULT FALSE
  extnValue   OCTET STRING
}
```

Der Wert `extnId` gibt mit Hilfe eines Object Identifiers den Typ der in `extnValue` enthaltenen Extension an. Das Flag `critical` zeigt an, ob die Extension als kritisch markiert worden ist. Wenn `critical` auf TRUE gesetzt wird, bedeutet dies, dass der Client das Zertifikat als ungültig betrachten muss, wenn er die Extension nicht auswerten kann.

### 2.11.1 Die Extension AuthorityKeyIdentifier (vorgeschrieben)

Diese Extension dient zur eindeutigen Identifizierung des Ausstellerzertifikates, mit dem das Benutzerzertifikat unterschrieben worden ist.

Diese Extension wird als nicht kritisch markiert.

Die Extension AuthorityKeyIdentifier hat folgende Datenstruktur:

```
AuthorityKeyIdentifier ::= SEQUENCE
{
  keyIdentifier          [0] IMPLICIT KeyIdentifier OPTIONAL
  authorityCertIssuer    [1] IMPLICIT GeneralNames OPTIONAL
  authorityCertSerialNumber [2] IMPLICIT CertificateSerialNumber
                                                                    OPTIONAL
                                                                    NAL
}
```

Das Feld `keyIdentifier` enthält einen eindeutigen Wert zur Identifizierung des öffentlichen Schlüssels des Herausgebenden CA-Zertifikats. Der Wert enthält den mit dem Algorithmus SHA-1 (160 bit) berechneten Hashwert über den `subjectPublicKey` des CA-Zertifikates (ohne die Bytes Tag, Länge und nicht benutzte Bits).

Die Felder `authorityCertIssuer` und `authorityCertSerialNumber` werden nicht verwendet.

### 2.11.2 Die Extension `SubjectKeyIdentifier` (vorgeschrieben)

Diese Extension dient zur eindeutigen Identifizierung des im Zertifikat enthaltenen öffentlichen Schlüssels.

Diese Extension wird als nicht kritisch markiert.

Die Extension `SubjectKeyIdentifier` hat folgende Datenstruktur:

```
SubjectKeyIdentifier ::= KeyIdentifier
```

```
KeyIdentifier ::= OCTET STRING
```

Der Wert der Extension ist der mit dem Algorithmus SHA-1 (160 bit) berechnete Hashwert über den `subjectPublicKey` (ohne die Bytes Tag, Länge und nicht benutzte Bits).

### 2.11.3 Die Extension KeyUsage (vorgeschrieben)

Mit dieser Extension wird festgelegt, für welchen Verwendungszweck der zum Zertifikat gehörende private Schlüssel benutzt werden darf.

Diese Extension wird als kritisch markiert.

Die Extension hat folgenden Aufbau:

```
KeyUsage ::= BIT STRING
{
  digitalSignature (0),
  nonRepudiation (1),
  keyEncipherment (2),
  dataEncipherment (3),
  keyAgreement (4),
  keyCertSign (5),
  crlSign (6),
  encipherOnly (7),
  decipherOnly (8)
}
```

Für qualifizierte Signaturzertifikate wird nur das Bit `nonRepudiation` gesetzt.

### 2.11.4 Die Extension CertificatePolicies (vorgeschrieben)

Diese Extension legt die Bedingungen fest, unter denen das Zertifikat herausgegeben wurde und unter denen es verwendet werden darf.

Diese Extension wird als nicht kritisch markiert.

```
CertificatePolicies ::= 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
}
PolicyQualifierId ::= OBJECT IDENTIFIER {id-qt-cps | id-qt-
unotice}

```

Als `policyIdentifier` wird immer `id-isismtt-cp-sigconform {1 3 36 8 1 1}` verwendet. Die zusätzliche `PolicyQualifierInfo` enthält die URL wo die CPS zum Abruf bereitgehalten wird.

### 2.11.5 Die Extension `SubjectAltNames` (optional)

Mit Hilfe dieser Extension können alternative (technische) Namen für den Zertifikatsinhaber im Zertifikat eingefügt werden.

Diese Extension wird als nicht kritisch markiert.

Die Extension hat folgende Datenstruktur:

```
SubjectAltNames ::= GeneralNames
```

Wenn diese Extension vorhanden ist enthält sie genau eine E-Mail-Adresse des Zertifikatsinhabers als `rfc822Name`.

### 2.11.6 Die Extension `CRLDistributionPoints` (vorgeschrieben)

Diese Extension liefert Informationen, wie Sperrinformationen zu dem Zertifikat bezogen werden können.

Diese Extension wird als nicht kritisch markiert.

Die Extension hat folgenden Aufbau:

```

CrlDistributionPoints ::= SEQUENCE SIZE (1..MAX) OF
                                CrlDistributionPoint
CrlDistributionPoint ::= SEQUENCE
{
    distributionPoint [0] EXPLICIT DistributionPointName
                                OPTIONAL

```

```

reasons          [1] IMPLICIT ReasonFlags OPTIONAL
cRLIssuer        [2] IMPLICIT GeneralNames OPTIONAL
}
DistributionPointName ::= CHOICE
{
  fullName          [0] IMPLICIT GeneralNames
  nameRelativeToCRLIssuer [1] IMPLICIT
                                     RelativeDistinguishedName
}

```

Das Feld `fullName` enthält die LDAP-URL einschließlich des DNAMES der CRL. Zusätzlich wird die optionale HTTP-URL verwendet. Das Feld `nameRelativeToCRLIssuer` wird nicht verwendet.

Der `cRLIssuer` wird verwendet, da er sich immer von dem Herausgeber der Zertifikate unterscheidet. Das Feld enthält die Attribute `CountryName`, `OrganizationName` und `OrganizationalUnitName` aus dem Subject-Feld des Directory-Zertifikates, von dem die Sperrliste unterschrieben worden ist.

Das Feld `reasons` wird nicht verwendet.

### 2.11.7 Die Extension `AuthorityInfoAccess` (vorgeschrieben)

Diese Extension enthält eine URL, unter der ein OCSP-Server für die Zertifikatsprüfung angesprochen werden kann.

Diese Extension wird als nicht kritisch markiert.

Die Extension `AuthorityInfoAccess` hat folgenden Aufbau:

```

AuthorityInfoAccessSyntax ::=
    SEQUENCE (1..MAX) OF AccessDescription

AccessDescription ::= SEQUENCE
{
    accessMethod    OBJECT IDENTIFIER
    accessLocation  GeneralName
}

```

Als Zugriffsmethode wird im Feld `accessMethod` der Wert `id-ad-ocsp` verwendet. Der Zugriffsort wird über eine URL im Feld `accessLocation` beschrieben.

### 2.11.8 Die Extension QCStatements (vorgeschrieben)

Mit dieser Extension wird festgelegt, dass es sich um ein qualifiziertes Zertifikat handelt.

Diese Extension wird als nicht kritisch markiert.

Die Extension QCStatements hat folgenden Aufbau:

```
QCStatements ::= SEQUENCE OF QCStatement
QCStatement ::= SEQUENCE
{
    statementId      ObjectIdentifier
    statementInfo    ANY DEFINED BY statementId OPTIONAL
}
```

Als `statementId` wird der folgende Wert verwendet:

- `id-etsi-qcs-QcCompliance` zeigt an, dass es sich um ein qualifiziertes Zertifikat handelt, dessen `CertificatePolicy` konform zu ETSI TS 101 456 v1.1.1 ist und muss vorhanden sein.

Zusätzlich dazu kann der nachstehende Wert zur Einschränkung vom finanziellen Verfügungsrahmen des Zertifikatsinhabers verwendet werden:

- `id-etsi-qcs-QcLimitValue`.

```
QcEuLimitValue ::= MonetaryValue
MonetaryValue ::= SEQUENCE
{
    currency      Iso4217CurrencyCode
    amount        INTEGER
    exponent      INTEGER
}
```

```
Iso4217CurrencyCode ::= CHOICE
{
    PrintableString (SIZE 3), -- Recommended
    INTEGER (1..999)
}
```

Als Datentyp für die Kodierung des Feldes `currency` wird entsprechend der Empfehlung `PrintableString` verwendet.

Der Betrag für die Limitierung ergibt sich wie folgt: Betrag = amount \* 10<sup>exponent</sup>.

### 2.11.9 Die Extension LiabilityLimitationFlag (bedingt vorgeschrieben)

Mit dieser Extension wird angezeigt, dass ein Attributzertifikat existiert, durch das die Verwendung dieses Zertifikats eingeschränkt wird.

Die Extension wird als nicht kritisch markiert.

Die Extension hat den folgenden Aufbau:

```
LiabilityLimitationFlag ::= BOOLEAN
```

Der Wert des Flags wird immer auf TRUE gesetzt. Wenn keine Einschränkung vorhanden ist, wird die Extension nicht benutzt.

### 2.11.10 Die Extension Procuration (optional)

Diese Extension wird verwendet, wenn der Zertifikatsinhaber für eine andere Person Unterschriften leisten darf.

Diese Extension wird als nicht kritisch markiert.

Die Extension hat den folgenden Aufbau:

```
ProcurationSyntax ::= SEQUENCE
{
  country          [1] EXPLICIT PrintableString OPTIONAL
  typeOfSubstitution[2] EXPLICIT DirectoryString OPTIONAL
  signingFor       [3] EXPLICIT SigningFor
}
```

```
SigningFor ::= CHOICE
{
  thirdPerson  GeneralName
  certRef      IssuerSerial
}
```

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

Wenn das Feld `thirdPerson` in der Sequence `SigningFor` vorhanden ist, kann es neben den Attributen aus Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.** zusätzlich `OrganizationName` und /oder `OrganizationalUnitName` enthalten.

Die Auswahl `IssuerSerial` enthält im Feld `issuer` den kompletten Namen (`DirectoryName`) des Herausgebers des Zertifikates, für das Prokura erlaubt werden soll. Zusätzlich wird in das Feld `serial` die Zertifikatsnummer des Zertifikates eingetragen, für das eine Unterschrift geleistet werden darf. Das Feld `issuerUID` wird nicht genutzt.

Die Länge des Feldes `typeOfSubstitution` wird auf eine Länge von 128 Bytes begrenzt. Der Datentyp für dieses Feld ist `PrintableString`.

### 2.11.11 Die Extension Admission (optional)

Diese Extension Admission wird verwendet, um Informationen über die Berechtigung bestimmte Aufgaben erledigen zu dürfen in das Zertifikat aufzunehmen.

Diese Extension wird als nicht kritisch markiert.

Die Extension Admission hat die folgende Datenstruktur:

```
AdmissionSyntax ::= SEQUENCE
{
  admissionAuthority  GeneralName OPTIONAL
  contentsOfAdmissions  SEQUENCE OF Admissions
}
Admissions ::= SEQUENCE
{
  admissionAuthority[0] EXPLICIT GeneralName OPTIONAL
  namingAuthority  [1] EXPLICIT] NamingAuthority OPTIONAL
  professionInfos  SEQUENCE OF ProfessionInfos
}

NamingAuthority ::= SEQUENCE
{
  namingAuthorityId  OBJECT IDENTIFIER OPTIONAL
  namingAuthorityUrl  IA5String OPTIONAL
  namingAuthorityText  DirectoryString OPTIONAL
}

ProfessionInfo ::= SEQUENCE
{
  namingAuthority  [0] EXPLICIT NamingAuthority OPTIONAL
```

```
professionItems    SEQUENCE OF DirectoryString
professionOIDs     SEQUENCE OF OBJECT IDENTIFIER Optional
registrationNumberPrintableString OPTIONAL
addProfessionInfo OCTET STRING OPTIONAL
}
```

Das Feld `admissionAuthority` wird immer nur an einer Stelle kodiert. Wenn alle vorhandenen Bestätigungen von einer Instanz bestätigt wurden, wird das Feld in der Sequence `AdmissionSyntax` genutzt. Andernfalls muss das Feld in der Sequence `Admissions` verwendet werden. Der Inhalt des Feldes ist ein `DirectoryName`, der sich aus den Attributen aus Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.** und den zusätzlichen Attributen `OrganizationName` und `OrganizationalUnitName` zusammensetzt.

In dem Feld `namingAuthority` befinden sich Informationen über den Aussteller der Bestätigungen. Wenn alle Bestätigungen von einer Instanz ausgestellt worden sind, wird das entsprechende Feld in der Sequence `Admissions` genutzt. Andernfalls muss das Feld in der Sequence `ProfessionInfos` benutzt werden. Wenn in dem Feld `namingAuthorityText` Informationen vorhanden sind, müssen diese als UTF8-String kodiert werden.

Für den Inhalt des Feldes `professionItems` wird die Kodierung `PrintableString` benutzt.

### 2.11.12 Die Extension Restriction (optional)

Diese Extension kann für den Einbau von Einschränkungen in das Zertifikat aufgenommen werden.

Diese Extension wird als nicht kritisch markiert.

Die Extension Restriction hat die folgende Datenstruktur:

```
RestrictionSyntax ::= DirectoryString
```

Die Länge des Strings wird auf 100 Zeichen beschränkt. Der Inhalt der Extension `Restriction` wird als `PrintableString` kodiert.

## Anhang A: Verwendete Attributtypen

Name des Attributs	Object Identifier	ASN.1 String Typ	maximale Länge
commonName	{id-at 3}	UTF8	64
surName	{id-at 4}	UTF8	64
givenName	{id-at 42}	UTF8	64
serialNumber	{id-at 5}	PrintableString	64
title	{id-at 12}	UTF8	64
organizationName	{id-at 10}	UTF8	64
organizationalUnit-Name	{id-at 11}	UTF8	64
BusinessCategory	{id-at 15}	UTF8	128
localityName	{id-at 7}	UTF8	128
stateOrProvinceName	{id-at 8}	UTF8	128
countryName	{id-at 6}	PrintableString	2 (ISO 3166 Code)
distinguishedName-Qualifier	{id-at 46}	PrintableString	64
initials	{id-at 43}	UTF8	64
generationQualifier	{id-at 44}	UTF8	64
eMailAddress	{pkcs-9 1}	IA5String	128
domainComponent	{0 9 2342 19200300 100 1 25}	IA5String	definiert in RFC 2247
postalAddress	{id-at 16}	SEQUENCE SIZE (1..6) OF UTF8	6 * 30, Verwendung wird in RFC 3039 beschrieben
pseudonym	{pkix 9 3}	UTF8	64
dateOfBirth	{id-pda 1}	GeneralizedTime	15
placeOfBirth	{id-pda 2}	UTF8	128
gender	{id-pda 3}	PrintableString SIZE (1)	Inhalt: „M“ oder „F“
countryOfCitizenship	{id-pda 4}	PrintableString	2 (ISO 3166 Code)
countryOfResidence	{id-pda 5}	PrintableString	2 (ISO 3166 Code)
nameAtBirth	{id-isismtt-at 14}	UTF8	64

Für die UTF8-Kodierung wird einen Auszug aus dem UTF8-Zeichensatz verwendet, der nur ANSI / ISO 8859-1 Zeichen (Unicode Latin-1 Seite) enthält. Andere Zeichen, die nicht in diesem Zeichensatz enthalten sind, dürfen nicht benutzt werden.

Besonderheiten für bestimmte Attribute:

- commonName: Ein Pseudonym wird immer mit der Endung „:PN“ als Common-Name eingefügt werden. Zusätzlich wird der gleiche Inhalt (einschließlich der Endung) in dem Attribut pseudonym eingefügt.

- postalAddress: Die Adresse wird komplett in diesem Attribut gespeichert. z.B.
  1. Element: Turmstraße 123
  2. Element: 10123 Berlin
  3. Element: Germany.
- DateOfBirth: Das Geburtsdatum wird im GeneralizedTime-Format kodiert. Der Teil mit der Angabe der Uhrzeit wird mit „0“ gefüllt (z.B. 19720508000000Z).



## Abkürzungsverzeichnis

CA	Certification Authority
HTTP	Hypertext Transmission Protocol
LDAP	Lightweight Directory Access Protocol
OCSP	Online Certificate Status Protocol
PN	Pseudonym

## Literaturverzeichnis

- [ISIS-MTT] ISIS-MTT Core Specification and ISIS-MTT Optional Profiles  
Version 1.1, 16. March 2004
- [X509] Recommendation X.509: The Directory – Authentication  
Frame