# Zertifikatsprofile für X.509 Basiszertifikate, Version 2.3.2

## Bundesärztekammer, Berlin

| | Datum | Name, Abteilung, Firma |
|---|---|---|
| Autor, Ansprechpartner | | Georgios Raptis |
| Status (HPC-Projektbüro) | 12.05.11 | Freigegeben |

| Versionshistorie | | | | |
|---|---|---|---|---|
| **Version** | **Datum** | **Bearbeiter** | **Änderungen** | **Bemerkungen** |
| 0.1 | 27.05.05 | Georgios Raptis (RS) | | Initiale Version |
| 0.4 | 28.06.05 | Georgios Raptis (RS) | Hinzufügen: Beschreibung, gültige Werte, bis auf Extensions | |
| 0.5 | 01.07.05 | Georgios Raptis | Extensions | nicht abschließend |
| 0.8 | 05.07.05 | Georgios Raptis | Dokument vollständig | bereit für QS. Literaturteil noch unvollständig |
| 0.81 | 06.07.05 | Stefanie Wiegand | 1. QS, Syntax | |
| 0.82 | 19.07.05 | Georgios Raptis | Attributzertifikate möglich, keyUsage für AUTH-certs, SHA-1 | 2. QS. Änderungen: Tab. 2,<br><br>S. 14, Empfehlung laut BSI, SHA-1 für neue Produkte nicht mehr zu verwenden |
| 0.83 | 20.07.05 | Georgios Raptis | Kleinere redakt. Änderungen | |

| | | | | |
|---|---|---|---|---|
| 0.84 (intern) | 24.10.05 | Georgios Raptis | Admission für Signatur in Attributzertifikaten | + Tippfehlerkorrekturen Klarstellung Bezeichner für admissionAuthority |
| 0.85 | 07.03.06 | Georgios Raptis | basicConstraints in Enduser-Zertifikaten, SubjectDN für nicht SigG-CRL-Signer kann recycled werden, neue Attribute im SubjectDN als multivaluedRDN | + Tippfehlerkorrekturen Klarstellung Bezeichner für admissionAuthority |
| 0.86 | 07.03.06 | Dirk Schladweiler | QS | |
| 0.87 | 14.03.06<br><br>23.03.06 | Raptis | Korrektur CRL-DP, neue SubjectDNs für CA-Zertifikate<br><br>Zertifikate für Kammerangehörige definiert. Nummernräume für das SubjectDN-Attribut serialNumber | URL muss URL und UTF-8-kodiert sein, s. Tabelle 2. CA-Zertifikate mit O=ZDA |
| 0.8.8 | 31.08.06 | Dirk Schladweiler | -Überarbeitung hinsichtlich Auditergebnnissen<br><br>-Einarbeitung der Kommentare aus D-TRUST Mail vom 03.08.06 | QS erforderlich |
| 0.8.9 | 19.04.07 | Georgios Raptis<br><br>Dirk Schladweiler (QS) | Aufnahme Telematik-ID in die Admission | Editorische Korrekturen, E-Mail aus qSIG streichen, keyEncipherment obligatorisch für EE-AUTH, CRL-DP Erläuterungen nach Klärung mit ISIS-MTT |
| 0.8.10 | 13.08.07 | Dirk Schladweiler | Klarstellung zur Verwendung sprechender Namen für den Kammercode | |

| | | | | |
|---|---|---|---|---|
| 0.9.0 | 27.08.08 | Georgios Raptis<br>Stefanie Wiegand | Neue OIDs<br>Literaturverzeichnis | |
| 0.9.1 | 08.10.08 | Georgios Raptis | emailProtection in AUT | |
| 0.9.2 | 10.10.08 | Dirk Schladweiler | QS | |
| 2.3.1 | 10.03.09 | Georgios Raptis | Konsolidierung zum Paket V2.3.1<br>Admission kann im Signaturzertifikat aufgenommen werden | Flexibilisierung Algorithmen, RFC5280, Common-PKI V2.0, kein Restriction in ENC und AUT |
| 2.3.1 | 30.03.09<br>29.05.09 | Jessica Beyer | QS | |
| 2.3.2 | 12.05.11 | Dirk Schladweiler | Aktualisierung der Referenzen | |

| **Fertigstellungszustand** | | | | |
|---|---|---|---|---|
| **Lfd Nr.** | **Probleme / Offene Punkte / Defizite** | **Ursachen** | **Maßnahmen / Lösungen** | **Kapitel-verweis** |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# Inhalt

# Tabellenverzeichnis

# 1 Zertifikatsprofile für X.509 Basiszertifikate, Übersicht und Inhalte

| **Zertifikatstyp** | **Root** | **CA** | **CROSS** | **End-Entity qSIG** | **End-Entity AUTH** | **End-Entity ENC** | **CRL-Signer[1]** | **OCSP-Signer[2]** | **qCA (nicht normativ)** |
|---|---|---|---|---|---|---|---|---|---|
| Aussteller | BÄK Root | BÄK Root | BÄK Root | ZDA-qCA für Ärzte | ZDA-CA für Ärzte | ZDA-CA für Ärzte | BÄK Root | BÄK Root | BNetzA-Root |
| tbsCertificate | | | | | | | | | |
| Version | 2 (X.509v3) | | | | | | | | |
| SerialNumber | Zertifikatsseriennummer | | | | | | | | |
| Signature | OID des verwendeten Signaturalgorithmus | | | | | | | | |
| Issuer | CN=1R BÄK CA 1:PN, O=Bundesärztekammer, C=DE | CN=1R BÄK CA 1:PN, O=Bundesärztekammer, C=DE | CN=1R BÄK CA 1:PN, O=Bundesärztekammer, C=DE | CN=ZDA qCA für Ärzte 1:PN, O=ZDA, C=DE | CN=ZDA CA für Ärzte 1:PN, O=ZDA, C=DE | CN=ZDA CA für Ärzte 1:PN, O=ZDA, C=DE | CN=1R BÄK CA 1:PN, O=Bundesärztekammer, C=DE | CN=1R BÄK CA 1:PN, O=Bundesärzte-kammer, C=DE | CN=Root XY:PN, O=Bundesnetzagentur, C=DE |
| Validity | gemäß Gültig-keitsmodell | gemäß Gültig-keitsmodell | gemäß Gültig-keitsmodell | max. 5 Jahre | max. 5 Jahre | max. 5 Jahre | max. 5 Jahre | max. 5 Jahre | max. 5 Jahre |
| Subject | CN=1R BÄK CA 1:PN, O=Bundesärztekammer, C=DE | CN=ZDA CA für Ärzte 1:PN, O=ZDA, C=DE | CN=ZDA qCA für Ärzte 1:PN, O=ZDA, C=DE | CN=[Vollst. Name (:PN)[3]+ GN=[Vornamen]+SN=[Nachname]+SerNr=[int], C=DE | CN=[Vollst. Name (:PN)] + GN=[Vornamen]+SN=[Nachname]+ SerNr=[int[4]], C=DE | CN=[Vollst. Name (:PN)] + GN=[Vornamen]+SN=[Nachname]+ SerNr=[int], C=DE | CN=ZDA CRL Signer 1:PN, O=ZDA, C=DE | CN=ZDA OCSP Signer 1:PN, O=ZDA, C=DE | CN=ZDA qCA für Ärzte 1:PN, O=ZDA, C=DE |

[1] Hier beschriebene OCSP/CRL-Signer beziehen sich ausschließlich auf den nicht-qualifizierten Bereich der ENC/AUTH-Zertifikate.

[2] In der Regel sollten aus Ausfall- und Performancegründen mehrere von der BÄK-Root abgeleitete OCSP-Signer-Zertifikate zum Einsatz kommen, um die hier beschriebenen nicht-qualifizierten OCSP-Responder-Antworten zu ENC/AUTH-Zertifikaten zu signieren.

[3] Da Vor- und Nachnamen der Personen stets vollständig in anderen Zertifikatsattributen enthalten sind, und Pseudonyme für den elektronischen Arztausweis nicht zulässig sind, soll die Kennzeichnung von technisch bedingten Kürzungen im CN durch: PN nur durchgeführt werden, wenn dies vom SigG gefordert wird. D. h. falls die Länge des vollständigen Namens (inkl. vollst. Titeln und ggf. „:PN“) 64 Zeichen übersteigt, muss er

| | |
|---|---|
| SubjectPublicKeyInfo | Zertifizierter Public Key (einschl. AlgorithmusOID). RSA2048bit min., opt.: 256bit ECDSA[5] |
| Extensions | siehe Tabelle 2 |
| SignatureAlgorithm | |
| SignatureAlgorithm | OID des verwendeten Algorithmus (bzgl. Signatur des Zertifizierers) |
| SignatureValue | |
| SignatureValue | Bit-String (octetString) |

*Tabelle 1: Zertifikatstypen mit Grundstruktur und Namenskonzept. Attributzertifikate werden in einem gesonderten Dokument behandelt*

gekürzt werden. Die Kürzungsregeln sind an den Kürzungsregeln der eGK angelehnt. (Wenn der Inhalt des CN anschließend nicht dem Namen des Zertifikatsinhabers entspricht, kann der CN als Pseudonym gekennzeichnet werden (Suffix: „:PN“), falls dies durch Anforderungen der SigG-Bestätigung notwendig ist.)

[4] Das Attribut serialNumber im ENC und AUTH-Zertifikat soll den gleichen Wert wie im qSIG-Zertifikat haben. Hiermit soll ermöglicht werden, dass mit einem präsentierten AUTH-Zertifikat leichter das entsprechende ENC-Zertifikat desselben eArztausweises, mittels Konstruktion des DN, aufgefunden werden kann.

[5] bzw. ECDH für Verschlüsselung. Signaturalgorithmen müssen stets konform zur jeweils aktuellen Algorithmenkatalog (z.Z. [externAlgCat]) sein und in Abstimmung mit der BÄK eingesetzt werden

| Name des Feldes | Root | CA | CROSS | End-Entity qSIG | End-Entity AUTH | End-Entity ENC | CRL-Signer | OCSP-Signer | qCA (nicht normativ) |
|---|---|---|---|---|---|---|---|---|---|
| **Extensions** | | | | | | | | | |
| AuthorityKeyIdentifier | Key Identifier vom öffentlichen Schlüssel des Ausstellers | | | | | | | | |
| SubjectKeyIdentifier | Key Identifier vom öffentlichen Schlüssel des Zertifikatsinhabers | | | | | | | | |
| KeyUsage (CRITICAL) | keyCertSign | keyCertSign | keyCertSign | content commitment | key encipherment,<br>digital-signature | key encipherment, data encipherment | cRLSign | content commitment | keyCertSign |
| CertificatePolicy[6] | [1]Zertifikatsrichtlinie: Richtlinienkennung={s. Kap. 4.9.4} [1,1]Richtlinien-qualifier-informationen: Richtlinien-qualifier Id=CPS Qualifier:<br>http://www.e-arztausweis.de/policies/root_policy.html | [1]Zertifikatsrichtlinie: Richtlinienkennung={s. Kap. 4.9.4} [1,1]Richtlinien-qualifier-informationen: Richtlinien-qualifier Id=CPS Qualifier:<br>http://www.e-arztausweis.de/policies/ca_policy.html | [1]Zertifikatsrichtlinie: Richtlinienkennung={ s. Kap. 4.9.4} [1,1]Richtlinien-qualifier-informationen: Richtlinien-qualifier Id=CPS Qualifier:<br>http://www.e-arztausweis.de/policies/ca_policy.html | [1]Zertifikatsrichtlinie: Richtlinienkennung=id-commonpki-cp-accredited<br>[1]Zertifikatsrichtlinie: Richtlinienkennung={ s. Kap. 4.9.4} [1,1]Richtlinien-qualifier-informationen: Richtlinien-qualifier Id=CPS Qualifier: http://www.e-arztausweis.de/policies/EE_policy.html | [1]Zertifikatsrichtlinie: Richtlinienkennung={ s. Kap. 4.9.4} [1,1]Richtlinien-qualifier-informationen: Richtlinien-qualifier Id=CPS Qualifier:<br>http://www.e-arztausweis.de/policies/EE_policy.html | [1]Zertifikatsrichtlinie: Richtlinienkennung={ s. Kap. 4.9.4} [1,1]Richtlinien-qualifier-informationen: Richtlinien-qualifier Id=CPS Qualifier:<br>http://www.e-arztausweis.de/policies/EE_policy.html | [1]Zertifikatsrichtlinie: Richtlinienkennung={ s. Kap. 4.9.4} [1,1]Richtlinien-qualifier-informationen: Richtlinien-qualifier Id=CPS Qualifier:<br>http://www.e-arztausweis.de/policies/ca_policy.html | [1]Zertifikatsrichtlinie: Richtlinienkennung={ s. Kap. 4.9.4} [1,1]Richtlinien-qualifier-informationen: Richtlinien-qualifier Id=CPS Qualifier:<br>http://www.e-arztausweis.de/policies/ca_policy.html | [1]Zertifikatsrichtlinie: Richtlinienkennung=id-commonpki-cp-accredited |

---

[6] Die hier aufgeführten OIDs und URLs (Konfigurationsdaten) stellen lediglich den aktuellen Stand dar und können von der BÄK ohne Anpassung dieses Dokumentes geändert werden.

| Name des Feldes | Root | CA | CROSS | End-Entity qSIG | End-Entity AUTH | End-Entity ENC | CRL-Signer | OCSP-Signer | qCA (nicht normativ) |
|---|---|---|---|---|---|---|---|---|---|
| BasicConstraints (CRITICAL) | CA=true | CA=true, Length=0 | CA=true | CA=FALSE | CA=FALSE | CA=FALSE | CA=FALSE | CA=FALSE | CA=true, Length=0 |
| CrlDistributionPoints<br>(URL-konforme UTF-8-Kodierung)<br>(Konfigurationsdaten, können von der BÄK geändert werden, s.[baekConfigData]) | URL=ldap://ldap.e-arztausweis.de:389/ CN=CRL, O=Bundes%5CC3%5CA4rztekammer, C=DE?certificateRevocationList?base?objectClass=cRLDistributionPoint<br>CRLIssuer= CN=BÄK CRL Signer 1:PN, O=Bundesärztekammer, C=DE | | | CRL-DP des Ausstellers, Struktur wie beim Root-Zertifikat, CRLIssuer exakt wie im Zertifikat kodiert (auch Reihenfolge), ldap-url muss URL-konform und UTF-8-kodiert sein. | | | URL=ldap://ldap.e-arztausweis.de:389/ CN=CRL, O=Bundes%5CC3%5CA4rzte-kammer, C=DE?certificateRevocationList?base?objectClass=cRLDistributionPoint<br>CRLIssuer= CN=BÄK CRL Signer 1:PN, O=Bundesärztekammer, C=DE | | CRL-DP der Bundesnetz-agentur |
| Admission (ggf. in Attribut-Zertifikaten) | nicht gesetzt | nicht gesetzt | nicht gesetzt | flexibel, nach Festlegung der BÄK | gesetzt, s.u. | gesetzt, s.u. | nicht gesetzt | nicht gesetzt | nicht gesetzt |
| ValidityModel | id-validity-Model-chain {1 3 6 1 4 1 8301 3 5 1} | id-validity-Model-chain {1 3 6 1 4 1 8301 3 5 1} | id-validity-Model-chain {1 3 6 1 4 1 8301 3 5 1} | id-validity-Model-chain {1 3 6 1 4 1 8301 3 5 1} | nicht gesetzt | nicht gesetzt | id-validity-Model-chain {1 3 6 1 4 1 8301 3 5 1} | id-validity-Model-chain {1 3 6 1 4 1 8301 3 5 1} | id-validity-Model-chain {1 3 6 1 4 1 8301 3 5 1} |
| | | | | | | | | | |
| SubjectAlternativeName | nicht gesetzt | nicht gesetzt | nicht gesetzt | nicht gesetzt | rfc822Name =<e-mail> | rfc822Name =<e-mail> | nicht gesetzt | nicht gesetzt | nicht gesetzt |
| extendedKeyUsage | nicht gesetzt | nicht gesetzt | nicht gesetzt | nicht gesetzt | clientAuth<br>email-Protection | nicht gesetzt | nicht gesetzt | OCSP-Signing | nicht gesetzt |
| optional: additionalInformation | nicht gesetzt | nicht gesetzt | nicht gesetzt | ggf. Text gem. Policy[7] | ggf. Text gem. Policy | ggf. Text gem. Policy | nicht gesetzt | nicht gesetzt | nicht gesetzt |

[7] Wenn kein Text aufgenommen wird, dann darf die Extension nicht gesetzt werden. Dies gilt auch für AUTH und ENC-Zertifikate

| Name des Feldes | Root | CA | CROSS | End-Entity qSIG | End-Entity AUTH | End-Entity ENC | CRL-Signer | OCSP-Signer | qCA (nicht normativ) |
|---|---|---|---|---|---|---|---|---|---|
| QCStatements | nicht gesetzt | nicht gesetzt | nicht gesetzt | QcCompliance, ggf. EuLimit-Value | nicht gesetzt | nicht gesetzt | nicht gesetzt | nicht gesetzt | QcCompliance |
| optional: Restriction | nicht gesetzt | nicht gesetzt | nicht gesetzt | ggf Text gem. Policy[8] | nicht gesetzt | nicht gesetzt | nicht gesetzt | nicht gesetzt | nicht gesetzt |
| authorityInfoAccess | method=OCSP, URI des jeweiligen OCSP-Responders | | | | | | | | |

*Tabelle 2: Zertifikatsprofile – Extensions*

Die URLs und DNs in der Tabelle sind nicht normativ und können geändert werden.

[8] Wenn kein Text aufgenommen wird, dann darf die Extension nicht gesetzt werden. Für AUTH und ENC-Zertifikate wird Restriction nicht gesetzt, da sie von keiner Softwarekomponente ausgewertet werden kann. Es können Übergangsbestimmungen gelten.

# 2 Übersicht von Zertifikatstypen für Basiszertifikate

Für alle Zertifikate (auch die nicht-qualifizierten Zertifikate) gelten dieselben Sicherheitsanforderungen, wie für qualifizierte Zertifikate. Attributzertifikate werden in einem gesonderten Dokument beschrieben.

Es werden folgende Zertifikatstypen im Kontext der e-Arztausweise verwendet (s. Tabelle 1):

- Root-Zertifikate (nicht-qualifiziert): Es sind selbstsignierte Zertifikate, ausgegeben von der Bundesärztekammer als Arbeitsgemeinschaft und im Auftrag der Landesärztekammern (Herausgeber der e-Arztausweise). Mit Hilfe der Root-Zertifikate werden nicht-qualifizierte CA-Zertifikate ausgestellt, die nicht-qualifizierte User-Zertifikate (End-Entity-Zertifikate) für Verschlüsselung und Authentisierung ausstellen. Außerdem werden Cross-Zertifikate, (s. u.) sowie nicht-qualifizierte CRL-Signer-Zertifikate und nicht-qualifizierte OCSP-Signer-Zertifikate ausgestellt.
- Cross-Zertifikate (nicht-qualifiziert):
    - Cross-Zertifikate zur Zertifizierung von qualifizierten CA-Zertifikaten (einseitige Cross-Zertifizierung): Werden von den o. g. Root-Zertifikaten der Bundesärztekammer ausgestellt. Sie zertifizieren qualifizierte CA-Zertifikate der von der Bundesärztekammer zugelassenen nach SigG akkreditierten Zertifizierungsdiensteanbieter. Somit ist eine Ableitung (Zertifikatspfad-Validierung) der qualifizierten Signaturzertifikate der Ärzte auf das Root-Zertifikat der Bundesärztekammer möglich.
    - Cross-Zertifikate zwischen Root-Zertifikaten der Bundesärztekammer (doppelseitige Cross-Zertifizierung, Kette): Werden zwischen Root-Zertifikaten der Bundesärztekammer ausgestellt, damit eine Vertrauenskette zwischen den Root-Zertifikaten gebildet wird. Das Modell entspricht dem Modell der Cross-Zertifizierung zwischen Root-Zertifikaten der Bundesnetzagentur.
- CA-Zertifikate (nicht-qualifiziert): Werden von den Root-Zertifikaten der Bundesärztekammer ausgestellt. Stellen nicht-qualifizierte Authentisierungs- und Verschlüsselungs-Zertifikate für Ärzte aus, die auf Arztausweisen aufgebracht werden.
- End-Entity qSig-Zertifikate (qualifiziert): Werden von qualifizierten CA-Zertifikaten der von der Bundesärztekammer zugelassenen Zertifizierungsdiensteanbieter ausgestellt und dienen der Ausstellung bzw. Überprüfung von qualifizierten Signaturen. Sie erfüllen alle Anforderungen an qualifizierten Zertifikaten mit Anbieterakkreditierung.
- End-Entity ENC-Zertifikate (nicht qualifiziert): Werden von den o. g. nicht-qualifizierten CA-Zertifikaten ausgestellt und dienen der Verschlüsselung von Daten.
- End-Entity AUTH-Zertifikate (nicht qualifiziert): Werden von den o.g. nicht-qualifizierten CA-Zertifikaten ausgestellt und dienen der Authentisierung des Arztes.
- CRL-Signer-Zertifikate (nicht qualifiziert): Werden von den o. g. nicht-qualifizierten Root-Zertifikaten der Bundesärztekammer ausgestellt und dienen der Ausstellung von CRLs für alle nicht-qualifizierte Zertifikate, die in diesem Dokument beschrieben sind.
- OCSP-Signer-Zertifikate (nicht qualifiziert): Werden von den o. g. nicht-qualifizierten Root-Zertifikaten der Bundesärztekammer ausgestellt und dienen der Ausstellung

von OCSP-Responses in einem Verzeichnisdienst für alle nicht-qualifizierten Zertifikate, die in diesem Dokument beschrieben sind.

- qCA-Zertifikate: (qualifiziert). Werden von qualifizierten Root-Zertifikaten der Bundesnetzagentur ausgestellt. Stellen qualifizierte End-Entity-qSig-Zertifikate (s. o.) für Ärzte aus. Deren Beschreibung erfolgt deskriptiv und nicht normativ. Die Namensgebung (SubjectDN) ist jedoch, vorbehaltlich der Genehmigung durch die Bundesnetzagentur, normativ.

# 3 Übersicht und Beschreibung von Zertifikatsstrukturen

Die folgenden Tabellen enthalten eine Übersicht über die einzelnen Zertifikatsstrukturen mit einer kurzen Beschreibung. Die Kritikalität einer Extension wird mit „!!“ aufgeführt; nicht gekennzeichnete Extensions sind nicht kritisch. Ein Client (Software), der eine „kritische“ Extension nicht versteht, muss das Zertifikat ablehnen. Optionale Extensions sind für den Antragsteller oder für die Ärztekammer optional. Sie müssen von den Zertifizierungsdienste-anbietern unterstützt werden. Leere Extensions („nicht gesetzt“) werden im Zertifikat nicht aufgenommen.

| Name der Struktur | Semantik | Inhalt |
|---|---|---|
| tbsCertificate | | |
| Version | X.509-Versionsnummer | 2 (Version 3) |
| SerialNumber | Zertifikatsseriennummer. Eindeutig für alle von einem CA-Zertifikat ausgestellten Zertifikate | z.B. „1234567890“, muss als positiver Integer kodiert sein (MSB=0), max. 20 bytes lang |
| Signature | Kennzeichner (OID) für den Algorithmus für die Signatur des Zertifikates | z.B. sha256WithRsaEncryption |
| Issuer | Name des Ausstellers des Zertifikates (CA-Zertifikat). Bei Root-Zertifikaten: =Subject. Suffix: „:PN“ | s. Tabelle 1, UTF-8 kodiert (außer C, das printableString-kodiert ist) |
| Validity | Gültigkeitszeitraum des Zertifikates | s. Tabelle 1 |
| Subject | Name des Zertifikatsinhabers, ggf. mit :PN als Pseudonym | s. Tabelle 1, UTF-8 kodiert (außer C und serialNumber, die printableString-kodiert sind) |
| SubjectPublicKeyInfo | Öffentlicher Schlüssel des Zertifikatsinhabers, welches mit dem Zertifikat zertifiziert wird (Zuordnung zwischen Schlüssel und Person) | z. B. rsaEncryption-Schlüssel Länge: z. Z. 2048bit, optional: ECDSA-256 (nach Abstimmung), für alle Zertifikate stets nach jeweils aktuellem Algorithmenkatalog der Bundesnetzagentur |
| Extensions | siehe Tabelle 4 | |
| SignatureAlgorithm | | |
| SignatureAlgorithm | OID des Signatur-algorithmus, mit dem das Zertifikat signiert wurde | z. B. sha256WithRSAEncryption |
| SignatureValue | | |

| | | |
|---|---|---|
| SignatureValue | Signatur der Zertifizierungsstelle | Bit-String |

*Tabelle 3: Zertifikate, Grundstruktur*

| Name der Struktur | Semantik | | Inhalt |
|---|---|---|---|
| **Extensions** | | | |
| AuthorityKeyIdentifier | | Informationen zur Identifikation des öffentlichen Schlüssels des CA-Zertifikates | nur keyIdentifier = OCTET STRING, 20 Bytes, SHA-1 aus dem public-key ohne Tag- und Längen-bytes (s.[externRFC5280]) |
| SubjectKeyIdentifier | | Informationen zur Identifikation des öffentlichen Schlüssels des Zertifikates | OCTET STRING, 20 Bytes, wie AuthorityKeyIdentifier |
| KeyUsage | !! | Erlaubte Zwecke, für die das Zertifikat benutzt werden darf | z. B. content commitment |
| CertificatePolicy | | Policy/Policies der Zertifizierungsstelle für das Zertifikat. Identifikationsmerkmal für eArztausweis-Zertifikate | s. Tabelle 2<br>Konfigurationsdaten (können von der BÄK ohne Anpassung dieses Dokumentes geändert werden, s [baekConfigData]) |
| BasicConstraints | !! | Information, ob das Zertifikat ein CA-Zertifikat ist und ob es weitere CA-Zertifikate (Anzahl von Ebenen) ausstellen darf | s. Tabelle 2 |
| Admission | | Information über das Berufsgruppenattribut sowie ggf. die optionale Telematik-ID | enthält admissionAuthority (zuständige Landesärzte-kammer). professionItem als OID (s. Kap. 4.9.6) und Text (Ärztin/Arzt) sowie die Telematik-ID in der registrationNumber. Wird in ENC, AUTH (als Extension) und ggf. in Signatur- und/oder Attributzertifikaten (als Extension bzw. Attribut, aber ohne Telematik-ID) aufgenommen. |

| Name der Struktur | Semantik | | Inhalt |
|---|---|---|---|
| **Extensions** | | | |
| | | | Konfigurationsdaten |
| Restriction | | Information über Einschränkungen für die Anwendung des Zertifikats | optional. Enthält ggf. Text, welcher die Anwendung des Zertifikats beschränkt |
| AdditionalInformation | | Weitere Informationen nicht einschränkender Natur über das Zertifikat | optional. Enthält ggf. Text, welcher das Zertifikat als e-Arztausweis-Zertifikat ausweist |
| ExtendedKeyUsage | | Information über spezifische Anwendungszwecke | s. Tabelle 2 |

| Name der Struktur | Semantik | | Inhalt |
|---|---|---|---|
| **Extensions** | | | |
| CrlDistributionPoints | | Informationen, wie und wo die zugehörige Sperrliste (CRL) bezogen werden kann und wer der Aussteller dieser Liste ist | s. Tabelle 2 |
| QcStatements | | Informationen, dass das Zertifikat qualifiziert i. S. der EU-Direktive ist | Konformität zur EU-Direktive, optional Beschränkung (Oberlimit) für finanzielle Transaktionen |
| AuthorityInfoAccess | | Quelle für Status-informationen für die Validierung des Zertifikats | URL des zuständigen OCSP-Responders |
| SubjectAlternativeName | | Optional. Falls vorhanden, wird die E-Mail-Adresse aufgenommen | s. Tabelle 2, Optional. Darf nicht im qSIG-Zertifikat aufgenommen werden |
| ValidityModel | | Beschreibt das zugrunde liegende Gültigkeitsmodell für die Zertifikate und Signaturen | OID für das Kettenmodell<br>id-validityModel-chain<br>{1 3 6 1 4 1 8301 3 5 1} |

*Tabelle 4: Extensions mit Wert*

# 4 Beschreibung der Zertifikatsfelder und gültige Werte

## 4.1 Grundstruktur

Die Zertifikate sind konform zur Spezifikation X.509v3 [externRFC5280]. Sie enthalten folgende ASN.1-Strukturen, deren spezifische Ausprägung von Zertifikatstyp abhängt (s. Übersicht):

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

Das `tbsCertificate` („to be signed Certificate“) enthält alle Elemente des Zertifikats (ohne die Signatur der Zertifizierungsstelle), in der Form, wie sie von der Zertifizierungsstelle signiert werden. Insbesondere sind auch der öffentliche Schlüssel und die Identifikationsdaten (SubjectDN) des Zertifikatsinhabers enthalten.

Der Kennzeichner (OID) des Algorithmus, mit dem das Zertifikat signiert worden ist, ist in der Struktur `signatureAlgorithm` enthalten. Das Feld `signatureValue` enthält die Signatur der Zertifizierungsstelle.

## 4.2 X.509-Version, Feld `Version`

Das Feld `Version` definiert die X.509-Version des Zertifikats und hat den Wert 2 für X.509v3.

## 4.3 Seriennummer des Zertifikats, Feld `SerialNumber`

Das Feld `SerialNumber` (nicht zu verwechseln mit dem Attribut serialNumber im SubjectDN!) enthält die Seriennummer des Zertifikats, kodiert als signed Integer (also MSB=0) und darf eine Maximallänge von 20 Octets haben [externRFC5280]. Die Seriennummer muss für alle Zertifikate eines Issuers (CA-Zertifikat) eindeutig sein.

## 4.4 Erlaubte Signaturalgorithmen, Feld `Signature`

Das Feld `Signature` definiert den zugrunde liegenden Signaturalgorithmus, der benutzt wurde, um das Zertifikat zu signieren.

ASN.1-Struktur:

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

Gültige Werte:

Folgende Signaturalgorithmen sind derzeit zulässig (laut aktuellem Algorithmenkatalog der Bundesnetzagentur [externAlgCat]):

- sha256WithRSAEncryption OBJECT IDENTIFIER ::= { 1 2 840 113549 1 1 11 }
- sha512WithRSAEncryption OBJECT IDENTIFIER ::= { 1 2 840 113549 1 1 13 }

Derzeit (03.2009) muss sha256WithRSAEncryption verwendet werden. Die Signaturalgorithmen für alle Zertifikatsklassen müssen stets den gesetzlichen Anforderungen für qualifizierte Signaturzertifikate entsprechen und können aus diesem Grund ohne Anpassung dieses Dokumentes in Abstimmung mit der BÄK geändert werden.

Optional können auch ECDSA-Signaturalgorithmen zugelassen werden.

Im Feld `parameters` können zusätzliche Parameter definiert werden, wenn sie benötigt werden bzw. zulässig sind, bspw. Kurvenparameter für ECDSA-Algorithmen.

### 4.5 Gültigkeitszeitraum, Feld `Validity`

Der Gültigkeitszeitraum des Zertifikates wird im Feld `Validity` definiert. Erlaubte Werte sind in Tabelle 1 eingetragen. Ein Zertifikat darf nicht länger gültig sein, als die zugrunde liegenden Algorithmen nach [externAlgCat]. Ausnahmen in der Ausstellung gelten gemäß dem Kompromissmodell (Gültigkeitsmodell) für nicht-qualifizierte Root- CA- und CROSS-Zertifikaten [baekValidityModel].

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

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

Bis zum Jahr 2049 muss `Time` als `UTCTime` codiert werden, danach als `GeneralizedTime`. Das Format entspricht YYMMDDhhmmssZ für `UTCTime` und YYYYMMDDhhmmssZ für `GeneralizedTime`. Qualifizierte Signatur- und Attributzertifikate dürfen länger gültig sein, als ihr Aussteller. Alle nicht-qualifizierte Zertifikate werden nach dem „Kompromissmodell“ [baekValidityModel] ausgestellt und dürfen demnach nicht länger gültig sein als ihr Aussteller.

### 4.6 Aussteller des Zertifikats, Feld `Issuer`

Der Name des signierenden Zertifikats (Ausstellers) wird im Feld `Issuer` geführt. Es muss exakt mit dem Inhalt des Subject-Feldes des Ausstellers inkl. Kodierung und Reihenfolge der Strukturen übereinstimmen (so dass die Hashwerte beider Strukturen übereinstimmen), auch wenn ggf. Abweichungen im [externRFC5280] toleriert werden. Bei Root-Zertifikaten muss das Feld `Issuer` mit dem Subject-Feld desselben Zertifikats (self signed) exakt übereinstimmen (inkl. Reihenfolge der Strukturen und Kodierung).

Folgende Attribute werden benutzt. Die erlaubten Werte sind in der Tabelle 1 aufgeführt:

- `countryName` (DE, ISO 3166 Code)
- `organizationName` (gemäß Tabelle 1)
- optional `organizationalUnitName`
- `commonName` (gemäß Tabelle 1)

Die ASN.1-Struktur:

```
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
```

Die erlaubten Attribute werden in Tabelle 5 spezifiziert (id-at = 2.5.4).

| Attribut | OID | Kodierung | max. Länge |
|---|---|---|---|
| commonName | {id-at 3} | UTF8 | 64 |
| organizationName | {id-at 10} | UTF8 | 64 |
| (organizationalUnitName) | {id-at 11} | UTF8 | 64 |
| countryName | {id-at 6} | PrintableString | 2 |

*Tabelle 5: Attribute im IssuerDN*

### 4.7 Zertifikatsinhaber, Feld `Subject`

Im Feld `Subject` wird die Zuordnung zur Person des Zertifikatsinhabers hergestellt, indem ein Teil seiner persönlichen Daten aufgenommen wird. Ein SubjectDN darf nicht zwei oder mehreren Personen zugeordnet sein (aus diesem Grund werden Nummernräume für die serialNumber des SubjectDN für die ZDA definiert). Dies gilt auch für Pseudonyme. (Ausnahme: Pseudonyme bzw. SubjectDNs von nicht-SigG-konformen CRL-Signer-Zertifikaten, ausgestellt von der BÄK-Root-Instanz dürfen mehrmals verwendet werden und auch einer anderen Person zugeordnet werden, damit der Aussteller der CRL dem in der

Extension CRL-DP eingetragenen CRL-Issuer entspricht. Generell dürfen Trustcenter-Zertifikate der BÄK-Root-Instanz „umpersonalisiert“ werden, s. Konzept der BÄK-Root-Instanz). Die Kürzungsregeln sind an den Kürzungsregeln der eGK angelehnt. Wenn der Inhalt des CN nicht den Namen des Zertifikatsinhabers entspricht, muss nur dann der CN als Pseudonym gekennzeichnet werden (Suffix: „:PN“), falls dies durch Anforderungen der SigG-Bestätigung notwendig ist.

Folgende Attribute werden verwendet:

- `countryName` (vorgeschrieben)
- `organizationName` (gemäß Tabelle 1, nicht für User-Zertifikate. Ausnahme: (administrative) Zertifikate für Ärztekammermitarbeiter, O=Ärztekammer xyz)
- optional `organizationalUnitName` , nicht für User-Zertifikate
- `serialNumber` (gemäß Tabelle 1)
- `surname`
- `givenName`
- `(derzeit nicht zulässig, RFU): title`
- `commonName` (vorgeschrieben, Schema gemäß Tabelle 1)

Der `commonName` enthält den vollständigen Namen des Inhabers, ohne akademische Titel (auch wenn sie im Personalausweis des Antragstellers eingetragen sind). Die Länge des Attributes ist auf 64 Zeichen beschränkt. Falls der vollständige Name nicht aufgenommen werden kann (z. B. weil er zu lang ist), dann muss, nur dann wenn dies aus gesetzlichen Bestimmungen hervorgeht, der `commonName` als Pseudonym gekennzeichnet werden. In diesem Fall muss der Zusatz „:PN“ (ohne Anführungsstrichen) aufgenommen werden; die effektive Länge reduziert sich damit auf 61 Zeichen. Falls eine Kürzung vorgenommen werden soll, entsprechen die Kürzungsregeln den Regelungen in der eGK-Spezifikation:

- Rufname und Nachname bleiben vollständig, Vornamen werden auf den ersten Buchstaben plus Punktzeichen gekürzt
- falls immer noch >61 bzw. 64 Zeichen: der Nachname wird gekürzt und mit Punktzeichen gekennzeichnet, so dass die Gesamtlänge (ggf. inkl. :PN) 64 Zeichen beträgt

Der `surname` enthält (zusätzlich zum commonName) den Nachnamen des Inhabers. Evtl. vorhandene Namensbestandteile wie „Graf von“, „jr.“, so genannte „generation qualifier“ (üblich im amerikanischen Sprachraum, z.B. „III“) usw. werden im `surname` aufgenommen, wenn sie im Reisepass oder Personalausweis als Teile des Nachnamens betrachtet werden. Ggf. im Personalausweis aufgenommene akademische Titel (DR) werden nicht im surname aufgenommen

Das Attribut `givenName` enthält (zusätzlich zum commonName) alle Vornamen des Inhabers.

Das optionale Attribut title kann (zusätzlich zum commonName) zukünftig den akademischen Titel des Inhabers enthalten. Derzeit darf es nicht verwendet werden.

Die Attribute serialNumber, givenName, surname, ggf. title und commonName werden in einem SET als ein einziges multivaluedRDN kodiert. Die entsprechenden Kodierungsregeln von X.690 (Reihenfolge im SET) müssen berücksichtigt werden

User-Zertifikate müssen immer, (d. h. auch wenn sie als Pseudonym gekennzeichnet sind) den Namen des Antragstellers enthalten, ggf. gekürzt nach o. g. Regeln. Attribute, wie `organization` dürfen für User-Zertifikate nicht gesetzt werden. Eine Ausnahme stellen Zertifikate für Ärztekammer-Mitarbeiter dar, die für die Administration der PKI notwendig sind (z. B. für die Bestätigung von Berufsgruppenattributen in einem elektronischen Prozess). In diesem Falle wird das Attribut O=<die jeweilige Ärztekammer> gesetzt (s. [baekVerzD]; Struktur des Baums).

Root-, CA- und weitere PKI-Verwaltungszertifikate müssen Pseudonyme verwenden, die ihre Verwendung bezeichnen (z. B. CRL-Signer 1:PN oder DIR-3:PN). Sie sind Personen zugeordnet. Im Attribut `organization` muss die Bezeichnung des Betreibers enthalten sein. Ggf. dürfen weitere Angaben im `organizationalUnit` enthalten sein.

Um eine Unterscheidung bei Namensgleichheit zu gewährleisten, wird das Attribut `serialNumber` verwendet. Es wird empfohlen das Attribut auf End-Entity-Zertifikate zu beschränken, jedoch dürfen es auch CA- oder Root-Zertifikate enthalten. Um unterschiedliche Namen im globalen LDAP-Tree über alle ZDAs zu erreichen, sind zweistellige Prefixe definiert und den ZDAs zugewiesen [baekConfigData]. Das `serialNumber`-Attribut aller User-Zertifikate eines ZDA muss mit dem zugewiesenen Prefix anfangen. Sämtliche Zertifikate desselben eArztausweises müssen die gleiche Nummer im `serialNumber`-Attribut enthalten.

Das Attribut `country` enthält den ISO-3166 Code des Landes, also `DE`, kodiert als printableString.

Die Reihenfolge der `RelativeDistinguishedNames` in der `RDNSequence` ist:

country,

ggf. organizationName,

ggf. organizationalUnitName,

(givenName+surname+title+serialNumber+commonName)[9]

Die ASN.1-Struktur vom Feld `subject`:

```
Name ::= CHOICE {RDNSequence}
RDNSequence ::= SEQUENCE OF RelativeDistinguishedName
RelativeDistinguishedName ::= SET OF AttributeTypeAndValue
```

[9] Die Reihenfolge der einzelnen Elemente im SET muss DER-konform sein; die hier beschriebene Reihenfolge ist exemplarisch. Für Root, CA, CRL-Signer, TSS und OCSP-Signer-Zertifikate sind surname, givenName, title und ggf. serialNumber nicht relevant.

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

AttributeType ::== OBJECT IDENTIFIER

AttributeValue ::= ANY DEFINED BY AttributeType
```

Die erlaubten Attribute werden in Tabelle 6 spezifiziert (id-at = 2.5.4).

| Attribut | OID | Kodierung | max. Länge |
|---|---|---|---|
| commonName | {id-at 3} | UTF8 | 64 |
| (title) | {id-at 12} | UTF8 | 64 |
| surname | {id-at 4} | UTF8 | 64 |
| givenName | {id-at 42} | UTF8 | 64 |
| serialNumber | {id-at 5} | PrintableString | 64 |
| organizationName | {id-at 10} | UTF8 | 64 |
| (organizationalUnitName) | {id-at 11} | UTF8 | 64 |
| countryName | {id-at 6} | PrintableString | 2 (ISO 3166 Code) |

*Tabelle 6: Attribute im SubjectDN*

### 4.8 Der öffentliche Schlüssel, Feld `SubjectPublicKeyInfo`

Das Feld enthält den öffentlichen Schlüssel des Zertifikatsinhabers. In der Struktur wird auch der zugrunde liegende Algorithmus für den Schlüssel spezifiziert.

Die ASN.1-Struktur:

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

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

Es werden RSA Schlüssel verwendet; der OID lautet:

- rsaEncryption {1 2 840 113549 1 1 1}

(ECDSA-Schlüssel sind – nach Abstimmung mit der BÄK – ebenso erlaubt und für die Zukunft angedacht).

Die aktuell gültigen Schlüssellängen werden in der Tabelle 1 aufgeführt. Sie müssen stets dem jeweils aktuellen Algorithmenkatalog der BNetzA (derzeit [externAlgCat]) entsprechen und müssen deshalb regelmäßig angepasst werden.

Das Feld `parameters` enthält die für den Algorithmus zulässigen Parameter.

### 4.9 Zertifikatserweiterungen, Feld `Extensions`

In der Struktur „Extensions“ werden Zertifikatserweiterungen aufgenommen.

Die ASN.1-Struktur:

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

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

In `extnId` wird ein Bezeichner für die in `extnValue` enthaltene Extension definiert. Der boolean `critical` zeigt an, ob die Extension kritisch ist. Wenn eine Extension als `critical` gesetzt ist, muss ein Client, der sie nicht versteht, das Zertifikat ablehnen.

In Tabelle 7 werden die Extensions aufgeführt, die in den Zertifikaten verwendet werden.

| Extension | OID | Zertifikatstyp | critical? |
|---|---|---|---|
| AuthorityKeyIdentifier | {2 5 29 35} | alle | non critical |
| SubjectKeyIdentifier | {2 5 29 14} | alle | non critical |
| KeyUsage | {2 5 29 15} | gem. Tabelle 2 | critical |
| CertificatePolicies | {2 5 29 32} | gem. Tabelle 2 | non critical |
| SubjectAltNames | {2 5 29 17} | User-Zertifikate | non critical |
| Admission | {1 3 36 8 3 3} | User-Zertifikate | non critical |
| BasicConstraints | {2 5 29 19} | alle | critical |
| CRLDistributionPoints | {2 5 29 31} | gem. Tabelle 2 | non critical |
| extendedKeyUsage | {2 5 29 37} | gem. Tabelle 2 | non critical |
| QCStatements | {1.3.6.1.5.5.7.1.3} | gem. Tabelle 2 | non critical |
| additionalInformation | {1 3 36 8 3 15} | gem. Tabelle 2 | non critical |
| Restriction | {1 3 36 8 3 8} | gem. Tabelle 2 | non critical |
| authorityInfoAccess | {1.3.6.1.5.5.7.1} | gem. Tabelle 2 | non critical |
| ValidityModel | {1 3 6 1 4 1 8301 3 5} | gem. Tabelle 2 | non critical |

Tabelle 7: Extensions

#### 4.9.1 AuthorityKeyIdentifier (2.5.29.35)

Die nicht-kritische Extension AuthorityKeyIdentifier (OID: {2 5 29 35}) dient zur Identifizierung des öffentlichen Schlüssels des Ausstellers. Sie wird bei sämtlichen Zertifikatstypen (auch Root) gesetzt.

ASN.1-Struktur:

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

Gültige Werte:

In der Extension wird nur das Feld keyIdentifier verwendet. Der Wert enthält den kompletten SHA-1 Hashwert über den `subjectPublicKey` (ohne Tag, Länge, Padding-Bits) des Ausstellers (bei Root-Zertifikaten: des Zertifikats).

### 4.9.2 SubjectKeyIdentifier (2.5.29.14)

Die nicht-kritische Extension Subject Key Identifier (OID: {2 5 29 14}) dient zur eindeutigen Identifizierung des öffentlichen Schlüssels des Zertifikats.

ASN.1-Struktur:

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

Gültige Werte:

Der Wert des keyIdentifiers enthält den kompletten SHA-1 berechneten Hashwert über den `subjectPublicKey` (ohne Tag, Länge, Padding-Bits) des Zertifikats.

### 4.9.3 KeyUsage (2.5.29.15)

Mit der keyUsage-Extension wird der Verwendungszweck des zum Zertifikat gehörenden privaten Schlüssels angegeben.

Diese Extension wird als „critical“ markiert.

ASN.1-Stuktur:

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

Die gültigen Werte für die jeweiligen Zertifikatstypen sind in der Tabelle 2 beschrieben. Die Extension wird in sämtlichen Zertifikatstypen aufgenommen.

### 4.9.4 CertificatePolicies (2.5.29.32)

Die Extension enthält die Referenzen zu den zugrunde liegenden Policies für die Zertifikate. Die CP und die zugehörige CPS werden über den OID der CP des Zertifikates, sowie über einen Link (URL) für die CPS des Trustcenters, referenziert. Die Extension ist „non-critical“.

Die PolicyOIDs und PolicyURLs sind Konfigurationsdaten, die jederzeit von der BÄK definiert und geändert werden können, ohne dass es einer Anpassung dieses Dokumentes bedarf.

Die neuen oder geänderten Konfigurationsdaten werden den ZDAs über das Dokument [baekConfigData] mitgeteilt und müssen in die Zertifikate aufgenommen werden.

Folgende OIDs sind derzeit (03.2009) gültig:

HPC Version 2.1.1, nicht qualifiziert

- Entwicklerkarten, HPC-Version 2.1.1:
  - 1.3.6.1.4.1.24796.1.2: id-baek-cp-hbaEntwicklerkarteArzt (Inhaber ist KEIN Arzt, nur für Entwicklerkarten)
- Test-HBAs[Arzt], HPC-Version 2.1.1:
  - 1.3.6.1.4.1.24796.1.1: id-baek-cp-hbaTestkarteArzt (Inhaber ist Ärztin/Arzt)

HPCqsig (kompatibel zu Version 2.1.1), qualifiziert

- eArztausweise, alle Zertifikatsklassen
  - 1.3.6.1.4.1.24796.1.10: id-baek-cp-eArztausweisV1 (Policy (Version 1) für den elektronischen Arztausweis mit qualifizierter Signatur)
- eArztausweise, SIG-Zertifikat (qualifiziert), zusätzlich:
  - 1.3.36.8.1.1: id-commonpki-cp-accredited

Die Zertifikate der qualifizierten HPCqsig-Karte haben also einen Policy-OID: id-baek-cp-eArztausweisV1. Das qualifizierte Signaturzertifikat hat zwei OIDs (id-baek-cp-eArztausweisV1 und id-commonpki-cp-accredited). Die Policy id-baek-cp-eArztausweisV1 entspricht der Gemeinsamen Policy [leoGemPolicy] und darf nur für elektronische Arztausweise mit qualifizierter Signatur, die von einer Ärztekammer als solche freigegeben wurden, verwendet werden.

HPC Version 2.3.0

- Entwicklerkarten, alle Zertifikatsklassen
  - 1.2.276.0.76.4.112: policy-muster-010000-cp (Unpersonalisiertes Zertifikat, Inhaber ist KEIN Arzt)
- Entwicklerkarten, AUT-Zertifikat, zusätzlich:
  - 1.2.276.0.76.4.75: hba-aut
- Entwicklerkarten, ENC-Zertifikat, zusätzlich:
  - 1.2.276.0.76.4.74: hba-enc
- Entwicklerkarten, SIG-Zertifikat, zusätzlich:
  - 1.2.276.0.76.4.73: hba-sig

Die Zertifikate der Entwicklerkarten der HPC-Spec Version 2.3.0 haben also zwei Policy-OIDs: policy-muster-010000-cp und den klassenspezifischen OID.

- eArztausweise, alle Zertifikatsklassen
  - 1.2.276.0.76.4.62: policy-hba-010000-cp (Gemeinsame Policy für die Ausgabe der HPC, [leoGemPolicy])
- eArztausweise, AUT-Zertifikat, zusätzlich:
  - 1.2.276.0.76.4.75: hba-aut
- eArztausweise, ENC-Zertifikat, zusätzlich:
  - 1.2.276.0.76.4.74: hba-enc
- eArztausweise, SIG-Zertifikat (qualifiziert), zusätzlich:
  - 1.2.276.0.76.4.72: hba-qes
  - 1.3.36.8.1.1: id- commonpki-cp-accredited

Die AUT und ENC Zertifikate der eArztausweise der HPC-Spec Version 2.3.0 haben also zwei Policy-OIDs, das qualifizierte Signaturzertifikat hat drei Policy-OIDs.

| Karte | Zertifikatstyp | PolicyOID |
|---|---|---|
| HPC Version 2.1.1, Entwicklerkarte | alle | 1.3.6.1.4.1.24796.1.2: id-baek-cp-hbaEntwicklerkarteArzt (Inhaber ist KEIN Arzt, nur für Entwicklerkarten) |
| HPC-Version 2.1.1, Test-HBAs[Arzt] | alle | 1.3.6.1.4.1.24796.1.1: id-baek-cp-hbaTestkarteArzt (Inhaber ist Ärztin/Arzt), |
| HPCqsig | AUT, ENC | 1.3.6.1.4.1.24796.1.10: id-baek-cp-eArztausweisV1 |
| HPCqsig | qSIG, qATTR | 1.3.6.1.4.1.24796.1.10: id-baek-cp-eArztausweisV1<br>1.3.36.8.1.1: id- commonpki-cp-accredited |
| HPC Version 2.3.0, Entwicklerkarte | AUT | 1.2.276.0.76.4.112: policy-muster-010000-cp<br>1.2.276.0.76.4.75: hba-aut |
| HPC Version 2.3.0, Entwicklerkarte | ENC | 1.2.276.0.76.4.112: policy-muster-010000-cp<br>1.2.276.0.76.4.74: hba-enc |
| HPC Version 2.3.0, Entwicklerkarte | SIG, ATTR | 1.2.276.0.76.4.112: policy-muster-010000-cp<br>1.2.276.0.76.4.73: hba-sig |
| HPC Version 2.3.0, Testkarte | | TBD |
| HPC Version 2.3.0, eArztausweis | AUT | 1.2.276.0.76.4.62: policy-hba-010000-cp<br>1.2.276.0.76.4.75: hba-aut |
| HPC Version 2.3.0, eArztausweis | ENC | 1.2.276.0.76.4.62: policy-hba-010000-cp<br>1.2.276.0.76.4.74: hba-enc |
| HPC Version 2.3.0, eArztausweis | qSIG, qATTR | 1.2.276.0.76.4.62: policy-hba-010000-cp<br>1.2.276.0.76.4.72: hba-qes<br>1.3.36.8.1.1: id- commonpki-cp-accredited |

Zertifikate für Ärztekammer-Mitarbeiter (für die Wahrnehmung administrativer Funktionen in der PKI), die keine Ärzte sind, enthalten diese OIDs nicht. Wenn Anforderungen (z. B. Sicherheitsanforderungen) einer aufgeführten Policy durch eine andere aufgeführte Policy im selben Zertifikat „aufgeweicht“ werden, dann gelten stets die jeweiligen schärferen Anforderungen.
ASN.1-Struktur:

```
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 }
CPSUri ::= IA5String
```

Der Wert vom `policyIdentifier` ist der in der Tabelle 2 aufgeführter OID. `policyQualifierId` hat den Wert `id-qt-cps`. Die URL auf die CPS der Bundesärztekammer wird als Wert im `CPSUri` aufgenommen (s. Tabelle 2).

### 4.9.5 SubjectAltNames (2.5.29.17)

In der nicht-kritischen SubjectAltNames-Extension wird die E-Mail-Adresse des Zertifikatsinhabers aufgenommen. Die Extension wird nur in User-Zertifikaten aufgenommen.

ASN.1-Struktur:

```
SubjectAltNames ::= GeneralNames
```

Das Regelwerk für die Bildung und Verwaltung von E-Mail-Adressen wird in einem separaten Dokument beschrieben. Diese Extension ist optional. Falls eine E-Mail-Adresse nicht aufgenommen wird, darf die (leere) Extension nicht gesetzt werden. In qualifizierten Signaturzertifikaten wird ebenso keine E-Mail-Adresse aufgenommen

### 4.9.6 Admission (1.3.36.8.3.3)

Die Admission-Extension beinhaltet das Berufsgruppenattribut „Ärztin/Arzt“ sowohl als Text als auch in Form einer maschinenlesbarer OID. Sie kann ferner die Telematik-ID – zum Zwecke des Berechtigungserhalts i. S. des Identitymanagements - beinhalten, soweit sie vom Antragsteller beantragt wurde. Die Admission-Extension wird bei User-Zertifikaten für Authentifizierung und Verschlüsselung sowie – nach vorheriger Festlegung der BÄK – bei qualifizierten Signaturzertifikaten (ohne Telematik-ID) aufgenommen.

ASN.1-Struktur:

```
id-commonpki-at-admission OBJECT IDENTIFIER ::= { commonpki-at 3 }
id-commonpki-at-namingAuthorities OBJECT IDENTIFIER ::= { commonpki-
at 11 }
AdmissionSyntax                 ::= SEQUENCE {
      admissionAuthority        GeneralName OPTIONAL,
      contentsOfAdmissions      SEQUENCE OF Admissions}

Admissions                      ::= SEQUENCE {
      admissionAuthority        [0] EXPLICIT GeneralName OPTIONAL,
      namingAuthority           [1] EXPLICIT NamingAuthority
OPTIONAL,
```

```
      professionInfos            SEQUENCE OF ProfessionInfo}

NamingAuthority                  ::= SEQUENCE {
      namingAuthorityId          OBJECT IDENTIFIER OPTIONAL,
      namingAuthorityUrl         IA5String OPTIONAL,
      namingAuthorityText        DirectoryString (SIZE(1..128))
OPTIONAL }

ProfessionInfo                   ::= SEQUENCE {
      namingAuthority            [0] EXPLICIT NamingAuthority
OPTIONAL,
      professionItems            SEQUENCE OF DirectoryString
(SIZE(1..128)),
      professionOIDS             SEQUENCE OF OBJECT IDENTIFIER
OPTIONAL,
      registrationNumber         PrintableString (SIZE(1..128))
OPTIONAL,
      addProfessionInfo          OCTET STRING OPTIONAL }
```

Gültige Werte:

Es wird genau eine admissionAuthority auf der obersten globalen Ebene der Extension gesetzt (also eine main admission authority). Diese besteht aus einem DistinguishedName (X.501-Name) mit folgenden Elementen:

- C=“DE“ (kodiert als printableString)
- O=“Landesärztekammer Nordrhein“ (Beispiel), UTF-8-kodiert

und bezeichnet die Bestätigende Stelle (bestätigende Landesärztekammer) für das Berufsgruppenattribut und die Telematik-ID.

Als admissionAuthority-Bezeichner der jeweiligen, bestätigenden Ärztekammern sind ausschließlich die den Codes entsprechenden, vollständigen, ausgesprochenen Namen der in [baekXML] mit dem XML-Schema-Elementen `<xarzt:aerztekamer>` bzw. `<xarzt:kammercode>` übertragenen Werten (Codes) zu verwenden. [Anmerkung: BZÄK → Bezirksärztekammer; LÄK → Landesärztekammer; ÄK → Ärztekammer]. [10]

Es wird genau eine „Admissions“-Struktur aufgenommen, die genau ein ProfessionInfo enthält. Weder eine admissionAuthority noch eine NamingAuthority werden auf diesem Level gesetzt.

Die ProfessionInfo enthält genau ein ProfessionItem, genau einen ProfessionOID und keine (Signaturzertifikate) oder genau eine (ENC, AUTH-Zertifikate) registrationNumber.

Das ProfessionItem enthält den UTF-8-kodierten String „Ärztin/Arzt“

[10] Ein Bsp.: Mittels Vorbefüllungsnachricht wird der Kammercode ‚078’ übertragen. Demzufolge ist als Bezeichner ‚Bezirksärztekammer Nordbaden’ in die Zertifikate zu personalisieren.

Der ProfessionOID enthält den OID für „Ärztin/Arzt“ wie in der folgenden Tabelle beschrieben:

| Karte | ProfessionOID | Name |
|---|---|---|
| HPC Version 2.1.1 Entwicklerkarte | 1.3.6.1.4.1.24796.4.11.1 | id-baek-at-namingAuthorityÄrzteschaft-Ärztin/Arzt |
| HPC Version 2.1.1 Testkarte | 1.3.6.1.4.1.24796.4.11.1 | id-baek-at-namingAuthorityÄrzteschaft-Ärztin/Arzt |
| HPCqsig | 1.3.6.1.4.1.24796.4.11.1 | id-baek-at-namingAuthorityÄrzteschaft-Ärztin/Arzt |
| HPC Version 2.3.0 alle Karten | 1.2.276.0.76.4.30 | Ärztin/Arzt |

Die registrationNumber enthält bei den ENC und AUTH-Zertifikaten die von der Ärztekammer bestätigte Telematik-ID.

Die Admission mit dem Berufsgruppenattribut „Ärztin/Arzt“ kann im Signaturzertifikat oder in einem dem Signaturzertifikat zugeordneten qualifizierten Attributzertifikat aufgenommen werden. Dies wird von der Bundesärztekammer festgelegt bzw. kann auch nachträglich geändert werden und wird den ZDA mitgeteilt. Außerdem kann auch bei gesetzter Admission im Signaturzertifikat ein Attributzertifikat ggf. mit einem anderen Attribut (z. B. Facharzt) ausgestellt werden. Dies ist im Ermessen der BÄK (Definition der Konfigurationsdaten, s. [baekConfigData]), der bestätigender Ärztekammer (Freigabe des Antrags) und des Antragstellers. Außerdem können durch die BÄK jederzeit neue OIDs und professionItems definiert werden (Konfigurationsdaten), die nach entsprechender Mitteilung von den ZDAs implementiert werden müssen.

Das Zertifikatsprofil für Attributzertifikate ist in [baekAttr] definiert. Die Admission-Extension wird in Zertifikaten für Ärztekammer-Mitarbeiter (zur Wahrnehmung von administrativen Funktionen), die keine Ärzte sind, nicht gesetzt.

### 4.9.7 BasicConstraints (2.5.29.19)

Die Extension kennzeichnet ein Root- oder CA- oder Cross-Zertifikat. Sie ist als „critical“ markiert. Sie wird bei allen Zertifikatstypen (s. auch Tabelle 2) gesetzt[11]. Der Parameter `pathLenConstraint` gibt die Anzahl hierarchisch nachfolgender CA-Zertifikate an.

ASN.1-Struktur:

```
BasicConstraints ::= SEQUENCE
{
      ca                      BOOLEAN DEFAULT FALSE
      pathLenConstraint       INTEGER (0..MAX) OPTIONAL
}
```

[11] Obwohl eine Nicht-Aufnahme der Extension bedeuten würde, dass es sich um ein nicht-CA-Zertifikat handelt, wird die Extension aus Kompatibilitätsgründen (Windows) auch in den User-Zertifikaten aufgenommen.

Für Root- CA- und Cross-Zertifikate wird der Parameter ca mit dem Wert TRUE gesetzt. Der Parameter `pathLenConstraint` wird nur bei CA-Zertifikaten mit dem Wert 0 gesetzt. Bei Root- und Cross-Zertifikate wird der Parameter `pathLenConstraint` nicht aufgenommen. Bei User-Zertifikaten wird CA=FALSE (kodiert als leere Sequence gemäß DER) ohne `pathLenConstraint` aufgenommen.

### 4.9.8 CRLDistributionPoints (2.5.29.31)

Die Extension gibt den CRL-Issuer an und die URL, die die CRL enthält. Sie ist „nicht-kritisch“.

ASN.1-Struktur:

```
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
}
```

Gültige Werte:

Im `fullName` wird die vollständige URL (LDAP oder HTTP oder beides, UTF-8 und URL-konform [RFC2253 und nachfolgende; URL-konforme Kodierung von „\“ ist „%5C“] kodiert, z.B. ä=%5CC3%5CA4) aufgeführt, unter welche die aktuelle CRL abgerufen werden kann (s. auch Tabelle 2). Die Felder `reasons` und `nameRelativeToCRLIssuer` werden nicht verwendet. Das Feld `cRLIssuer` muss laut [externCommonPKI] aufgenommen werden, da indirekte CRLs ausgestellt werden. Es muss den SubjectDN des CRL-Signers enthalten, exakt wie im CRL-Signer-Zertifikat kodiert (inkl. Reihenfolge in der ASN.1-Struktur).

Anmerkungen:

Der Wert einer CRL ist im SigG-Kontext sowie auch für Authentisierungs- und Verschlüsselungszertifikate, die für die e-Arztausweise den SigG-Anforderungen genügen müssen, zumindest für eine erste Überprüfung eines Zertifikates, zweifelhaft. Der Grund ist, dass ausgestellte Zertifikate, die nicht im Verzeichnisdienst freigeschaltet wurden, nicht in einer CRL aufgenommen werden können, obwohl sie ungültig sind. Eine Verweigerung zur Freischaltung kann z. B. entstehen, wenn der Empfänger die Chipkarte nie bekommen hat (jemand anders hat sie abgefangen) oder wenn die Transport-PIN gebrochen war. Solche Zertifikate sind mathematisch korrekt aber formal „nicht existent“ und somit ungültig. Solche

Signatur- und Attributzertifikate können auch nicht in der CRL aufgenommen werden, weil damit suggeriert würde, dass sie bis zur Sperrung gültig wären (eine rückwirkende Sperrung ist laut SigG nicht zulässig). Authentisierungs- und Verschlüsselungszertifikate müssen im Gegenteil in der zugehörigen CRL aufgenommen werden, wenn sie sich als ungültig herausstellen. Der Grund für die unterschiedliche Behandlung liegt darin, dass Signaturzertifikate stets zu einem Zeitpunkt in der Vergangenheit (Erstellungszeitpunkt der Signatur), AUTH und ENC-Zertifikate jedoch stets zum aktuellen Zeitpunkt geprüft werden.

Es wird somit hingewiesen, dass eine CRL-Prüfung nicht ausreicht, um die Gültigkeit eines Zertifikates zu einem bestimmten Zeitpunkt zu ermitteln. Eine OCSP-Prüfung ist erforderlich. Wenn allerdings ein Zertifikat einmal als „vorhanden“ über OCSP deklariert wurde, dann kann man künftig dafür auch eine CRL-Prüfung vornehmen.

Der CRL-Issuer kann sich im SigG-Kontext ändern. Dies ist z. B. möglich, wenn die Person, auf die das Zertifikat personalisiert ist, gestorben ist und somit auch nicht mehr unterschreiben kann. Laut SigG/SigV ist es nicht zulässig, ein bereits vergebenes Pseudonym (z. B. CRL-Signer 1:PN) einer zweiten Person zuzuordnen. Der DistinguishedName im Feld cRLSigner ist jedoch fest. Auch für ENC und AUTH-Zertifikaten ist es möglich, dass ein anderer CRL-Signer, als der im CRL-DP Eingetragene, die CRL unterschreiben muss. Dies ist dann der Fall, wenn ein Trustcenter aus Gründen der Hochverfügbarkeit zwei Standorte betreibt und Chipkarten/HSMs mit nicht-kopierbaren Schlüsseln einsetzt. Der zweite Standort muss dann einen anderen CRLSigner einsetzen als der erste Standort. Folgender Lösungsweg wird dafür sowohl für qualifizierte als auch für alle nicht-qualifizierten-Zertifikate definiert, gemäß Common-PKI Spezifikation [externCommonPKI] Part 9 S. 27ff:

Der Überprüfer einer CRL soll die CRL mittels dem URL im CRL-DP herunterladen und mit den in Common-PKI V2.0 beschriebenen Validierungsalgorithmen überprüfen. Wenn der CRLSigner, der die CRL unterschrieben hat, nicht identisch ist mit dem CRLSigner, der im Zertifikat eingetragen ist, dann soll überprüft werden, ob das Zertifikat (dessen Gültigkeit mit Hilfe der CRL geprüft wird) und der CRLSigner, der die CRL signiert hat, aus der selben Zertifizierungshierarchie stammen. Wenn dies der Fall ist und auch alle anderen Prüfschritte positiv verlaufen, dann soll die CRL als gültig betrachtet werden.

### 4.9.9 ExtendedKeyUsage (2.5.29.37)

Spezifische Anwendungen für bestimmte Zertifikatstypen können mit dieser Extension gekennzeichnet werden. Die Extension ist nicht kritisch und wird nur bei Authentisierungszertifikaten, OCSP-Signer- und TSS-Zertifikaten gesetzt.

Anmerkung: TSS-Zertifikate müssen qualifizierte Zertifikate sein. Sie werden von der Bundesnetzagentur ausgestellt, deshalb sind sie nicht Gegenstand dieses Dokuments. Ein Zeitstempeldienst wird für den Einsatz der e-Arztausweise vorausgesetzt.

ASN.1-Struktur:

```
ExtendedKeyUsage ::= SEQUENCE SIZE (1..MAX) OF KeyPurposeId
     KeyPurposeId ::= OBJECT IDENTIFIER
```

Gültige Werte:

- Für OCSP-Signer-Zertifikate wird als `KeyPurposeId` die OID `id-kp-OCSPSigning` {id-kp 9} ({1 3 6 1 5 5 7 3 9}) gesetzt.
- Für Authentisierungszertifikate werden als `KeyPurposeId` die OIDs `id-kp-clientAuth` {id-kp 2} ({1 3 6 1 5 5 7 3 2}) sowie `id-kp-emailProtection` {id-kp 4} ({1 3 6 1 5 5 7 3 4}) gesetzt.
- Für TSS-Zertifikate wird als `KeyPurposeId` die OID `id-kp-timeStamping` {id-kp 8} ({1 3 6 1 5 5 7 3 8}) gesetzt.

### 4.9.10 QCStatements (1.3.6.1.5.5.7.1.3)

Die Kennzeichnung des Signaturzertifikats als qualifiziertes Signaturzertifikat nach der entsprechenden EU-Direktive (1999/93/EC) erfolgt mit der QCStatements-Extension und dem entsprechenden OID (s. gültige Werte). Außerdem kann eine Begrenzung für finanzielle Transaktionen, die auch die Haftung des Zertifizierungsdiensteanbieters begrenzt, aufgenommen werden. Die Extension ist nicht-kritisch und wird ausschließlich bei den qualifizierten Signaturzertifikaten sowie bei den qualifizierten CA-Zertifikaten gesetzt.

ASN.1-Struktur:

```
QCStatements ::= SEQUENCE OF QCStatement

QCStatement ::= SEQUENCE
{
      statementId                ObjectIdentifier
      statementInfo              ANY DEFINDED BY statementId OPTIONAL
}
```

Gültige Werte:

Die Extension muss (für qualifizierte Signaturzertifikate und qualifizierte CA-Zertifikate) mindestens ein QCStatement mit dem OID id-etsi-qcs-QcCompliance {id-etsi-qcs 1} ({0 4 0 1862 1 1}) als `statementId` enthalten. Ein `statementInfo` wird nicht gesetzt.

Die Extension kann (für qualifizierte Signaturzertifikate aber nicht für qualifizierte CA-Zertifikate) zusätzlich ein weiteres QCStatement mit dem OID id-etsi-qcs-QcLimitValue {id-etsi-qcs 2} ({0 4 0 1862 1 2}) als `statementId` enthalten. `statementInfo` wäre dann QcEuLimitValue mit folgender ASN.1-Struktur:

```
QcEuLimitValue ::= MonetaryValue


 MonetaryValue::= SEQUENCE {
    currency Iso4217CurrencyCode,
    amount INTEGER,
    exponent INTEGER}
    -- value = amount * 10^exponent


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

```
-- Alphabetic or numeric currency code as defined in ISO 4217
-- It is recommended that the Alphabetic form is used
```

Gültige Werte:

Der Betrag (maximale Beschränkung für finanzielle Transaktionen, Haftungsgrenze des ZDA pro Unterschrift)[12] wird nach der Regel im ASN-1-Kommentar „`-- value = amount * 10^exponent`„ gebildet. Als `currency` muss der alphanumerische Code „EUR“ verwendet werden. Falls ein Betrag aufgenommen wurde, muss in der Restriction-Extension ein Text aufgenommen werden, der verdeutlicht, dass die eingetragene monetäre Limitierung nicht für Anwendungen nach SGB V §291a gilt.

### 4.9.11 Restriction (1 3 36 8 3 8)

Diese optionale Extension gibt Einschränkungen (nicht monetärer Natur) in der Anwendung des Zertifikates an. Sie ist nicht-kritisch.

ASN.1-Struktur:

```
RestrictionSyntax ::= DirectoryString
```

Der `DirectoryString` darf maximal 1024 bytes lang sein und muss UTF-8-kodiert sein.

Gültige Werte:

Der Text wird vom Antragsteller definiert. Bei fehlendem Text darf die (leere) Extension nicht gesetzt werden. Falls ein Text aufgenommen wurde, muss dazu ein weiterer Text automatisch angehängt werden, der verdeutlicht, dass die eingetragene Beschränkung nicht für Anwendungen nach SGB V § 291a gilt.

### 4.9.12 AdditionalInformation (1 3 36 8 3 15)

Diese optionale Extension gibt weitere Informationen (nicht einschränkender Natur) über die Verwendung des Zertifikates an. Sie ist nicht-kritisch.

ASN.1-Struktur:

```
AdditionalInformationSyntax ::= DirectoryString
```

Der `DirectoryString` darf maximal 2048 bytes lang sein und muss UTF-8-kodiert sein.

Gültige Werte:

In Abhängigkeit vom Herausgabemodell könnte der Inhalt der Extension beispielsweise lauten: “Zertifikat als Teil eines elektronischen Arztausweises, herausgegeben durch die zuständige Landesärztekammer”. Bei fehlendem Text darf die (leere) Extension nicht gesetzt

[12] Über den Endnutzervertrag o. ä. muss sichergestellt sein, dass ggf. angegebene Haftungsgrenzen und Beschränkungen sich nicht auf Anwendungen nach § 291a SGB V beziehen.

werden. Derzeit wird diese Extension nicht verwendet, sie ist für zukünftige Anwendungen reserviert.

### 4.9.13 AuthorityInfoAccess (1.3.6.1.5.5.7.1)

Die Extension AuthorityInfoAccess enthält Angaben über die Quelle von Statusinformationen für die Validierung des Zertifikates. Für Statusinformationen für die e-Arztausweise wird das OCS-Protokoll verwendet. Die Extension ist nicht-kritisch und enthält die URL des zuständigen OCSP-Responders.

ASN.1-Struktur:

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

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

Gültige Werte:

Als `accessMethod` wird `id-ad-ocsp` {1 3 6 1 5 5 7 48 1} gesetzt. In der `accessLocation` wird die URL des für das Zertifikat zuständigen OCSP-Responders aufgenommen (URL-konform kodiert).

### 4.9.14 ValidityModel (1.3.6.1.4.1.8301.3.5)

Die Extension `ValidityModel` gibt das Gültigkeitsmodell an, das von einem Client für die Zertifikatsprüfung und Signaturvalidierung verwendet werden muss. Der verwendete OID kennzeichnet das Kettenmodell. Die Extension ist nicht kritisch.

Der OID der Extension `ValidityModel` ist `id-validityModel` {1 3 6 1 4 1 8301 3 5}.

ASN.1-Struktur:

```
ValidityModel::= SEQUENCE
{
     validityModelId       OBJECT IDENTIFIER
     validityModelInfo     ANY DEFINED BY validityModelId OPTIONAL
}
```

Gültige Werte:

Als `validityModelId` wird der Wert für das Kettenmodell `id-validityModel-chain` {1 3 6 1 4 1 8301 3 5 1} für alle Verwaltungszertifikate (Zertifikate der Root-Instanz) sowie für die qualifizierten Signaturzertifikate festgelegt.

Verschlüsselungs- und Authentisierungszertifikate enthalten die Extension validityModel nicht, gleichwohl unterliegen sie dem Kompromissmodell (s. [baekValidityModel]).

Das Feld `validityModelInfo` wird nicht verwendet.

### 4.10 Das Feld `signatureAlgorithm`

Das Feld `signatureAlgorithm` entspricht dem Feld `signature`, wie im Abschnitt 4.4 „Erlaubte Signaturalgorithmen, Feld `Signature`“ beschrieben.

### 4.11 Das Feld `signatureValue`

Die Signatur auf das `tbsCertificate` wird im Feld `signatureValue` als BIT STRING aufgenommen.

# 5 Literatur

[externRFC5280] Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, Cooper, D.; Santesson, S.; Farrell, S.; Boeyen, S.; Housley, R.; Polk, W., Request for Comments (RFC) 5280, May 2008.

[externCommonPKI] Common PKI Specification for Interoperable Applications, T7 & TeleTrusT, Version 2.0, 20.01.2009

[externAlgCat] Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung (Übersicht über geeignete Algorithmen) vom 17.11.2008, veröffentlicht am 27.01.2009 im Bundesanzeiger Nr. 13 S. 346, Bundesnetzagentur. http://www.bundesnetzagentur.de/media/archive/15549.pdf.

[baekAttr] Zertifikatsprofile für X.509 Attributzertifikate; Version 2.3.2; 12.05.11

[baekValidityModel] Gültigkeitsmodell der elektronischen Arztausweise und Laufzeit der Zertifikate; Version 2.3.1; 29.05.09

[baekVerzD] Verzeichnisdienstkonzept; Version 2.3.2; 12.05.11

[baekConfigData] Konfigurationsdaten für die PKI der elektronischen Arztausweise, Version 2.3.4; 12.05.11

[baekXML] XML-Schema XArzt (Basistypen, Baukasten, Nachrichten); Version 1.1; 12.02.08

[leoGemPolicy] Gemeinsame Policy für die Herausgabe der HPC; Version 0.9.3w2; 03.03.06, Bundesärztekammer, Bundespsychotherapeutenkammer, Bundeszahnärztekammer, Kassenzahnärztliche Bundesvereinigung, Werbe- und Vertriebsgesellschaft Deutscher Apotheker mbH