Bundesamt
für Sicherheit in der
Informationstechnik

Deutschland
Digital•Sicher•BSI•

BSI Technische Richtlinie TR-02103: X.509 Zertifikate und Zertifizierungspfadvalidierung

Version 1.0

Stand: 29.09.2020

Änderungshistorie

Version	Datum	Beschreibung
1.0	September 2020	Erstellung

Bundesamt für Sicherheit in der Informationstechnik Postfach 20 03 63
53133 Bonn
Tel.: +49 22899 9582-0
E-Mail: cpt@bsi.bund.de
Internet: https://www.bsi.bund.de

Inhalt

1 Einleitung

X.509 Zertifikate bilden einen wichtigen Pfeiler der IT-Sicherheit, indem sie die Realisierung von Public-Key-Infrastrukturen ermöglichen. Die hier gegebenen Hinweise stellen die wesentlichen Punkte heraus, auf die bei dem Einsatz von X.509-Zertifikaten geachtet werden muss.

Die wesentliche Funktion eines X.509 Zertifikats ist es, den öffentlichen Schlüssel eines Inhabers als authentisch und zu dem Inhaber zugehörig verifizierbar zu machen, indem über den sogenannten Zertifizierungspfad eine kryptographische Verbindung zu einem Vertrauensanker hergestellt werden kann. Diese Verbindung besteht darin, dass jedes ausgestellte Zertifikat im Pfad mit dem öffentlichen Schlüssel des vorhergehenden verifiziert werden kann. Dabei steht am Anfang der als Vertrauensanker fungierende öffentliche Schlüssel, und am Ende das Zertifikat, für welches die Vertrauensprüfung durchgeführt wird. Des Weiteren sind in jedem Zertifikat wichtige Informationen enthalten, die neben Verweisen auf den Inhaber den Verwendungszweck und die Gültigkeit des Zertifikats einschränken. Da die zugehörigen Felder ebenfalls in die Berechnung der vom Aussteller erzeugten kryptographischen Signatur des Zertifikats einfließen, sind diese Werte im Rahmen der Zertifikatsprüfung ebenfalls vertrauenswürdig.

Die Ausstellung von Zertifikaten obliegt sogenannten „Certification Authorities“ oder kurz „CAs“. Eine CA stellt sicher, dass die Daten, die in ein Zertifikat einfließen, der Richtigkeit entsprechen, bevor das Zertifikat ausgestellt wird. Sie versieht das Zertifikat mit einer Signatur, die mittels des Schlüssels erstellt wird, der ihrem eigenen Zertifikat, dem sogenannten CA-Zertifikat, zugeordnet ist. CAs können auch hierarchisch gegliedert sein. Eine CA, die keine weitere CA über sich hat, wird als „Wurzel-“ oder „Root-CA“ bezeichnet. Ein solches Zertifikat trägt eine Selbstsignatur und fungiert als Vertrauensanker, der selbst nicht überprüft werden kann.

Des Weiteren ist es vorgesehen, dass ein Zertifikat während des in selbigem ausgewiesenen Gültigkeitszeitraums für ungültig erklärt werden kann. Dieser Vorgang wird als Sperrung oder Revokation bezeichnet (Englisch: „Revocation“). Technisch wird dieser Vorgang vollzogen, indem der Aussteller des zu sperrenden Zertifikats eine neue Version seiner Sperrliste (Englisch: „Certificate Revocation List“, kurz „CRL“) erstellt. Diese Liste enthält eindeutige Referenzen auf gesperrte Zertifikate dieses Ausstellers und ist wiederum mit einer kryptographischen Signatur versehen. Die Signatur ist entweder direkt mit dem Ausstellerzertifikat verifizierbar, oder aber mit einem mit dem Ausstellerzertifikat verifizierbaren dedizierten Sperrlistenunterzeichnerzertifikat (CRL Signer). Eine weitere Möglichkeit ist es, die Sperrung eines Zertifikats über das Online Certificate Status Protocol (OCSP) zu prüfen. Hierbei beantwortet ein sogenannter OCSP-Responder Anfragen nach einem Zertifikat mit einer kryptographisch signierten Antwort, welche den aktuellen Revokationsstatus des Zertifikats wiedergibt.

Die in diesem Dokument enthaltenen Hinweise beziehen sich auf die Wahl der Zertifikatsinhalte bei deren Erstellung und die korrekte und sichere Prüfung der Gültigkeit eines Zertifikats in bestimmten Anwendungskontexten. Diese Hinweise orientieren sich an [RFC 5280] und anderen anwendungsspezifischen Spezifikationen, aber gehen teilweise über diese hinaus. Wo immer Anforderungen aus [RFC 5280] oder anderen allgemeinverbindlichen Spezifikationen abgebildet werden, ist dies durch die Verwendung der für RFCs spezifischen Schlüsselworte MUST, SHALL, SHOULD, MAY, RECOMMENDS [RFC 2119] erkennbar. Alternativ wird auf solche Anforderungen als „Vorgaben“ Bezug genommen. Von dem vorliegenden Dokument darüberhinausgehend gemachte Empfehlungen werden durch das Wort „Empfehlung“ oder „empfohlen“ kenntlich gemacht. Wenn unter Verwendung der gleichen Worte auf Empfehlungen aus einem anderen Kontext Bezug genommen wird, so wird dies im Text entsprechend kenntlich gemacht. Wenn die Verwendung einer Funktionalität als vollständig abhängig von ihrem spezifischen Anwendungskontext ist, so wird diese als „optional“ gekennzeichnet.

Im Zusammenhang mit der Erzeugung und Verwendung von Zertifikaten gibt es noch eine ganze Reihe weiterer sicherheitsrelevanter Aspekte, die nicht Gegenstand des vorliegenden Dokuments sind. Dies umfasst z.B.

- die zuverlässige Identifikation von Entitäten als Voraussetzung der Zertifikatsausstellung,
- die sichere Erzeugung und Speicherung der zugehörigen geheimen Schlüssel,
- angemessene Authentisierung als Voraussetzung für deren Verwendung,
- sowie die Einhaltung von Empfehlungen bezüglich der Verwendung von kryptographischen Algorithmen.

Bezüglich des letzten Punktes sei auf [TR-02102-1], [TR-02102-2] und [TR-02102-3] verwiesen.

2 X.509 Zertifikate

In diesem Abschnitt werden sowohl Endbenutzer- als auch CA-Zertifikate behandelt. Zunächst wird der Aufbau von X.509 Zertifikaten erläutert. Anschließend werden die einzelnen Felder im Detail besprochen und die jeweils geltenden Vorgaben und Empfehlungen angegeben.

2.1 Struktur von X.509 Zertifikaten

Ein X.509 Public-Key-Zertifikat ist eine Datenstruktur, welche die Bindung zwischen einem öffentlichen Schlüssel und einer Entität darstellt. Die Entität ist der Besitzer des Schlüsselpaars. Die Entität kann z.B. eine Person, ein Client-System oder ein Internet Server sein.

Ein X.509-Zertifikat ist hierarchisch in einer Baumstruktur aufgebaut. Der Aufbau wird in einem ASN.1-Modul [X.680] beschrieben. ASN.1 ist eine abstrakte Beschreibungssprache für Datenstrukturen, die zunächst unabhängig von deren konkreter Darstellung ist. Ein spezifisches ASN.1-Modul definiert den syntaktischen Aufbau von Datenstrukturen auf Basis der vordefinierten Datentypen.

Für X.509 Zertifikate, Sperrlisten und OCSP-Antworten kommen die Distinguished Encoding Rules (DER) zur Anwendung, welche eindeutig festlegen, wie die Inhalte der in den ASN.1 Strukturen definierten Felder binär zu kodieren sind. Die DER Kodierung ist eine sogenannte Tag-Length-Value (TLV) Kodierung, in der jedes Feld im jeweiligen Kontext durch einen Tag-Wert identifiziert wird, gefolgt von einem Feld welches die Länge seines Wertes enthält, auf welches der Wert selbst folgt. Dabei existieren sogenannte „constructed“ Felder, bei denen der Value wiederum TLV Strukturen enthalten kann.

In einem X.509 Zertifikat ist die Struktur auf oberster Ebene wie folgt:

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

Die drei Hauptfelder haben demnach die Namen *tbsCertificate*, *signatureAlgorithm* und *signatureValue*. Die Angabe jeweils rechts daneben (z.B. *TBSCertificate*) referenziert einen bestimmten ASN.1 Datentyp, der in dem ASN.1 Modul definiert ist, und der wiederum eine eigene Struktur aufweist. Die in Großbuchstaben geschriebenen Datentypen, hier SEQUENCE und BIT STRING, verweisen auf allgemeine ASN.1 Datentypen, die unabhängig von dem ASN.1 Modul für X.509 Zertifikate definiert sind. Dabei bildet SEQUENCE wie auch SET einen constructed-Datentyp, der eine Aneinanderreihung von Feldern enthält.

Das Feld *tbsCertificate* bildet den eigentlichen Inhalt des Zertifikats, der durch die Signatur im *signatureValue* bestätigt wird. Die Signatur muss gemäß dem angegebenen *signatureAlgorithm* erstellt sein.

Der Aufbau des *tbsCertificate* ist wie folgt:

```
TBSCertificate ::= SEQUENCE {
   version         [0]  Version DEFAULT v1,
   serialNumber         CertificateSerialNumber,
   signature            AlgorithmIdentifier,
   issuer               Name,
   validity             Validity,
   subject              Name,
   subjectPublicKeyInfo SubjectPublicKeyInfo,
   issuerUniqueID  [1]  IMPLICIT UniqueIdentifier OPTIONAL, -- If present, version MUST be v2 or v3
   subjectUniqueID [2]  IMPLICIT UniqueIdentifier OPTIONAL, -- If present, version MUST be v2 or v3
   extensions      [3]  Extensions OPTIONAL -- If present, version MUST be v3 -- }
```

Es sei darauf hingewiesen, dass die Verwendung des Wortes „OPTIONAL“ in obiger Spezifikation eine spezielle Bedeutung im Kontext von ASN.1 hat. Und zwar zeigt dies an, dass ein Element nicht zwingend in der ASN.1 Struktur vorhanden sein muss. Dies ist nicht zu verwechseln mit der Kennzeichnung als „optional“ im Sinne der Empfehlungen des vorliegenden Dokuments. In den nachfolgenden Abschnitten werden die einzelnen Zertifikatsfelder und Zertifikatserweiterungen näher behandelt und die jeweils im Sinne dieses Dokuments geltenden Vorgaben und Empfehlungen angegeben.

2.2 Zertifikatsfelder

In den folgenden Unterabschnitten wird die Bedeutung der einzelnen Zertifikatsfelder erläutert und die Vorgaben und Empfehlungen für die einzelnen Zertifikatsfelder angegeben. Die für ein bestimmtes Feld geltenden Anforderungen und Besonderheiten werden jeweils in Form einer Tabelle zusammenfassend dargestellt.

2.2.1 version

Das Feld *version* gibt die Version der X.509-Spezifikation an, zu der das Zertifikat konform ist. Die derzeit möglichen Versionen sind v1, v2 und v3. Die Version wird als ein Integer dargestellt. Der Integer-Wert 0 steht für v1, der Wert 1 für v2 und der Wert 2 für v3. Die Inhalte des Zertifikats und die Version müssen zueinander passen. Ein Zertifikat mit Zertifikatserweiterungen muss die Version 3 haben, da Erweiterungen erst ab dieser Version möglich sind. Fast alle aktuell benutzten Zertifikate im Internet sind Version-3-Zertifikate. Zertifikate der Version 2 haben kaum Anwendung gefunden. Entsprechend wird empfohlen, nur Version-3-Zertifikate auszustellen.

Name	Version
Merkmale	MUST: v3 wenn Erweiterungen vorhanden. Empfehlung: Ausstellung nur von Version-3-Zertifikaten

Tabelle 1: Basisfeld Version – Übersicht

2.2.2 serialNumber

Die Zertifikatsseriennummer (*serialNumber*) ist ein Integer-Wert. Sie dient zusammen mit dem Aussteller (*issuer*) als Identifikationsmerkmal eines Zertifikats. Daher ist es entscheidend, dass bei der Ausstellung von Zertifikaten durch eine CA niemals die gleiche Seriennummer zweimal vergeben wird.

Bei der Vergabe der Seriennummer sind zwei Ansätze denkbar:

- aufsteigende Seriennummern oder
- zufällige Seriennummern.

Der erste Ansatz bietet eine einfachere Umsetzung, da eine CA bei der Vergabe einer neuen Seriennummer nur die zuletzt vergebene kennen muss. Bei der Verwendung von zufälligen Seriennummern muss sichergestellt werden, dass eine neu zu vergebende Seriennummer von dieser CA noch nicht vergeben wurde.

Zufällige Seriennummern bieten einen erhöhten Schutz gegen die Fälschung von Zertifikaten basierend auf dem Finden von Hashkollisionen für den im Signaturverfahren zur Anwendung kommenden Hashalgorithmus. Sofern hier die CA nur Hashalgorithmen aus der SHA-2 Familie verwendet, und nicht etwa SHA-1, ist diese zusätzliche Sicherheitsmaßnahme von nachrangiger Bedeutung.

Name	Seriennummer
Merkmale	MUST: Positive Ganzzahl.

	MUST: Kodierte Seriennummer darf maximal eine Länge von 20 Byte haben. MUST: Jede Seriennummer darf von einer CA (d.h. für einen bestimmten Wert von *issuer*) nur einmal vergeben werden.

Tabelle 2: Basisfeld Seriennummer – Übersicht

2.2.3 signature

Das Feld *signature* in *TBSCertificate* enthält nicht etwa, wie der Name nahelegt, eine Signatur, sondern ein Identifikationsmerkmal in Form eines *Object Identifiers* (OID) für den Signaturalgorithmus, mit dem das Zertifikat signiert ist. Die relevanten in der Praxis eingesetzten Algorithmen sind RSA, DSA und ECDSA. Die zu diesen Signaturalgorithmen jeweils zugeordneten OIDs können [RFC 5758], [RFC 4055] und [RFC 3279] entnommen werden. Der Wert muss identisch sein mit demjenigen in dem Feld *signatureAlgorithm*. Außer dass dieses Feld Teil des *TBSCertificate* ist und somit in die Zertifikatssignatur einfließt und das Feld s*ignatureAlgorithm* nicht, gibt es zwischen diesen Feldern keinen Unterschied.

Name	Signatur
Merkmale	MUST: identisch zu Feld *signatureAlgorithm* außerhalb des TBS-Teils.

Tabelle 3: Basisfeld Signatur – Übersicht

2.2.4 issuer

Das Feld *issuer*, oftmals auch als „issuer-DN" bezeichnet, muss den Namen des Ausstellers, d.h. der CA, in Form eines nicht-leeren Distinguished Name (DN) enthalten. Der issuer-DN ist ein Distinguished Name in Form einer SEQUENCE von RDNs, wobei jeder RDN eine Menge (SET) von *AttributeTypeAndValue*-Paaren ist, welche typischerweise aus genau einem Element besteht, d.h. aus genau einer Sequenz von OID und Value. Weniger gebräuchlich, aber ebenfalls möglich ist ein Set mit mehreren *AttributeTypeAndValue*-Elementen (Multi-Valued RDN). [RFC 5280] definiert eine Reihe von Attributen, welche von Implementierungen unterstützt werden müssen. Diese sind in Tabelle 4 aufgeführt. Darüber hinaus unterstützt [RFC 5280] die Verwendung von weiteren Attributen, welche jedoch nicht verpflichtend für Implementierungen sind.

Attribut	**OID**
country (C)	2.5.4.6
organization (O)	2.5.4.10
organizational unit (OU)	2.5.4.11
distinguished name qualifier	2.5.4.44
state or province name (ST)	2.5.4.8
common name (CN)	2.5.4.3
serialNumber	2.5.4.5
domain component (DC)	0.9.2342.19200300.100.1.25

Tabelle 4: Standardattribute im Distinguished Name

Als String-Datentypen für die Felder im Distinguished Name sollten nur *PrintableString* oder *UTF8String* verwendet werden, da [RFC 5280] konforme Implementierungen nur diese unterstützen müssen. Die einzige Ausnahme ist gegeben, wenn aus Gründen der Abwärtskompatibilität zu bestehenden Zertifikaten andere Kodierungen verwendet werden müssen.

Bei der Kodierung des *issuer* Feldes ist zu beachten, dass dieser binär identisch sein soll mit dem im Ausstellerzertifikat kodierten Feld *subject*. Zwar sieht [RFC 5280] in Abschnitt 7.1 im Prinzip eine Vergleichsmethode vor, bei der Abweichungen in Form von unterschiedlichen Anzahlen von Leerzeichen sowie Unterschiede bei Groß- und Kleinschreibung ignoriert werden, jedoch wird empfohlen, bei der Ausstellung von Zertifikaten keinen Gebrauch davon zu machen. Denn da diese Art des Vergleichs nicht bindend umgesetzt werden muss, besteht sonst die Gefahr, dass Implementierungen, die sich an [RFC 3280] orientieren, die Zertifikate nicht verarbeiten können. Dies liegt daran, dass [RFC 3280] nur den binären Vergleich vorsieht. Ferner setzt beispielsweise OCSP [RFC 6960] eine eindeutige binäre Kodierung des Ausstellernamens voraus.

Name	Aussteller (issuer-DN)
Merkmale	MUST: Es darf nicht leer sein. MUST: Verwendung von *PrintableString* oder *UTF8String* für die Kodierung. Zwei Ausnahmen liegen vor (siehe Abschnitt 4.1.2.4 [RFC 5280]). Empfehlung: Die Kodierung sollte mit der Kodierung des subject-DN im Zertifikat des Ausstellers binär übereinstimmen.

Tabelle 5: Basisfeld Aussteller (issuer) - Übersicht

2.2.5 validity

Das Feld *validity* bestimmt den Zeitraum, in dem das Zertifikat gültig ist, sofern es nicht zurückgezogen wurde. Das Feld besteht aus den Komponenten *notBefore* und *notAfter*, die beide jeweils einen Zeitpunkt angeben. Der Zeitraum beginnt bei *notBefore*. Nach *notAfter* ist das Zertifikat nicht mehr gültig.

Die Werte für beide Zeitpunkte sind für die Zeitzone Greenwich Mean Time (GMT) angegeben. Die Zeitpunkte müssen sekundengenau angegeben werden. Sekundenbruchteile dürfen nicht angegeben werden.

Für die Kodierung der Zeitpunkte werden *UTCTime* und *GeneralizedTime* verwendet. Dabei muss *UTCTime* für Zeitpunkte vor 2050 verwendet werden, für spätere Zeitpunkte muss *GeneralizedTime* verwendet werden. Bei *UTCTime* wird der Zeitpunkt in der Zeichenkette *YYMMDDhhmmssZ* kodiert, wobei jeder Buchstabe für eine Dezimalziffer steht. Die jeweiligen durch Doppelbuchstaben dargestellten Felder stehen der Reihe nach von links nach rechts für das Jahr, den Monat, den Tag im Monat, die Stunden, Minuten und Sekunden. Das „Z“ steht für „Zulu“, welches die GMT Zeitzone repräsentiert. Alle in *UTCTime* kodierten Zeitpunkte sind in der Zeitspanne vom 1.1.1950 bis zum 31.12.2049. *GeneralizedTime* wird kodiert als *YYYYMMDDhhmmssZ*, im Gegensatz zu *UTCTime* also mit 4 Ziffern für das Jahr.

Folgendes Beispiel verdeutlicht diese Regeln. Bei der Ausstellung eines Zertifikats welches vom 01.01.2017 bis zum 01.01.2050 gültig sein soll, muss *notBefore* in *UTCTime* als „170101000000Z“ kodiert werden, und *notAfter* in *GeneralizedTime* als „20500101235959Z“.

Name	Gültigkeit (notBefore, notAfter)
Merkmale	MUST: UTCTime (YYMMDDhhmmssZ) kodiert bis zum Jahr 2049 (inklusive), GeneralizedTime (YYYYMMDDhhmmssZ) kodiert ab dem Jahr 2050 (inklusive). MUST: Greenwich Mean Time (Zulu) muss verwendet werden.

Tabelle 6: Basisfeld Gültigkeit – Übersicht

2.2.6 subject

Das Feld *subject* enthält den Namen des Inhabers des Zertifikats. Der Wert des Feldes ist ein Distinguished Name (DN) wie beim Feld *issuer*, das in Abschnitt 2.2.4 beschrieben ist, somit gelten die gleichen Regeln bezüglich der verwendbaren Attribute. Das Feld wird auch als „subject-DN“ bezeichnet.

Eine CA darf nicht zwei Zertifikate mit dem gleichen Inhalt von *subject* für zwei verschiedene Entitäten ausstellen. Nur wenn die Zertifikate für die gleiche Entität ausgestellt werden, ist das erlaubt. Dies kommt zum Beispiel bei der Erneuerung eines abgelaufenen Zertifikates vor.

Im Gegensatz zum Aussteller (*issuer*) darf das Inhaber-Feld in End-Entity-Zertifikaten leer sein, wenn der Verweis auf den Inhaber in anderen Feldern des Zertifikats enthalten ist. Beispielsweise kann ein DNS Name oder eine E-Mail-Adresse in der Subject Alternative Name Zertifikatserweiterung enthalten sein. Wenn es beabsichtigt ist, dass dieser Name die einzige Referenz auf den Zertifikatsinhaber ist, dann kann der subject-DN leer bleiben. In der Praxis ist dieser Fall jedoch sehr selten.

Name	Inhaber
Merkmale	MUST: Bei CA-Zertifikaten oder CRL Ausstellern muss das Feld einen Namen enthalten. MUST: Verwendung von *PrintableString* oder *UTF8String* für die Kodierung. Drei Ausnahmen liegen vor (siehe Abschnitt 4.1.2.6 [RFC 5280]). MUST: Die CA darf nicht denselben Wert für zwei verschiedene Entitäten vergeben.

Tabelle 7: Basisfeld Inhaber (subject) – Übersicht

2.2.7 subjectPublicKeyInfo

Das Feld *subjectPublicKeyInfo* enthält den öffentlichen Schlüssel des Inhabers des Zertifikats. Die Struktur *SubjectPublicKeyInfo* hat zwei Komponenten, nämlich *algorithm* vom Typ *AlgorithmIdentifier* und *subjectPublicKey* vom Typ BIT STRING. Der *AlgorithmIdentifier* beschreibt mit einem OID, um welche Art von Schlüssel es sich handelt, z.B. um einen RSA-Schlüssel oder einen EC-Schlüssel. Der *AlgorithmIdentifier* kann weitere Parameter enthalten, die bei der Verwendung des öffentlichen Schlüssels notwendig sind. Der *subjectPublicKey* enthält den kodierten öffentlichen Schlüssel.

Es besteht kein Zusammenhang zwischen dem Schlüssel des Inhabers und dem des Ausstellers. So kann beispielsweise ein Zertifikat einen EC-Schlüssel enthalten und mit einem RSA-Schlüssel signiert sein.

Name	Schlüssel
Merkmale	-

Tabelle 8: Basisfeld Schlüssel - Übersicht

2.2.8 issuerUniqueID und subjectUniqueID

Die laut ASN.1 Beschreibung optionalen Felder *issuerUniqueID* und *subjectUniqueID* gibt es erst ab Version 2 der X.509-Zertifikate. Sie waren vorgesehen, um eine eindeutige Referenz auf den Aussteller oder den Inhaber in das Zertifikat zu integrieren. Entsprechend den aktuellen Vorgaben aus [RFC 5280] darf keines dieser beiden Felder mehr in Zertifikaten gesetzt werden.

Name	issuerUniqueID

Merkmale	MUST: Es darf nicht verwendet werden.

Tabelle 9: issuerUniqueID – Übersicht

Name	subjectUniqueID
Merkmale	MUST: Es darf nicht verwendet werden.

Tabelle 10: subjectUniqueID – Übersicht

2.3 Zentrale Zertifikatserweiterungen

In diesem Abschnitt werden diejenigen Zertifikatserweiterungen vorgestellt, welche grundsätzlich in jedes Zertifikat aufgenommen werden sollten. Ausnahmen bilden lediglich die Subject Alternative Name und Extended Key Usage, die in CA-Zertifikaten typischerweise nicht vorkommen. Es wird empfohlen, diese beiden Erweiterungen nur in End-Entity-Zertifikaten zu verwenden.

2.3.1 Basic Constraints

Die Basic Constraints Erweiterung wird dazu verwendet, um ein Zertifikat als CA-Zertifikat oder End-Entity-Zertifikat erkennbar zu machen. Ein Version-3-Zertifikat darf nur dann für die Überprüfung von Signaturen anderer Zertifikate benutzen werden, wenn diese Erweiterung vorhanden ist und das Zertifikat als CA-Zertifikat markiert.

Name	Basic Constraints
OID	2.5.29.19
Kritisch	MUST: als kritisch markiert in CA-Zertifikaten
Spezifikation	Abschnitt 4.2.1.9 in [RFC 5280]
Merkmale	MUST: Muss in CA-Zertifikaten vorhanden sein. MUST: In CA-Zertifikaten muss das Feld *cA* vorhanden sein und den Wert *true* haben.

Tabelle 11: Basic Constraints Erweiterung – Übersicht

2.3.2 Key Usage

In dieser Erweiterung werden die erlaubten kryptographischen Verwendungszwecke des im Zertifikat ausgewiesenen Schlüssels festgelegt. Die hier zu setzenden Werte sind abhängig von der intendierten Verwendung des Zertifikats. Die möglichen Verwendungszwecke sind:

- *digitalSignature*: Verifikation von digitalen Signaturen mit dem öffentlichen Schlüssel für allgemeine Objekte, aber nicht für Zertifikate oder Sperrlisten.
- *nonRepudiation*: Verifikation von digitalen Signaturen mit dem öffentlichen Schlüssel für allgemeine Objekte, aber nicht für Zertifikate oder Sperrlisten. Die Entität, welche diese Signatur berechnet hat, kann nicht abstreiten, diese Signatur geleistet zu haben. Dieser Wert wird öfters in qualifizierten Zertifikaten gesetzt.
- *keyEncipherment*: Verwendung des öffentlichen Schlüssels zur Verschlüsselung von Schlüsseln, die wiederum zur Datenverschlüsselung eingesetzt werden (hybride Verschlüsselung).

- *dataEncipherment*: Verwendung des öffentlichen Schlüssels zur direkten Verschlüsselung von Daten (ohne Verwendung eines hybriden Verfahrens).
- *keyAgreement*: Verwendung des öffentlichen Schlüssels innerhalb eines Schlüsseleinigungsverfahrens.
- *keyCertSign*: Verwendung des öffentlichen Schlüssels zur Verifikation von Signaturen von Zertifikaten.
- *cRLSign*: Verwendung des öffentlichen Schlüssels zur Verifikation von Signaturen von Sperrlisten.
- *encipherOnly*: Nach [RFC 5280] nur von Bedeutung, wenn *keyAgreement* gleichzeitig gesetzt ist. In diesem Dokument wird die Empfehlung gegeben, diesen Schlüsselverwendungszweck aufgrund seiner nicht eindeutigen Interpretierbarkeit nicht zu verwenden.
- *decipherOnly*: Nach [RFC 5280] nur von Bedeutung, wenn *keyAgreement* gleichzeitig gesetzt ist. In diesem Dokument wird die Empfehlung gegeben, diesen Schlüsselverwendungszweck aufgrund seiner nicht eindeutigen Interpretierbarkeit nicht zu verwenden.

Name	Key Usage
OID	2.5.29.15
Kritisch	SHOULD: kritisch
Spezifikation	Abschnitt 4.2.1.3 in [RFC 5280]
Merkmale	MUST: Für CA-Zertifikate (Wert: *keyCertSign*) und CRL-Aussteller (Wert: *cRLSign*). MUST: Wenn die Erweiterung im Zertifikat vorkommt, ist mindestens eine explizite Verwendung gesetzt. Empfehlung: Weder *encipherOnly* noch *decipherOnly* sollten gesetzt werden.

Tabelle 12: Key Usage Erweiterung – Übersicht

2.3.3 Extended Key Usage

Diese Erweiterung wird in End-Entity-Zertifikaten benutzt, um die erlaubten Verwendungen des Schlüssels ähnlich der Key Usage Erweiterung einzuschränken. Während in der Key Usage Erweiterung kryptographische Operationen festgelegt werden, werden in der Extended Key Usage für den Schlüssel erlaubte Anwendungsarten angegeben, wie etwa dessen Nutzung zur Authentisierung von TLS Clients oder Servern, oder zur Absicherung von E-Mail Nachrichten. Das Zertifikat darf von Anwendungen nur für diejenigen Nutzungsarten, die in dieser Erweiterung spezifiziert sind, verwendet werden.

Die unterschiedlichen Verwendungen werden jeweils durch einen OID identifiziert. Man kann auch eigene Verwendungen definieren. Der spezielle Wert *anyExtendedKeyUsage* mit OID 2.5.29.37.0 kann, ggf. in Kombination mit anderen OIDs, benutzt werden, wenn keine Einschränkungen für die Verwendung vorhanden sein sollen.

Diese Erweiterung findet in End-Entity-Zertifikaten Verwendung, aber nicht in CA-Zertifikaten.

Mögliche vordefinierte Verwendungszwecke [RFC 5280], die in dieser Erweiterung gesetzt werden können, sind in der folgenden Liste wiedergegeben. Dabei wird als Name jeweils der letzte Bezeichner des zugeordneten OID verwendet.

- *serverAuth*: Authentisierung eines TLS Servers.
- *clientAuth*: Authentisierung eines TLS Clients.
- *codeSigning*: Signaturen in Software-Paketen verifizieren.
- *emailProtection*: E-Mail-Signaturen verifizieren und/oder E-Mails verschlüsseln.
- *timeStamping*: Signaturen von Zeitstempeln verifizieren.
- *OCSPSigning*: Signaturen von OCSP-Servern verifizieren.

Ferner besteht die Möglichkeit, unter Verwendung proprietärer OIDs eigene Verwendungszwecke zu definieren.

Name	Extended Key Usage
OID	2.5.29.37
Kritisch	Darf sowohl als kritisch als auch als nicht-kritisch markiert sein.
Spezifikation	Abschnitt 4.2.1.12 in [RFC 5280]
Merkmale	MUST: Die eingetragenen Verwendungen müssen die intendierten Schlüsselverwendungszwecke vollständig wiedergeben.

Tabelle 13: Extended Key Usage Erweiterung – Übersicht

2.3.4 Subject Alternative Name

Diese Erweiterung, manchmal abgekürzt als „SAN“, fügt dem Zertifikat weitere Namen des Inhabers hinzu. Beispiele sind die E-Mail-Adresse des Inhabers, um E-Mail-Verschlüsselung und -Signatur zu unterstützen, und der Domain-Name eines Servers für TLS. Die ausstellende CA muss sich davon überzeugen, dass die SAN-Einträge im ausgestellten Zertifikat tatsächlich dem Zertifikatsinhaber gehören und unter seiner Kontrolle sind. Zum Beispiel muss eine CA, die ein Zertifikat mit einem Domain-Name-Eintrag in der SAN ausstellt, überprüfen, ob der Antragsteller tatsächlich der Domain-Inhaber ist.

Name	Subject Alternative Name (SAN)
OID	2.5.29.17
Kritisch	MUST: kritisch, wenn leeres *subject* SHOULD: sonst nicht kritisch
Spezifikation	Abschnitt 4.2.1.6 in [RFC 5280]
Merkmale	MUST: Die individuellen Werte und deren Kodierung müssen konform zu den entsprechenden Spezifikationen sein. MUST: Der Wert der Erweiterung darf nicht leer sein.

Tabelle 14: Subject Alternative Name Erweiterung – Übersicht

2.3.5 CRL Distribution Points

Diese Erweiterung verweist auf Bezugsquellen für Sperrinformationen, unter denen ein Benutzer Sperrlisten für das Zertifikat abrufen kann. Die Erweiterung kann dabei auf mehrere Speicherorte verweisen.

Ein einzelner Verweis auf einen Speicherort enthält potentiell drei Felder: *distributionPoint*, *reasons* und *cRLIssuer*. Das Feld *distributionPoint* enthält die Adresse einer Sperrliste. *reasons* gibt die Sperrgründe an, die von der CRL abgedeckt werden. Die Abwesenheit dieses Feldes impliziert, dass die an dem angegebenen Speicherort hinterlegte Sperrliste die gesperrten Zertifikate unabhängig von dem jeweiligen Sperrgrund enthält. Durch die explizite Angabe von *reasons* ist eine Segmentierung der Sperrgründe möglich, so dass eine Applikation zur Bestimmung des Revokationsstatus eines Zertifikats mehrere Sperrlisten kombinieren muss, um die Abdeckung aller Sperrgründe zu erreichen. Es wird empfohlen *reasons* nicht zu verwenden. In jedem

Fall muss in dieser Erweiterung auf mindestens eine Sperrliste verwiesen werden, die alle Sperrgründe abdeckt.

cRLIssuer gibt den Aussteller der Sperrliste an. Dieses Feld muss genau dann gesetzt sein, wenn deren Aussteller nicht mit dem des geprüften Zertifikats übereinstimmt. In diesem Fall wird die Sperrliste als indirekte Sperrliste bezeichnet. Der Inhalt des Feldes muss binär mit dem Inhalt des Feldes *subject* im Zertifikat des Ausstellers der Sperrliste übereinstimmen. Falls es sich um eine direkte Sperrliste handelt, d.h. der Aussteller der Sperrliste und der des geprüften Zertifikats identisch sind, darf *cRLIssuer* nicht gesetzt werden.

Die Verweise auf die Verteilungspunkte müssen konform zu den entsprechenden Spezifikationen (z.B. HTTP [RFC 2585] und LDAP [RFC 4516]) sein.

Name	CRL Distribution Points
OID	2.5.29.31
Kritisch	SHOULD: nicht kritisch
Spezifikation	Abschnitt 4.2.1.13 in [RFC 5280]
Merkmale	RECOMMENDS: Anwesenheit in Zertifikaten. Empfehlung: Nichtverwendung von *reasons*. MUST: Es ist nicht erlaubt, diese Erweiterung leer zu lassen oder nur das Feld *reasons* zu setzen, mindestens ein weiteres Feld muss gesetzt werden. MUST: Bei indirekten Sperrlisten muss der Name des Ausstellers der CRL im Feld *cRLIssuer* eingetragen werden. MUST: Bei direkten Sperrlisten darf der Name des Ausstellers nicht eingetragen werden. In diesem Fall muss der Verteilungspunkt der CRL eingetragen werden. MUST: Die Kodierung des Namens des Ausstellers (*cRLIssuer*) muss mit der im Feld *issuer* der CRL übereinstimmen. MUST: Mindestens ein in dieser Erweiterung enthaltener Distribution Point verweist auf einer CRL, welche alle Sperrgründe abdeckt.

Tabelle 15: CRL Distribution Points Erweiterung – Übersicht

2.3.6 Certificate Policies

In dieser Erweiterung werden die Richtlinien und Verweise auf Certification Practice Statements (siehe auch [RFC 3647]) oder Hinweistexte gespeichert. Richtlinien werden durch OIDs referenziert. Es gibt die spezielle Richtlinie *anyPolicy* mit dem OID 2.5.29.32.0. Diese Richtlinie stellt einen Platzhalter für beliebige Richtlinien dar. Ihr Vorkommen in einem CA-Zertifikat bedeutet, dass explizit keine Einschränkungen für im Pfad nachfolgende Zertifikate festgesetzt werden.

Eine Zertifikatsrichtlinie (engl. „certificate policy“) definiert die Regeln, nach denen ein Zertifikat ausgestellt, verwaltet und benutzt wird. Dies beinhaltet auch wesentliche Eigenschaften der Ausstellungspraxis, wie beispielsweise die Prozeduren zur Identifikation von Antragstellern und die Sicherheitsmaßnahmen für den Schutz des geheimen Schlüssels des Ausstellers. Die Richtlinie wird in dem sogenannten Certification Practice Statement (CPS) festgehalten, einem Dokument, welches typischerweise öffentlich verfügbar gemacht wird. Dieses Dokument beschreibt neben der Zertifikatsrichtlinie auch die vom Aussteller getroffenen Maßnahmen zu deren Umsetzung. Die in [RFC 3647] für solche Dokumente vorgeschlagene Gliederung wird

meistens übernommen, da sie sehr umfangreich ist und alle wichtigen Aspekte abdeckt, die von einem Aussteller berücksichtigt werden müssen. Individuelle Änderungen und Erweiterungen sollten vorgenommen werden wo immer dies notwendig und sinnvoll ist.

Die Certificate Policies Erweiterung erlaubt es, für jede der aufgeführten Richtlinien neben deren OID im Feld *qualifier* auch einen Verweis auf einen Speicherort aufzunehmen, von dem das CPS abgerufen werden kann. Es wird in [RFC 5280], Abschnitt 4.2.1.4, allerdings empfohlen, für jede Policy nur das OID zu setzen und auf das Setzen von *qualifier* Feldern zu verzichten, um eine größtmögliche Interoperabilität zu gewährleisten. Falls *qualifier* verwendet werden, so dürfen nur die in Abschnitt 4.2.1.4 von [RFC 5280] spezifizierten verwendet werden.

Name	Certificate Policies
OID	2.5.29.32
Kritisch	SHOULD: kritisch
Spezifikation	Abschnitt 4.2.1.4 in [RFC 5280]
Merkmale	MUST: Es darf nicht zweimal der gleiche OID eingetragen werden. RECOMMENDS: *qualifier* sollten nicht verwendet werden.

Tabelle 16: Certificate Policies Erweiterung - Übersicht

2.3.7 Authority Key Identifier

Diese Erweiterung, abgekürzt mit „AKI“, dient dazu, Anwendungen die Konstruktion des Zertifizierungspfades zu erleichtern. Sie enthält Informationen, mit denen das Ausstellerzertifikat des zu prüfenden Zertifikats zweifelsfrei bestimmt werden kann.

Der Wert dieser Erweiterung kann bis zu drei Komponenten haben:

- *keyIdentifier*: Ein aus dem öffentlichen Schlüssel des Ausstellers abgeleiteter Prüfwert. Dieser ist identisch mit dem Wert der Subject Key Identifier Erweiterung im Ausstellerzertifikat.
- *authorityCertIssuer*: issuer-DN des Ausstellerzertifikats.
- *authorityCertSerialNumber*: Die Seriennummer des Ausstellerzertifikats.

Der Schlüssel des Ausstellers kann entweder über den *keyIdentifier* oder das Paar issuer-DN und Seriennummer ermittelt werden.

Eine standardkonforme CA muss die AKI-Erweiterung mit einem *keyIdentifier* in alle von ihr ausgestellten Zertifikate eintragen, die nicht selbst-signiert sind.

Name	Authority Key Identifier (AKI)
OID	2.5.29.35
Kritisch	MUST: nicht kritisch
Spezifikation	Abschnitt 4.2.1.1 in [RFC 5280]
Merkmale	MUST: Pflicht für jedes Zertifikat (außer selbst-signierte).

Tabelle 17: Authority Key Identifier Erweiterung – Übersicht

2.3.8 Subject Key Identifier

Diese Erweiterung, abgekürzt mit „SKI“, enthält einen Prüfwert bzw. *keyIdentifier* des öffentlichen Schlüssels des Zertifikats. Der *keyIdentifier* in der SKI eines CA-Zertifikats muss mit dem *keyIdentifier* in der Authority Key Identifier Erweiterung (Abschnitt 2.3.7) der mittels diesem CA-Zertifikat ausgestellten Zertifikate übereinstimmen. Jedes CA-Zertifikat muss diese Erweiterung beinhalten. Für End-Entity-Zertifikate ist dies ebenfalls empfohlen, um die verschiedenen Zertifikate einer Entität anhand des öffentlichen Schlüssels identifizieren zu können.

Im Falle von CA-Zertifikaten wird mit dieser Erweiterung die Zertifizierungspfad-Konstruktion vereinfacht: Das CA-Zertifikat einer CA enthält einen SKI. Alle von ihr ausgestellten Zertifikate enthalten einen AKI (siehe Abschnitt 2.3.7), dessen Prüfwert mit demjenigen im SKI des Aussteller-Zertifikats übereinstimmt. Somit ist es einfach für Implementierungen, den Aussteller zweifelsfrei festzustellen.

In End-Entity-Zertifikaten vereinfacht diese Erweiterung für eine Applikation das Identifizieren von Zertifikaten, die einen bestimmten öffentlichen Schlüssel enthalten.

Name	Subject Key Identifier (SKI)
OID	2.5.29.14
Kritisch	MUST: nicht kritisch
Spezifikation	Abschnitt 4.2.1.2 in [RFC 5280]
Merkmale	MUST: Pflicht für jedes CA-Zertifikat. SHOULD: Sollte in End-Entity-Zertifikaten aufgenommen werden.

Tabelle 18: Subject Key Identifier Erweiterung – Übersicht

2.3.9 Authority Information Access

Diese Erweiterung unterstützt die folgenden zwei PKI-Vorgänge:

- Das Einholen einer Statusauskunft über das Zertifikat mittels OCSP. Zu diesem Zweck wird darin die URL eines OCSP-Servers eingebettet.
- Die Zertifizierungspfad-Konstruktion. In diesem Fall spezifiziert die Erweiterung Adressen, an denen Zertifikate abgerufen werden können, die vor dem vorliegenden Zertifikat in einem Zertifizierungspfad vorkommen.

Beide Methoden werden durch OIDs referenziert.

Es ist möglich, mehrere Adressen in der Erweiterung zu spezifizieren. Die Verweise auf die Adressen müssen konform zu den Spezifikationen [RFC 4516] und [RFC 7230] sein.

In der X.509-Spezifikation ist diese Erweiterung mit Hinweisen zu ihrer Anwendung (siehe Abschnitt 18.3.2.1 von [X.509/16]) referenziert.

Name	Authority Information Access (AIA)
OID	1.3.6.1.5.5.7.1.1
Kritisch	MUST: nicht kritisch

Spezifikation	Abschnitt 4.2.2.1 in [RFC 5280]
Merkmale	-

Tabelle 19: Authority Information Access Erweiterung – Übersicht

2.4 Alternative Zertifikatserweiterungen

In diesem Abschnitt werden Zertifikatserweiterungen behandelt, deren Verwendung in Zertifikaten unter bestimmten Umständen erwogen werden sollte.

2.4.1 Freshest CRL

Diese Erweiterung, manchmal auch bezeichnet als „Delta CRL Distribution Point“, verweist auf Verteilungspunkte für Delta-Sperrlisten. Sie entspricht der Erweiterung CRL Distribution Points (siehe Abschnitt 2.3.5) für Delta-Sperrlisten [RFC 5280]. Dabei handelt es sich um Sperrlisten, welche nur (bezogen auf eine als Referenz dienende vollständige Sperrliste) neu hinzugekommene, gesperrte Zertifikate enthalten. Somit können derartige Sperrlisten unter Umständen deutlich kürzer ausfallen als vollständige Sperrlisten.

Name	Freshest CRL
OID	2.5.29.46
Kritisch	MUST: nicht kritisch
Spezifikation	Abschnitt 4.2.1.15 in [RFC 5280]
Merkmale	-

Tabelle 20: Freshest CRL Erweiterung – Übersicht

2.4.2 Issuer Alternative Name

Diese Erweiterung entspricht der SAN-Erweiterung (siehe Abschnitt 2.3.4) für den Aussteller des Zertifikats. Ihre Verwendung ist dann sinnvoll, wenn im Anwendungskontext die Referenzierung eines Ausstellers über einen der Typen von alternativen Namen, die in dieser Erweiterung gesetzt werden können, sinnvoll bzw. notwendig ist.

Name	Issuer Alternative Name (IAN)
OID	2.5.29.18
Kritisch	SHOULD: nicht kritisch
Spezifikation	Abschnitt 4.2.1.7 in [RFC 5280]
Merkmale	Siehe Merkmale von SAN (Tabelle 14).

Tabelle 21: Issuer Alternative Name Erweiterung – Übersicht

2.5 Proprietäre Zertifikatserweiterungen

Nach [RFC 5280] ist auch die Verwendung eigener, proprietärer Zertifikatserweiterungen möglich. Dabei wird das generelle Format einer Extension mit den Feldern *extnId* (vom Typ *OID*), *critical* (vom Typ BOOLEAN) und *extnValue* beibehalten. Für *extnId* muss dazu ein passender, freier OID registriert werden. Der Inhalt des Feldes *extnValue* kann frei gestaltet werden.

Ob die Erweiterung als kritisch markiert wird, ist ebenfalls freigestellt. Für diese Entscheidung muss erwogen werden, dass ein Wert von *false* zur Folge hat, dass Implementierungen, welche die Erweiterung nicht interpretieren können, ein Zertifikat trotz deren Vorhandensein akzeptieren werden. Sofern also in der Erweiterung Informationen zur Steuerung der Zertifikatsverwendung enthalten sind, deren Nichtbeachtung zu Sicherheitsproblemen führen kann, dann ist dies eine starke Indikation dafür, die Erweiterung als kritisch zu markieren.

Gleichzeitig muss beachtet werden, dass, falls die proprietäre Erweiterung als kritisch markiert wird, dies dazu führt, dass Implementierungen, die diese nicht interpretieren können, keine Pfade erfolgreich prüfen können, in denen diese Zertifikatserweiterung vorkommt.

Die allgemeine Beschreibung von Zertifikatserweiterungen und die Erwähnung der Möglichkeit von proprietären Erweiterungen ist zu finden in [RFC 5280], Abschnitt 4.2.

3 CA-Zertifikate

Ein CA-Zertifikat dient der Verifikation der von der betreffenden CA ausgestellten Zertifikate. Die Anforderungen an die Inhalte eines CA-Zertifikats sind in Tabelle 22 dargestellt. Auf die dort aufgeführten Zertifikatserweiterungen wird in den nachfolgenden Unterabschnitten genauer eingegangen.

Feld	**Spezifikation**	
version	MUST: enthalten, Wert gleich „v3“	
serialNumber	MUST: enthalten, eindeutig für einen Aussteller	
signature	MUST: enthalten	
issuer	MUST: binär identisch zu subject im Ausstellerzertifikat	
validity	MUST: enthalten	
subject	MUST: enthalten, nicht-leer, Wert ist eine unverwechselbare Referenz auf den Inhaber	
subjectPublicKeyInfo	MUST: enthalten	
issuerUniqueID	MUST: abwesend	
subjectUniqueID	MUST: abwesend	
Zertifikatserweiterungen	**Spezifikation**	**Kritisch**
Basic Constraints	MUST: enthalten, siehe Unterabschnitte	MUST: kritisch
Key Usage	SHOULD: enthalten, siehe Unterabschnitte	SHOULD: kritisch
Certificate Policies	Optional	SHOULD: kritisch
Authority Key Identifier	MUST: enthalten	MUST: nicht kritisch
Subject Key Identifier	MUST: enthalten	MUST: nicht kritisch
Authority Information Access	Empfehlung: enthalten	MUST: nicht kritisch

Tabelle 22: Zertifikatsprofil CA

3.1 Zentrale Zertifikatserweiterungen

In diesem Abschnitt werden die zentralen Zertifikatserweiterungen, die für jedes CA-Zertifikat relevant sind, im Detail erläutert.

3.1.1 Basic Constraints

Mit dieser Erweiterung wird das Zertifikat als CA-Zertifikat ausgewiesen, indem das Feld *cA* innerhalb dieser Erweiterung auf den Wert *true* gesetzt wird.

Das optionale zusätzliche Feld *pathLenConstraint* dieser Erweiterung gibt an, wie viele CA-Zertifikate nach diesem Zertifikat im Pfad erlaubt sind. Dadurch kann eine CA zum Beispiel einer untergeordneten CA

verbieten, weitere CA- Zertifikate auszustellen. Der im Feld *pathLenConstraint* eingetragene Wert ist eine nicht-negative ganze Zahl.

Ein v3-Zertifikat darf nur dann für die Überprüfung von Signaturen anderer Zertifikate benutzt werden, wenn diese Erweiterung vorhanden ist und das Zertifikat als CA-Zertifikat markiert.

Name	Basic Constraints
OID	2.5.29.19
Kritisch	MUST: als kritisch markiert
Spezifikation	Abschnitt 4.2.1.9 in [RFC 5280]
Merkmale	MUST: Muss vorhanden sein. MUST: Feld *cA* muss vorhanden sein und den Wert *true* haben.

Tabelle 23*: Basic Constraints Erweiterung bei CA-Zertifikaten – Übersicht*

3.1.2 Key Usage

In dieser Erweiterung werden die kryptographischen Verwendungen des jeweiligen Schlüssels festgelegt. Im Fall einer CA, die Zertifikate ausstellt, muss hier der Zweck *keyCertSign* gesetzt sein. Wenn ein Zertifikat für die Ausstellung von Sperrlisten vorgesehen ist, dann muss der Zweck *cRLSign* gesetzt sein. Für den Fall, dass ein Ausstellerzertifikat auch für die Verifikation von OCSP-Antworten verwendet werden soll, wird empfohlen, den Verwendungszweck *digitalSignature* zu setzen. Für den Fall, dass ein Zertifikat für mehrere dieser Zwecke eingesetzt werden soll, so wird die entsprechende Kombination von Verwendungszwecken in der Key Usage Erweiterung gesetzt.

Name	Key Usage
OID	2.5.29.15
Kritisch	SHOULD: als kritisch markiert
Spezifikation	Abschnitt 4.2.1.3 in [RFC 5280]
Merkmale	MUST: Die passenden Verwendungszwecke müssen eingetragen sein. Im Falle der Ausstellung von Zertifikaten: *keyCertSign* Im Falle der Ausstellung von Sperrlisten: *cRLSign* Im Falle der Signatur von OCSP-Antworten: *digitalSignature*

Tabelle 24: Key Usage Erweiterung bei CA-Zertifikaten – Übersicht

3.2 Alternative Zertifikatserweiterungen

In diesem Abschnitt werden solche Zertifikatserweiterungen besprochen, die unter bestimmten Umständen in CA-Zertifikaten inkludiert werden können.

3.2.1 Policy Mappings

Diese Erweiterung wird bei CA-Zertifikaten benutzt, um Richtlinien verschiedener Domains einander zuzuordnen und als vergleichbar oder gleichwertig zu deklarieren. Dadurch können Anwendungen, die einer Richtlinie einer CA vertrauen, auch der zugeordneten Richtlinie einer anderen CA vertrauen, deren Richtlinien durch diese Erweiterung als mindestens äquivalent ausgewiesen werden.

Die Verwendung von Policy Mappings kann in Public-Key Infrastrukturen erwogen werden, in denen verschiedene Richtlinien zum Einsatz kommen, und in denen Cross-Zertifizierung zwischen unterschiedlichen an der Infrastruktur beteiligten Domänen verwendet wird. In einem solchen Szenario kann die Erweiterung dazu benutzt werden, zunächst unabhängig voneinander etablierte Richtlinien als äquivalent zu erklären.

Nicht erlaubt ist die Verwendung der speziellen *anyPolicy* in dieser Erweiterung.

Name	Policy Mappings
OID	2.5.29.33
Kritisch	SHOULD: als kritisch markiert
Spezifikation	Abschnitt 4.2.1.5 in [RFC 5280]
Merkmale	MUST: Es ist keine Abbildung von oder zu *anyPolicy* erlaubt.

Tabelle 25: Policy Mappings Erweiterung bei CA-Zertifikaten – Übersicht

3.2.2 Policy Constraints

Diese Erweiterung schränkt die Nutzung von Richtlinien für im Pfad nachfolgende Zertifikate ein. Möglich ist z.B. die Einschränkung der Verarbeitung von Policy Mappings Erweiterungen (siehe Abschnitt 3.2.1). Dadurch kann die Abbildung von Richtlinien verboten werden. Ferner kann mit dieser Erweiterung die Anwesenheit einer expliziten Richtlinie in im Pfad nachfolgenden Zertifikaten erzwungen werden.

Die Verwendung dieser Erweiterung kommt, ebenso wie im Fall von Policy Mappings, dann in Frage, wenn in einer Public-Key Infrastruktur verschiedene Domänen durch Cross-Zertifizierung verbunden sind. In diesem Fall hat eine CA über die von einer von ihr cross-zertifizierten CA ausgestellten Zertifikate keine direkte Kontrolle. Daher kann es im Interesse einer CA sein, Einschränkungen für die Richtlinien festzulegen, die in den über ihre eigenen Zertifikate validierbaren Pfaden auftreten.

Name	Policy Constraints
OID	2.5.29.36
Kritisch	MUST: kritisch
Spezifikation	Abschnitt 4.2.1.11 in [RFC 5280]
Merkmale	MUST: Darf nicht leer sein, mindestens ein Wert muss eingetragen sein.

Tabelle 26: Policy Constraints Erweiterung bei CA-Zertifikaten – Übersicht

3.2.3 Inhibit anyPolicy

Diese Erweiterung wird in CA-Zertifikaten benutzt, um die Bearbeitung der speziellen Richtlinie *anyPolicy* (siehe Abschnitt 2.3.6) zu verbieten. Sie gibt eine Anzahl von im Zertifizierungspfad nachfolgenden Zertifikaten an, nach der diese Richtlinie in weiteren Zertifikaten im Pfad nicht verarbeitet werden darf. Dadurch behält die CA die Kontrolle darüber, ob die *anyPolicy* in im Pfad nachfolgenden Zertifikaten benutzt werden darf. Ein Grund für den Einsatz ist, dass sie selbst nicht überprüfen kann, ob eine andere CA im Pfad die *anyPolicy* Richtlinie in den von ihr ausgestellten Zertifikaten benutzt. Sie kann zum Beispiel angeben, dass sie selbst die *anyPolicy* verwenden darf, aber keine im Pfad nachfolgende CA dies darf. In diesem Fall trägt sie die Zahl 0 in diese Erweiterung ein.

Die typischen Szenarien für den Einsatz dieser Erweiterung sind die gleichen wie im Falle der Policy Mappings und Policy Constraints Erweiterung.

Name	Inhibit anyPolicy
OID	2.5.29.54
Kritisch	MUST: kritisch
Spezifikation	Abschnitt 4.2.1.14 in [RFC 5280]
Merkmale	-

Tabelle 27: Inhibit anyPolicy Erweiterung bei CA-Zertifikaten – Übersicht

3.2.4 Name Constraints

Diese Erweiterung schränkt den Namensraum für die Namen der Inhaber aller im Pfad nachfolgenden Zertifikate ein. Nicht betroffen von der Einschränkung sind selbst ausgestellte[1] (self-issued) Zertifikate. Auf diese Weise ist es einem Aussteller möglich, solche Zertifikate für die eigenen Schlüssel unabhängig von den Einschränkungen auszustellen, die für Zertifikate gelten, die für andere Entitäten ausgestellt werden.

Die Verwendung dieser Erweiterung sollte dann in Betracht gezogen werden, wenn CA-Zertifikate ausgestellt werden, die einen beschränkten Gültigkeitsbereich haben sollen. Ein Beispiel wäre, dass man für das eigene Unternehmen ein Zertifikat für eine CA erzeugt, die nur TLS Server Zertifikate für eine bestimmte Domäne ausstellen können soll. In diesem CA-Zertifikat können mittels der Name Constraints Erweiterung die möglichen DNS Namen in der Subject Alternative Name Erweiterung und im *subject* beispielsweise auf die Subdomain *.restricteddomain.example.com eingeschränkt werden, wobei „*“ als Platzhalter für beliebige weitere Labels fungiert.

Diese Erweiterung wird nur bei nicht selbst ausgestellten (self-issued) CA-Zertifikaten verwendet.

Name	Name Constraints
OID	2.5.29.30

[1] Ein selbst ausgestelltes (self-issued) Zertifikat ist dadurch gekennzeichnet, dass dessen Felder *issuer* und *subject* inhaltlich übereinstimmen. In einem Zertifizierungspfad können z.B. dann selbst ausgestellte Zertifikate vorkommen, wenn ein Aussteller eigene Zertifikate mit verschiedenen Schlüssellängen sich gegenseitig signieren lässt.

Kritisch	MUST: als kritisch markiert
Spezifikation	Abschnitt 4.2.1.10 in [RFC 5280]
Merkmale	MUST: Darf nur in CA-Zertifikaten verwendet werden. MUST: Wenn sie vorhanden ist, darf sie nicht leer sein.

Tabelle 28: Name Constraints Erweiterung bei CA-Zertifikaten – Übersicht

3.2.5 Subject Information Access

Diese Erweiterung entspricht der Erweiterung Authority Information Access (siehe Abschnitt 2.3.9) für den Inhaber des Zertifikats. Sie unterstützt zwei Methoden. Die eine gibt den Ort an, an dem ein Server mit einem Zeitstempeldienst liegt. Sie wird bei Zertifikaten von Zeitstempeldiensten benutzt. Die andere gibt den Ort an, an dem eine CA von ihr ausgestellte Zertifikate veröffentlicht. Es ist möglich, mehrere Orte in der Erweiterung zu spezifizieren. Diese Methode kann in CA-Zertifikaten benutzt werden.

Name	Subject Information Access (SIA)
OID	1.3.6.1.5.5.7.1.11
Kritisch	MUST: nicht kritisch
Spezifikation	Abschnitt 4.2.2.2 in [RFC 5280]
Merkmale	-

Tabelle 29: Subject Information Access Erweiterung bei CA-Zertifikaten – Übersicht

4 TLS Server Zertifikate

TLS Server Zertifikate dienen dazu, einen TLS Server gegenüber einem TLS Client zu authentisieren. Die von diesem Dokument vorgegebenen Zertifikatsinhalte für TLS Server Zertifikate sind in der Tabelle 30 angegeben.

Feld	**Spezifikation**	
version	MUST: enthalten MUST: Version 3	
serialNumber	MUST: enthalten, eindeutig für einen Aussteller	
signature	MUST: enthalten	
issuer	MUST: binär identisch zu subject im Ausstellerzertifikat	
validity	MUST: enthalten	
subject	MUST: enthalten; Empfehlung: enthält nicht die Adresse des Servers	
subjectPublicKeyInfo	MUST: enthalten	
issuerUniqueID	MUST: abwesend	
subjectUniqueID	MUST: abwesend	
Zertifikatserweiterungen	**Spezifikation**	**Kritisch**
Basic Constraints	Empfehlung: enthalten	Empfehlung: nicht kritisch
Key Usage	SHOULD: enthalten, siehe Unterabschnitte	SHOULD: kritisch
Certificate Policies	Optional	SHOULD: kritisch
Authority Key Identifier	MUST: enthalten	MUST: nicht kritisch
Subject Key Identifier	Empfehlung: enthalten	MUST: nicht kritisch
Authority Information Access	Empfehlung: enthalten	MUST: nicht kritisch
Subject Alternative Name	Empfehlung: enthalten	MUST: kritisch, falls *subject* leer ist
Extended Key Usage	Empfehlung: enthalten	Empfehlung: nicht kritisch

Tabelle 30: Zertifikatsprofil TLS Server Zertifikate

4.1 Zentrale Zertifikatserweiterungen

In diesem Abschnitt werden alle für TLS Server Zertifikate relevante Zertifikatserweiterungen im Detail erläutert.

4.1.1 Basic Constraints

Um das TLS Server Zertifikat als ein End-Entity-Zertifikat auszuweisen, wird empfohlen, diese Erweiterung zu verwenden. Wenn die Erweiterung verwendet wird, muss das Feld *cA* den Wert *false* haben[2] und das Feld *pathLenConstraint* sollte nicht vorhanden sein.

Name	Basic Constraints
OID	2.5.29.19
Kritisch	Darf sowohl als kritisch als auch als nicht-kritisch markiert sein.
Spezifikation	Abschnitt 4.2.1.9 in [RFC 5280]
Merkmale	MUST: wenn die Erweiterung vorhanden ist, dann muss das Feld *cA* den Wert *false* haben. Empfehlung: *pathLenConstraint* nicht vorhanden.

Tabelle 31: Basic Constraints Erweiterung in TLS Server Zertifikaten – Übersicht

4.1.2 Subject Alternative Name

In einem TLS Server Zertifikat muss neben dem öffentlichen Schlüssel auch die Adresse des Servers festgelegt sein. In der Vergangenheit wurde zu diesem Zweck der DNS Name im *CommonName* des Inhaberfeldes (*subject*) angegeben. Diese Praxis wird heute nicht mehr empfohlen. Stattdessen soll der DNS Name in der Subject Alternative Name Erweiterung angegeben werden [RFC 6125], [RFC 2818].

Ebenso sollte von der Verwendung von Wildcard-Zertifikaten nach Möglichkeit abgesehen werden [RFC 6125]. Dies sind Zertifikate, bei denen im relevanten Namensfeld der Platzhalter (Wildcard Character) „*“ eingetragen ist. Solche Zertifikate werden benutzt, um ein Zertifikat auf Servern mit verschiedenen Adressen innerhalb einer Domain einzusetzen. TLS-Implementierungen steht es entsprechend den Vorgaben aus [RFC 6125] frei, ob sie einen Namensvergleich auf Basis von Platzhaltern durchführen oder solche Zertifikate grundsätzlich abweisen.

Name	Subject Alternative Name (SAN)
OID	2.5.29.17
Kritisch	MUST: kritisch, wenn leeres *subject* SHOULD: sonst nicht kritisch
Spezifikation	Abschnitt 4.2.1.6 in [RFC 5280], [RFC 6125]
Merkmale	MUST: Enthält DNS Namen (in *dNSName*) oder die IP-Adressen (in *iPAddress*) des Servers. MUST: Die individuellen Werte und deren Kodierung müssen konform zu den entsprechenden Spezifikationen sein. MUST: Der Wert der Erweiterung darf nicht leer sein.

[2] Weil *false* der Defaultwert ist und die Zertifikate DER kodiert sind, darf dieser Wert nicht kodiert werden (siehe Abschnitt 11.5 in [X.690]).

	SHOULD, Empfehlung: Es sollen keine Wildcard Namen verwendet werden.

Tabelle 32: Subject Alternative Name Erweiterung in TLS Server Zertifikaten – Übersicht

4.1.3 Key Usage

Die Key Usage Erweiterung legt die Verwendungszwecke des im Zertifikat enthaltenen öffentlichen Schlüssels fest. Im Falle eines TLS Server Zertifikats muss der eingetragene Verwendungszweck zu dem verwendeten Schlüsselaustauschverfahren (Key Exchange Algorithm) passen. Dabei gelten im Falle von TLS 1.2 die in Tabelle 33 wiedergegebenen Vorgaben aus [RFC 5246], Abschnitt 7.4.2.

Schlüsselaustauschverfahren nach [RFC 5246]	Notwendige Key Usage Verwendungszwecke
RSA oder PSA_PSK	MUST: keyEncipherment
DHE_RSA, ECDHE_RSA, DHE_DSS, ECDHE_ECDSA	MUST: digitalSignature
DH_DSS, DH_RSA	MUST: keyAgreement
ECDH_ECDSA, ECDH_RSA	Empfehlung: keyAgreement

Tabelle 33: Schlüsselverwendungszwecke in TLS Server Zertifikaten in Abhängigkeit von TLS 1.2 Schlüsselaustauschverfahren

Im Fall von TLS 1.3 [RFC 8446] gilt die Anforderung, dass im Falle einer gesetzten Key Usage Erweiterung der Verwendungszweck *digitalSignature* gesetzt sein muss.

Name	Key Usage
OID	2.5.29.15
Kritisch	SHOULD: kritisch
Spezifikation	Abschnitt 4.2.1.3 in [RFC 5280], Abschnitt 4.4.2.2 in [RFC 8446]
Merkmale	SHOULD: Sollte gesetzt werden. MUST: Wenn die Erweiterung im Zertifikat vorkommt, ist im Fall von TLS 1.2 mindestens eine der Verwendungen *keyEncipherment*, *keyAgreement* oder *digitalSignature* gesetzt, im Fall von TLS 1.3 ist *digitalSignature* gesetzt.

Tabelle 34: Key Usage Erweiterung in TLS Server Zertifikaten – Übersicht

4.1.4 Extended Key Usage

Die Extended Key Usage Erweiterung legt fest, für welche Anwendungszwecke ein Schlüssel benutzt werden darf. Wenn die Extended Key Usage Erweiterung in einem TLS Server Zertifikat gesetzt ist, muss dort der Zweck *serverAuth* (siehe Abschnitt 2.3.3) gesetzt werden. Dies kann Implementierungen helfen, die Verwendung des Zertifikats auf die TLS Server Authentisierung einzuschränken.

Name	Extended Key Usage
OID	2.5.29.37

Kritisch	Darf sowohl als kritisch als auch als nicht-kritisch markiert sein.
Spezifikation	Abschnitt 4.2.1.12 in [RFC 5280]
Merkmale	SHOULD: Die Erweiterung sollte gesetzt werden. Empfehlung: Wenn vorhanden, muss mindestens der Zweck *serverAuth* mit dem OID 1.3.6.1.5.5.7.3.1 gesetzt sein.

Tabelle 35: Extended Key Usage Erweiterung in TLS Server Zertifikaten – Übersicht

5 TLS Client Zertifikate

TLS Client Zertifikate dienen dazu, einen TLS Client gegenüber einem TLS Server zu authentisieren. Die für solche Zertifikate geltenden Vorgaben und Empfehlungen sind in Tabelle 36 zusammengestellt.

Feld	**Spezifikation**	
version	MUST: enthalten MUST: Version 3	
serialNumber	MUST: enthalten, eindeutig für einen Aussteller	
signature	MUST: enthalten	
issuer	MUST: binär identisch zu subject im Ausstellerzertifikat	
validity	MUST: enthalten	
subject	MUST: enthalten	
subjectPublicKeyInfo	MUST: enthalten	
issuerUniqueID	MUST: abwesend	
subjectUniqueID	MUST: abwesend	
Zertifikatserweiterungen	**Spezifikation**	**Kritisch**
Basic Constraints	Empfehlung: enthalten	Empfehlung: nicht kritisch
Key Usage	SHOULD: enthalten, siehe Unterabschnitte	SHOULD: kritisch
Certificate Policies	Optional	SHOULD: kritisch
Authority Key Identifier	MUST: enthalten	MUST: nicht kritisch
Subject Key Identifier	Empfehlung: enthalten	MUST: nicht kritisch
Authority Information Access	Empfehlung: enthalten	MUST: nicht kritisch
Subject Alternative Name	Optional	MUST: kritisch, falls *subject* leer ist
Extended Key Usage	Empfehlung: enthalten	Empfehlung: nicht kritisch

Tabelle 36: Zertifikatsprofil TLS Client Zertifikate

5.1 Zentrale Zertifikatserweiterungen

In diesem Abschnitt werden alle für TLS Client Zertifikate relevante Zertifikatserweiterungen im Detail erläutert.

5.1.1 Basic Constraints

Um TLS Client Zertifikate als End-Entity-Zertifikate auszuweisen, wird empfohlen, diese Erweiterung zu verwenden. Wenn die Erweiterung verwendet wird, muss das Feld *cA* den Wert *false* haben[3] und das Feld *pathLenConstraint* sollte nicht vorhanden sein.

Name	Basic Constraints
OID	2.5.29.19
Kritisch	Darf sowohl als kritisch als auch als nicht-kritisch markiert sein.
Spezifikation	Abschnitt 4.2.1.9 in [RFC 5280]
Merkmale	MUST: wenn die Erweiterung vorhanden ist, dann muss das Feld *cA* den Wert *false* haben. Empfehlung: *pathLenConstraint* nicht vorhanden.

Tabelle 37: Basic Constraints Erweiterung in TLS Client Zertifikaten – Übersicht

5.1.2 Key Usage

Die Key Usage Erweiterung legt die Verwendungszwecke des im Zertifikat enthaltenen öffentlichen Schlüssels fest. Im Falle eines TLS Client Zertifikats muss im Fall von TLS 1.2 der Verwendungszweck zu einem von den vom Server im Protokollfeld „Client Certificate Type“ geforderten Zertifikatstypen passen. Dabei gelten im Falle von TLS 1.2 die in Tabelle 38 zusammengefassten Vorgaben aus [RFC 5246], Abschnitt 7.4.6.

Client Certificate Type nach [RFC 5246]	Notwendige Key Usage Verwendungszwecke
rsa_sign, dss_sign, ecdsa_sign	MUST: digitalSignature
rsa_fixed_dh, dss_fixed_dh, rsa_fixed_ecdh, ecdsa_fixed_ecdh	Empfehlung: keyAgreement

Tabelle 38: Schlüsselverwendungszwecke in TLS 1.2 Client Zertifikaten

Für TLS 1.3 gibt es grundsätzlich keine spezifischen Anforderungen an diese Zertifikatserweiterung. Allerdings sieht TLS 1.3 die Möglichkeit vor, dass der Server angeben kann, dass er spezielle Zertifikatserweiterungen mit speziellen Inhalten im TLS Client Zertifikat erwartet. Dies erfolgt mittels sogenannter OID Filter (Abschnitt 4.2.5 in [RFC 8446]). Auf diese Weise kann der Server auch Anforderungen bezüglich der Key Usage Erweiterung im TLS Client Zertifikat definieren.

Name	Key Usage
OID	2.5.29.15
Kritisch	SHOULD: kritisch

[3] Weil *false* der Defaultwert ist und die Zertifikate DER kodiert sind, darf dieser Wert nicht kodiert werden (siehe Abschnitt 11.5 in [X.690]).

Spezifikation	Abschnitt 4.2.1.3 in [RFC 5280], [RFC 5246]
Merkmale	SHOULD: Sollte gesetzt werden. SHOULD: Im Fall von TLS 1.2 gilt, dass wenn die Erweiterung im Zertifikat vorkommt, mindestens eine der Verwendungen *keyAgreement* oder *digitalSignature* gesetzt ist. MUST: Wenn die Erweiterung im Zertifikat vorkommt, ist mindestens ein Schlüsselverwendungszweck gesetzt.

Tabelle 39: Key Usage Erweiterung in TLS Client Zertifikaten – Übersicht

5.1.3 Extended Key Usage

Die Extended Key Usage Erweiterung legt fest, für welche Anwendungszwecke ein Schlüssel benutzt werden darf. Wenn die Extended Key Usage Erweiterung in einem TLS Client Zertifikat gesetzt ist, muss dort der Zweck *clientAuth* (siehe Abschnitt 2.3.3) gesetzt werden. Dies kann Implementierungen helfen, die Verwendung des Zertifikats auf die TLS Client Authentisierung einzuschränken.

Name	Extended Key Usage
OID	2.5.29.37
Kritisch	Darf sowohl als kritisch als auch als nicht-kritisch markiert sein.
Spezifikation	Abschnitt 4.2.1.12 in [RFC 5280]
Merkmale	SHOULD: Die Erweiterung sollte gesetzt werden. Empfehlung: Wenn vorhanden, muss mindestens der Zweck *clientAuth* mit dem OID 1.3.6.1.5.5.7.3.2 gesetzt sein.

Tabelle 40: Extended Key Usage Erweiterung in TLS Client Zertifikaten – Übersicht

6 S/MIME Zertifikate

S/MIME Zertifikate werden für die Signaturverifikation und Verschlüsselung von S/MIME E-Mail Nachrichten verwendet. Die für solche Zertifikate geltenden Vorgaben und Empfehlungen können der Tabelle 41 und den nachfolgenden Unterabschnitten entnommen werden.

Feld	Spezifikation	
version	MUST: enthalten MUST: Version 3	
serialNumber	MUST: enthalten, eindeutig für einen Aussteller	
signature	MUST: enthalten	
issuer	MUST: binär identisch zu subject im Ausstellerzertifikat	
validity	MUST: enthalten	
subject	MUST: enthalten	
subjectPublicKeyInfo	MUST: enthalten	
issuerUniqueID	MUST: abwesend	
subjectUniqueID	MUST: abwesend	
Zertifikatserweiterungen	**Spezifikation**	**Kritisch**
Basic Constraints	Empfehlung: enthalten	Empfehlung: nicht kritisch
Key Usage	SHOULD: enthalten, siehe Unterabschnitte	SHOULD: kritisch
Certificate Policies	Optional	SHOULD: kritisch
Authority Key Identifier	MUST: enthalten	MUST: nicht kritisch
Subject Key Identifier	Empfehlung: enthalten	MUST: nicht kritisch
Authority Information Access	Empfehlung: enthalten	MUST: nicht kritisch
Subject Alternative Name	Empfehlung: enthalten	MUST: kritisch, falls *subject* leer ist
Extended Key Usage	Empfehlung: enthalten	Empfehlung: nicht kritisch

Tabelle 41: Zertifikatsprofil S/MIME Zertifikate

6.1 Zentrale Zertifikatserweiterungen

In diesem Abschnitt werden alle für S/MIME Zertifikate relevante Zertifikatserweiterungen im Detail erläutert.

6.1.1 Basic Constraints

Um S/MIME Zertifikate als End-Entity-Zertifikate auszuweisen wird empfohlen, diese Erweiterung zu verwenden. Wenn die Erweiterung verwendet wird, muss das Feld *cA* den Wert *false* haben[4] und das Feld *pathLenConstraint* sollte nicht vorhanden sein.

Name	Basic Constraints
OID	2.5.29.19
Kritisch	Darf sowohl als kritisch als auch als nicht-kritisch markiert sein.
Spezifikation	Abschnitt 4.2.1.9 in [RFC 5280]
Merkmale	MUST: wenn die Erweiterung vorhanden ist, dann muss das Feld *cA* den Wert *false* haben. Empfehlung: *pathLenConstraint* nicht vorhanden.

Tabelle 42: Basic Constraints Erweiterung in S/MIME Zertifikaten – Übersicht

6.1.2 Subject Alternative Name

In einem S/MIME Zertifikat muss neben dem öffentlichen Schlüssel auch die E-Mail-Adresse des Benutzers enthalten sein. Entsprechend [RFC 5750], Abschnitt 4.4.3, soll zu diesem Zweck die Subject Alternative Name Erweiterung verwendet werden. Die E-Mail-Adresse muss dabei in einem Feld des Typs *rfc822Name* kodiert werden.

Name	Subject Alternative Name (SAN)
OID	2.5.29.17
Kritisch	MUST: kritisch, wenn leeres *subject* SHOULD: sonst nicht kritisch
Spezifikation	Abschnitt 4.2.1.6 in [RFC 5280], [RFC 5750]
Merkmale	Empfehlung: Enthält die E-Mail-Adresse des Benutzers MUST: Der Wert der Erweiterung darf nicht leer sein.

Tabelle 43: Subject Alternative Name Erweiterung in S/MIME Zertifikaten – Übersicht

6.1.3 Key Usage

Die Key Usage Erweiterung legt die Verwendungszwecke des im Zertifikat enthaltenen öffentlichen Schlüssels fest. Wenn die Key Usage Erweiterung in einem S/MIME Zertifikat gesetzt wird, so gelten folgende Regeln für die zu setzenden Schlüsselverwendungszwecke:

[4] Weil *false* der Defaultwert ist und die Zertifikate DER kodiert sind, darf dieser Wert nicht kodiert werden (siehe Abschnitt 11.5 in [X.690]).

Falls das Zertifikat für die Verifikation von Signaturen benutzt werden soll, dann muss mindestens einer der Zwecke *digitalSignature* oder *nonRepudiation* gesetzt werden, andernfalls müssen Clients das Zertifikat für diesen Zweck zurückweisen (Abschnitt 4.4.2 in [RFC 5750]).

Falls das Zertifikat für die Verschlüsselung von Nachrichten verwendet wird, dann muss laut [RFC 5280] *keyEncipherment* gesetzt sein. Allerdings wird im für S/MIME primär relevanten Standard [RFC 5750] eine Prüfung des Verwendungszwecks *keyEncipherment* durch den Client nicht vorgegeben. Als Empfehlung gilt, dass für den Zweck der Nachrichtenverschlüsselung *keyEncipherment* als Verwendungszweck in der Key Usage Erweiterung gesetzt sein muss, falls diese Erweiterung im Zertifikat vorhanden ist.

Name	Key Usage
OID	2.5.29.15
Kritisch	SHOULD: kritisch
Spezifikation	Abschnitt 4.2.1.3 in [RFC 5280], Abschnitt 4.4.2 in [RFC 5750]
Merkmale	MUST: Wenn die Erweiterung im Zertifikat vorkommt, ist mindestens eine explizite Verwendung gesetzt. Siehe den Text (Abschnitt 6.1.3) bzgl. der konkreten Werte im Falle von S/MIME.

Tabelle 44: Key Usage Erweiterung in S/MIME Zertifikaten – Übersicht

6.1.4 Extended Key Usage

Die Extended Key Usage Erweiterung legt fest, für welche Anwendungszwecke ein Schlüssel benutzt werden darf. Wenn die Extended Key Usage Erweiterung in einem S/MIME Zertifikat gesetzt ist, muss dort der Zweck *emailProtection* (siehe Abschnitt 2.3.3) gesetzt werden. Dies kann Implementierungen helfen, die Verwendung des Zertifikats auf S/MIME bzw. vergleichbare Verfahren einzuschränken.

Name	Extended Key Usage
OID	2.5.29.37
Kritisch	Darf sowohl als kritisch als auch als nicht-kritisch markiert sein.
Spezifikation	Abschnitt 4.2.1.12 in [RFC 5280]
Merkmale	SHOULD: Die Erweiterung sollte gesetzt werden. MUST: Wenn vorhanden, muss mindestens der Zweck *emailProtection* mit dem OID 1.3.6.1.5.5.7.3.4 gesetzt sein.

Tabelle 45: Extended Key Usage Erweiterung in S/MIME Zertifikaten – Übersicht

6.2 Alternative Zertifikatserweiterungen

In diesem Abschnitt werden solche Zertifikatserweiterungen besprochen, die unter bestimmten Umständen in S/MIME Zertifikaten verwendet werden können.

6.2.1 S/MIME Capabilities

Die S/MIME Capabilities stellen ein ASN.1 kodiertes Objekt dar, welches als Teil einer S/MIME Nachricht übertragen wird. Mit Hilfe der darin enthaltenen Informationen werden dem Empfänger die vom Sender unterstützten kryptographischen Algorithmen mitgeteilt. Dadurch wird der Empfänger in die Lage versetzt, beim Senden von Nachrichten eine gezielte Auswahl der für Verschlüsselung und Signatur verwendeten kryptographischen Algorithmen vornehmen zu können. Um diese Informationen schon vor dem Erhalt der ersten Nachricht verfügbar zu machen, kann die in [RFC 4262] definierte S/MIME Capabilities Zertifikatserweiterung in ein S/MIME Zertifikat aufgenommen werden.

Es ist allerdings nur dann sinnvoll von dieser Möglichkeit Gebrauch zu machen, wenn bei der Zertifikatserstellung bereits bekannt ist, welche kryptographischen Verfahren von den vom Inhaber des Zertifikats verwendeten E-Mail Clients mindestens unterstützt werden.

Name	S/MIME Capabilities
OID	1.2.840.113549.1.9.15
Kritisch	MUST: nicht kritisch
Spezifikation	[RFC 4262]
Merkmale	–

Tabelle 46: S/MIME Capabilities Erweiterung in S/MIME Zertifikaten – Übersicht

7 IPsec Zertifikate

Zertifikate für IPsec Knoten dienen dazu, sich gegenüber anderen IPsec Knoten zu authentisieren. Die für solche Zertifikate geltenden Vorgaben und Empfehlungen sind in Tabelle 47 angegeben

Feld	**Spezifikation**	
version	MUST: enthalten MUST: Version 3	
serialNumber	MUST: enthalten, eindeutig für einen Aussteller	
signature	MUST: enthalten	
issuer	MUST: binär identisch zu subject im Ausstellerzertifikat	
validity	MUST: enthalten	
subject	MUST: enthalten; Empfehlung: enthält nicht die Adresse des Servers	
subjectPublicKeyInfo	MUST: enthalten	
issuerUniqueID	MUST: abwesend	
subjectUniqueID	MUST: abwesend	
Zertifikatserweiterungen	**Spezifikation**	**Kritisch**
Basic Constraints	Empfehlung: enthalten	Empfehlung: nicht kritisch
Key Usage	SHOULD: enthalten, siehe Unterabschnitte	SHOULD: kritisch
Certificate Policies	Optional	SHOULD: kritisch
Authority Key Identifier	MUST: enthalten	MUST: nicht kritisch
Subject Key Identifier	Empfehlung: enthalten	MUST: nicht kritisch
Authority Information Access	Empfehlung: enthalten	MUST: nicht kritisch
Subject Alternative Name	MUST: enthalten, falls als ID nicht „DN“ benutzt wird	MUST: kritisch, falls *subject* leer ist

Tabelle 47: Zertifikatsprofil IPsec Zertifikate

7.1 Zentrale Zertifikatserweiterungen

In diesem Abschnitt werden alle für IPsec Zertifikate relevante Zertifikatserweiterungen im Detail erläutert.

7.1.1 Basic Constraints

Um IPsec Zertifikate als End-Entity-Zertifikate auszuweisen, wird empfohlen, diese Erweiterung zu verwenden. Wenn die Erweiterung verwendet wird, muss das Feld *cA* den Wert *false* haben[5] und das Feld *pathLenConstraint* sollte nicht vorhanden sein.

Name	Basic Constraints
OID	2.5.29.19
Kritisch	Darf sowohl als kritisch als auch als nicht-kritisch markiert sein.
Spezifikation	Abschnitt 4.2.1.9 in [RFC 5280]
Merkmale	MUST: wenn die Erweiterung vorhanden ist, dann muss das Feld *cA* den Wert *false* haben. Empfehlung: *pathLenConstraint* nicht vorhanden.

Tabelle 48: Basic Constraints Erweiterung in IPsec Zertifikaten – Übersicht

7.1.2 Subject Alternative Name

In dem Zertifikat eines IPsec Knotens muss die Identitätsinformation („ID“), mit der sich dieser gegenüber anderen Knoten ausweist, enthalten sein. Außer dem ID Typ „DN“ (Distinguished Name), dessen Wert durch das Feld *subject* im Zertifikat gegeben ist, sind die Werte aller ID Typen in der Subject Alternative Name Erweiterung enthalten. Dabei sollen (SHOULD) laut [RFC 4945], Abschnitt 5.1.3.6, nur die *GeneralName* Typen *rfc822Name*, *dNSName*, und *iPAddress* verwendet werden.

Name	Subject Alternative Name (SAN)
OID	2.5.29.17
Kritisch	MUST: kritisch, wenn leeres *subject* SHOULD: sonst nicht kritisch
Spezifikation	Abschnitt 4.2.1.6 in [RFC 5280], [RFC 4945]
Merkmale	SHOULD: Enthält DNS Namen (in *dNSName*) oder die IP-Adressen (in *iPAddress*) des Servers oder einen *rfc822Name* MUST: Die individuellen Werte und deren Kodierung müssen konform zu den entsprechenden Spezifikationen sein. MUST: Der Wert der Erweiterung darf nicht leer sein. MUST: Es dürfen keine Wildcard Namen verwendet werden. MUST: Bei IP-Adressen dürfen keine Adressbereiche ausgewiesen werden, sondern nur vollständig festgelegte Adressen.

Tabelle 49: Subject Alternative Name Erweiterung in IPsec Zertifikaten – Übersicht

[5] Weil *false* der Defaultwert ist und die Zertifikate DER kodiert sind, darf dieser Wert nicht kodiert werden (siehe Abschnitt 11.5 in [X.690]).

7.1.3 Key Usage

Die Key Usage Erweiterung legt die Verwendungszwecke des im Zertifikat enthaltenen öffentlichen Schlüssels fest. Im Falle eines Zertifikats, das im Rahmen des Internet Key Exchange (IKE bzw. IKEv2) Protokolls verwendet wird, muss mindestens einer der beiden Verwendungszwecke *digitalSignature* oder *nonRepudiation* [RFC 4945] in der Key Usage Erweiterung gesetzt sein. Es wird empfohlen, in solchen Zertifikaten keine weiteren Schlüsselverwendungszwecke aufzunehmen.

Name	Key Usage
OID	2.5.29.15
Kritisch	SHOULD: kritisch
Spezifikation	Abschnitt 4.2.1.3 in [RFC 5280], Abschnitt 5.1.3.2 in [RFC 4945]
Merkmale	SHOULD: Sollte gesetzt werden. MUST: Wenn die Erweiterung im Zertifikat vorkommt, ist mindestens eine der Verwendungen *nonRepudiation* oder *digitalSignature* gesetzt. Empfehlung: Im Fall von IKE oder IKEv2 sollte keine andere Verwendung außer *nonRepudiation* oder *digitalSignature* gesetzt sein.

Tabelle 50: Key Usage Erweiterung in IPsec Zertifikate – Übersicht

8 Zertifikatsprüfung

In diesem Abschnitt wird der Vorgang der Zertifikatsprüfung beschrieben. Hierbei ist mit „Zertifikatsprüfung“ ein Vorgang gemeint, bei dem die Verwendbarkeit eines Zertifikats in einem bestimmten Anwendungskontext für einen bestimmten Zweck geprüft wird. Dieser Vorgang setzt sich aus den folgenden Teilvorgängen zusammen:

1. Pfadkonstruktion: Basierend auf dem konfigurierten Satz von Vertrauensankern wird für das zu prüfende Zielzertifikat ein gültiger Pfad konstruiert.
2. Pfadvalidierung: Ein im vorherigen Schritt konstruierter Pfad wird entsprechend den Vorgaben zur Zertifizierungspfadvalidierung in [RFC 5280] auf seine Gültigkeit geprüft.
3. Revokationsprüfung: Der Revokationsstatus aller Zertifikate im Pfad wird geprüft.
4. Anwendungsspezifische Prüfungen: Entsprechend des Einsatzzweckes des Zertifikats sind nach der erfolgten Pfadvalidierung weitere Prüfungen bezüglich der Schlüsselverwendung und der im Zielzertifikat ausgewiesenen Namen notwendig.

8.1 Allgemeine Hinweise

Die Durchführung der Zertifikatsprüfung ist als ein äußerst sicherheitskritischer Vorgang anzusehen. Falls von einer Anwendung ein ungültiges Zertifikat als gültig akzeptiert wird, bedeutet dies im Allgemeinen einen sicherheitsrelevanten Vorfall. Daher ist es von größter Wichtigkeit, dass Implementierungen der Zertifikatsprüfung keine Fehler enthalten, die zu einem derartigen Fehlverhalten führen können. Dies kann nur gewährleistet werden, wenn eine Reihe von Sicherheitsmaßregeln bei der Erstellung und Verwendung einer solchen Implementierung zur Anwendung kommen.

Bevor die Darlegung der wesentlichen Sicherheitsmaßregeln erfolgt, sei darauf hingewiesen, dass die Darstellung derselben, sowie die aller weiter folgenden Vorgaben und Empfehlungen für die Umsetzung der Zertifikatsprüfung, auf einer relativ hohen Abstraktionsebene erfolgt. Dies ist der Tatsache geschuldet, dass davon alle denkbaren Implementierungen der Zertifikatsprüfung abgedeckt werden sollen. Diese können in Form von Anwendungen vorliegen, welche über eine entsprechende Funktionalität verfügen, oder in Form von Softwarebibliotheken, welche wiederum in Anwendungen eingesetzt werden. Somit muss für jede Implementierung bestimmt werden, an welcher Stelle und auf welche Weise die Umsetzung der hier getroffenen Vorgaben und Empfehlungen zu erfolgen hat.

8.1.1 Benutzung einer Anwendung

Bei einer Anwendung ist zu erwarten, dass die Benutzer im Wesentlichen folgende Vorgänge mit derselben durchführen:

- Die Konfiguration von Parametern. Ein typisches Beispiel ist die Option in der Bedienoberfläche eines Internet Browser, ob und wie der Sperrstatus eines Zertifikats zu prüfen ist.
- Die Konfiguration von Vertrauensankern. Hierbei sind die Hinweise in Abschnitt 8.2 zu beachten.
- Die Veranlassung von Zertifikatsprüfungen. Ein kritischer Punkt ist hierbei, ob es dem Benutzer zur Auswahl gestellt wird, auch Zertifikate zu akzeptieren, für die keine erfolgreiche Pfadvalidierung durchgeführt werden konnte. Im Falle von fehlgeschlagenen Zertifikatsprüfungen kann es je nach Anwendungskontext relevant werden, wie feingranular die Fehlerinformationen sind, die er erhält. Davon kann abhängen, ob er selbständig in der Lage ist, die Fehlerursache zu bestimmen und zu beheben.

Eine Voraussetzung für die sichere Benutzung einer Anwendung ist, dass die Gestaltung von deren Benutzeroberfläche in Zusammenhang mit der verfügbaren Dokumentation eine Fehlbenutzung möglichst unwahrscheinlich macht.

In manchen Anwendungskontexten kann es notwendig werden, die Optionen eines Benutzers bezüglich der oben genannten oder weiterer Interaktionsmöglichkeiten mit der Anwendung einzuschränken, um geltende

Sicherheitsrichtlinien durchzusetzen. Wie solche Maßnahmen durchgesetzt werden können, ist anwendungsspezifisch.

8.1.2 Implementierungen der Zertifikatsprüfung

Bei der Erstellung einer Anwendung muss einerseits sichergestellt werden, dass die unter Abschnitt 8.1.1 genannten Aspekte für die Benutzer einer Anwendung möglichst gut unterstützt werden. Andererseits basiert die Sicherheit der Zertifikatsprüfung auf der Korrektheit der konkreten Implementierung dieser Funktion und deren Einbindung in die Anwendung. Die korrekte Einbindung ist besonders dann relevant, wenn eine externe Softwarebibliothek für diesen Zweck verwendet wird.

Die Schnittstelle einer Bibliothek, d.h. deren Application Programming Interface (API), zur Zertifikatsprüfung, ist typischerweise wesentlich komplexer als eine Anwendungsschnittstelle, da sie eine größere Menge an Optionen für die Steuerung dieser Operation bietet. Um eine sichere Integration einer Bibliothek zu gewährleisten, sind die folgenden Aspekte besonders relevant.

Eine Voraussetzung für eine sichere Integration dieser Funktionalität in eine Anwendung ist die genaue Beachtung der Herstellerdokumentation für die Verwendung der Zertifikatsprüfung. Aus dieser muss hervorgehen,

- welche der Teilaufgaben Pfadkonstruktion, Pfadvalidierung, Revokationsprüfung und anwendungsspezifische Prüfungen von der Bibliothek umgesetzt werden,
- welche Parametrisierungs- und Konfigurationsoptionen die jeweiligen API Funktionen bieten und wie diese gesetzt werden können
- und welche Vorgabewerte gesetzt werden, wenn der Aufrufer selbst keine Vorgaben für die einzelnen Parameter macht. In diesem Fall sollten von der API sichere Vorgabewerte gesetzt werden.

Ein weiterer wichtiger Aspekt ist eine konsistente Fehlerbehandlung. Es muss unter allen Umständen gewährleistet sein, dass ein Prüfungs- oder genereller Verarbeitungsfehler zum Fehlschlagen der Zertifikatsprüfung führt. Dies gilt sowohl innerhalb der Implementierung der Zertifikatsprüfung als auch an der API Schnittstelle zur Anwendung und innerhalb der Anwendung selbst. Die verschiedenen Programmiersprachen bieten hier unterschiedliche Konzepte an. In manchen Programmiersprachen können Ausnahmen erzeugt werden, die solange durch die Aufrufkette nach oben propagiert werden, bis sie von einer geeigneten Stelle im Programmcode behandelt werden. Dieses Vorgehen bietet den Vorteil, dass die Propagation der Fehler ohne Zutun des übrigen Programmcodes sichergestellt ist, solange keine ungeeignete Behandlung der Ausnahmen erfolgt. Im Gegensatz dazu erfordert es weit mehr Programmierdisziplin, eine konsistente Fehlerbehandlung ohne das Konzept der Ausnahmen umzusetzen, etwa durch die Verwendung von Fehlercodes als Rückgabewerte der Funktionen. In solchen Fällen muss eine passende Systematik bei der Programmierung zum Einsatz kommen, die eine Propagation der Fehler bis zu einer Programmstelle, die sie geeignete behandelt, garantiert. Dabei muss ein konsistenter Zustand aller gültigen Objekte unter allen denkbaren Bedingungen garantiert sein.

Dabei ist auch zu beachten, dass die Implementierung der Verarbeitung von Zertifikaten und den weiteren beteiligten Datenformaten robust ausgelegt werden muss. Die Routinen müssen mit Datenformaten, die Fehler auf der Ebene der DER Kodierung oder des Aufbaus der Datenstruktur aufweisen, umgehen können, ohne dass dadurch die Sicherheit der Anwendung gefährdet wird, und das entsprechende Zertifikat in solchen Fällen abweisen.

Ein weiterer wichtiger Aspekt ist, dass im Falle von Implementierungen mit gegenüber dem Standard reduzierter Funktionalität, der Umgang mit nicht unterstützten Inhalten der verschiedenen Datenformate korrekt erfolgt. Dies bedeutet im Zweifelsfall immer die Abweisung der entsprechenden Zertifikate. So ist es z.B. von zentraler Bedeutung, dass ein Zertifikat mit nicht unterstützten, als kritisch markierten Erweiterungen abgewiesen wird.

8.2 Vertrauensanker

In diesem Abschnitt werden sicherheitsrelevante Aspekte zur Konfiguration der Vertrauensanker für eine Implementierung beschrieben.

8.2.1 Inhalte der Vertrauensanker

Aus Sicht des formalen Algorithmus nach [RFC 5280] zur Pfadvalidierung ist ein Vertrauensanker nicht notwendigerweise durch ein X.509 Zertifikat, sondern optional auch durch das Tupel seines Namens (*subject* im Zertifikat), den von ihm verwendeten Signaturalgorithmus und seinen öffentlichen Schlüssel inklusive möglicherweise zu dem Schlüssel gehörige Parameter definiert. Falls der Vertrauensanker in Form eines selbst-signierten Zertifikats bereitgestellt wird, dann wertet der formale Algorithmus nach [RFC 5280] trotzdem nur die in diesem Tupel zusammengefassten Eigenschaften aus.

In [RFC 5937] wird definiert, wie Inhalte eines Vertrauensankers ausgewertet werden. Dabei werden drei verschiedene Datenformate für Vertrauensanker berücksichtigt: „TrustAnchorInfo“, „Certificate“ und „TBSCertificate“. Das erste stellt ein spezielles Format dar, welches von den Standardzertifikatstypen abweicht und daher hier nicht behandelt wird. Beim zweiten handelt es sich um das gewohnte X.509 Zertifikatsformat, wobei das Zertifikat hierbei eine Selbstsignatur trägt. Das dritte bezeichnet den TBS („to-be-signed“) Teil eines X.509 Zertifikats, stellt also im Wesentlichen ein Zertifikat ohne Signatur dar. Sofern eine Implementierung eines der beiden letzteren Formate für Vertrauensanker verwendet, muss der Anwender mit folgenden beiden Möglichkeiten rechnen: Entweder nimmt die Implementierung die Pfadvalidierung nach [RFC 5280] vor und berücksichtigt somit das oben aufgeführte Tupel von Eigenschaften aus dessen Zertifikat. Oder sie verfährt abweichend von dem Standard und behandelt dieses Zertifikat wie die Zertifikate im Pfad und prüft alle Inhalte. Dies beinhaltet unter Umständen auch, im Falle von einem selbst-signierten Zertifikat im „Certificate “Datenformat, dessen Selbstsignatur.

Entsprechend den Vorgaben von [RFC 5937] werden die Inhalte der Vertrauensanker, sofern sie von der Implementierung verwendet werden, benutzt, um die initialen Werte der formalen Eingabeparameter des Algorithmus zur Pfadvalidierung aus [RFC 5280], die in Abschnitt 8.4.1.1 behandelt werden, zu setzen. Dabei handelt es sich um Parameter, welche die Möglichkeiten für gültige Zertifizierungspfade anhand bestimmter Kriterien einschränken. Ob eine Implementierung auf diese Weise verfährt, oder das (TBS) Zertifikat des Vertrauensankers wie das erste Zertifikat im Pfad behandelt, führt zum gleichen Ergebnis.

Aufgrund der Tatsache, dass Implementierungen unter Umständen Vertrauensanker, die im X.509 Format vorliegen, den gleichen Prüfungen unterziehen wie CA-Zertifikate, wird empfohlen, nur solche Zertifikate als Vertrauensanker zu verwenden, die nach den Kriterien für CA-Zertifikate gültig sind. Ferner wird empfohlen, sich darüber zu informieren, in wie weit eine verwendete Implementierung Inhalte der Vertrauensanker bei der Pfadvalidierung auswertet und über welche Parameter dies ggf. kontrolliert werden kann.

8.2.2 Handhabung von Vertrauensankern

Die Vertrauensanker bilden kritische Pfeiler von Systemen, die auf Public-Key Infrastrukturen aufbauen. Wenn in einem System ein Aussteller fälschlich als Vertrauensanker gesetzt ist, ist die Sicherheit aller auf dieser Infrastruktur aufbauenden Komponenten und Dienste gefährdet. Daraus leiten sich die folgenden Empfehlungen ab.

8.2.2.1 Übermittlung von Vertrauensankern

Wenn Vertrauensanker zwischen Parteien übermittelt werden, dann muss eine authentische Übermittlung sichergestellt sein. Wenn zu diesem Zweck kein spezifischer vertrauenswürdiger Kanal zwischen beiden Parteien zur Verfügung steht, kann die Übertragung der Information über einen beliebigen Kanal erfolgen. In diesem Fall muss aber ein kryptographischer Hashwert mit einem als sicher eingestuften Hashverfahren über alle zum Vertrauensanker gehörigen Daten gebildet, über einen vertrauenswürdigen Kanal übertragen und vom Empfänger überprüft werden.

8.2.2.2 Schutz vor unberechtigten Schreibzugriffen

Vertrauensanker müssen in den Komponenten vor unberechtigtem Schreiben bzw. Überschreiben geschützt abgelegt werden. Unberechtigte Benutzer dürfen keine Möglichkeit haben, neue Vertrauensanker zu setzen oder die Inhalte bestehender Vertrauensanker zu verändern. Falls Vertrauensanker in Form von selbstsignierten Zertifikaten vorliegen, so ist zu beachten, dass die Selbstsignatur im Allgemeinen keinerlei Integritätsschutz für die Zertifikatsinhalte bietet, da die Implementierung sie nicht prüfen muss, und da diese durch Austausch des öffentlichen Schlüssels leicht gefälscht werden können.

8.2.2.3 Durchsetzung einer Revokation

Bei der Pfadvalidierung nach [RFC 5280] ist keine Prüfung des Revokationsstatus des Vertrauensankers vorgesehen. Daher muss bei der Administration eines Systems mit konfigurierten Vertrauensankern sichergestellt sein, dass die Konfiguration eines entzogenen Vertrauensstatus technisch ohne Verzögerung umsetzbar ist. Dies ist beispielsweise notwendig, wenn eine Kompromittierung des zugehörigen geheimen Schlüssels erfolgt ist.

8.2.2.4 Durchsetzung eines Gültigkeitszeitraumes

Bei der Pfadvalidierung nach [RFC 5280] ist auch bei Verwendung der Inhalte eines Vertrauensankers nach [RFC 5937] keine Prüfung von dessen Gültigkeitszeitraum vorgesehen. Daher muss die rechtzeitige Entfernung eines als nicht mehr gültig angesehenen Vertrauensankers ggf. durch andere Mittel sichergestellt werden.

8.3 Pfadkonstruktion

<table>
<tr><td>Bezeichnung</td><td colspan="2">Pfadkonstruktion</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Durch die Pfadkonstruktion wird aus einer Menge von Vertrauensankern, einer Menge von CA-Zertifikaten ohne Vertrauensstatus und einem Zielzertifikat ein Pfad konstruiert, indem die Verkettung von Namen bzw. die Verkettung über Ausstellerschlüssel beachtet wird.</td></tr>
<tr><td>Anzuwenden auf</td><td colspan="2">Die Eingabeparameter</td></tr>
<tr><td>Referenzen</td><td colspan="2">Abschnitt 4.1.2.4 in [RFC 5280], [RFC 4158]</td></tr>
<tr><td>Typ des Eingabeparameters</td><td>Verwendete Parameter</td><td>Verwendung</td></tr>
<tr><td>Basisfelder des Zertifikats</td><td>subject, issuer</td><td>Überprüfung der Namensverkettung für jeden möglichen Pfadkandidaten</td></tr>
<tr><td rowspan="3">Zertifikatserweiterungen</td><td>Authority Key Identifier</td><td rowspan="2">Überprüfung der Verkettung über Ausstellerschlüssel für jeden möglichen Pfadkandidaten</td></tr>
<tr><td>Subject Key Identifier</td></tr>
<tr><td>Authority Information Access</td><td>Laden von weiteren CA-Zertifikaten zusätzlich zu denen aus den Eingabeparametern</td></tr>
<tr><td>Benutzerparameter</td><td>–</td><td>–</td></tr>
</table>

Eigenschaften des Zertifizierungspfads	–	–
Sonstige Eingabeparameter	Eine Menge von Vertrauensankern	Konstruktion des Pfades zwischen einem Vertrauensanker und dem Zielzertifikat
	Eine Menge von Zertifikaten ohne Vertrauensstatus	
	Das Zielzertifikat	

Tabelle 51: Pfadkonstruktion – Übersicht

Als Pfadkonstruktion wird die Operation bezeichnet, die aus einer vorgegebenen Menge von Vertrauensankern, dem Zielzertifikat und ggf. weiteren optionalen Eingabedaten in Form von Zertifikaten ohne Vertrauensstatus (d.h. CA-Zertifikaten, die keine Vertrauensanker darstellen) den Pfad in Form einer geordneten Liste von Zertifikaten ausgibt, an deren Anfang ein Vertrauensanker und an deren Ende das Zielzertifikat steht. Dabei gilt, wie oben dargelegt, dass eine Implementierung für den Vertrauensanker auch ein anderes Datenformat an Stelle eines X.509 Zertifikats verwenden kann.

Aus formaler Sicht ist in [RFC 5280] der Vertrauensanker nicht Bestandteil des Pfades. Diese Modellierung ist in jenem Dokument möglich, weil darin nur die Pfadvalidierung behandelt wird, in der nur ein einziger Vertrauensanker zum Einsatz kommt, und nicht die gesamte Zertifikatsprüfung, wie das im vorliegenden Dokument der Fall ist. Für die Zertifikatsprüfung lässt sich diese Modellierung nicht gut aufrechterhalten, da diese mehrere Vertrauensanker als Eingabe erhält und somit die Information, über welchen Vertrauensanker ein Zertifikat geprüft werden kann, explizit modelliert werden muss. Daher wird im vorliegenden Dokument der Vertrauensanker als Teil des Pfades betrachtet.

Für jedes Zertifikat in dem Pfad, mit Ausnahme des Zielzertifikats, gilt, dass der Inhalt seines subject-DN (*subject*) gleich dem des Inhaltes des issuer-DN (*issuer*) des von ihm ausgestellten Zertifikats sein muss. Die Inhalte der beiden Felder müssen binär übereinstimmen. Zwar gibt es in [RFC 5280] auch davon abweichende, permissivere Regeln für den Vergleich von internationalisierten Namen, dort ist aber gleichzeitig festgelegt, dass Implementierungen nur den binären Vergleich zwingend (MUST) umsetzen müssen.

Wenn für ein Zertifikat mehrere Ausstellerzertifikate auf Basis des Vergleichs von subject-DN und issuer-DN in Frage kommen, dann kann es passieren, dass die Pfadvalidierung für jeden der dadurch entstehenden Pfade aufgerufen wird um den tatsächlich gültigen zu finden. Eine solche Situation kann beispielsweise entstehen, wenn für die gleiche CA nacheinander verschiedene Zertifikate mit dem gleichen subject-DN ausgestellt wurden. In diesem Fall sollte die Authority Key Identifier Erweiterung des ausgestellten Zertifikats verwendet werden, um einen eindeutigen Bezug zum Ausstellerzertifikat über dessen Subject Key Identifier Erweiterung herzustellen (siehe Abschnitt 2.3.7). Auf diese Weise wird in solchen Situationen die Anzahl der zu validierenden Pfade reduziert.

Sollte die Implementierung feststellen, dass ihr zur Vervollständigung eines Pfades das nächste Ausstellerzertifikat als Eingabeparameter fehlt, so besteht die Möglichkeit, dass die Implementierung die Authority Information Access Erweiterung (siehe Abschnitt 2.3.9) auswertet. Wenn diese Erweiterung mindestens einen Eintrag für *caIssuers* enthält, dann wird an dem darin angegebenen Speicherort das Ausstellerzertifikat erwartet. Zu diesem Zweck muss die Implementierung in der Lage sein, auf den jeweiligen Speicherort zugreifen zu können.

Tabelle 51 gibt eine zusammenfassende Übersicht über alle hier aufgeführten Aspekte der Pfadkonstruktion.

In komplexen Public-Key Infrastrukturen kann die effiziente Pfadkonstruktion zu einer Aufgabe werden, die nach optimierten Implementierungen verlangt. [RFC 4158] behandelt dieses Thema ausführlich.

Bezogen auf Sicherheitsaspekte stellt die Pfadkonstruktion eine weitgehend unkritische Operation dar, sofern diese in der Implementierung strikt von der Pfadvalidierung abgegrenzt werden kann. Wichtige Kriterien sind, dass jedes Zertifikat im konstruierten Pfad nur einmal vorkommen darf, und dass in diesem Pfad ein Vertrauensanker am Anfang steht. Ferner wird empfohlen, die noch etwas stärkere Einschränkung zu verwenden, dass jedes Paar von *subject* und dem über das Zertifikat zugeordneten öffentlichen Schlüssel nur einmal im Pfad vorkommen darf ([RFC 4158], Abschnitt 2.4.2).

Falls kein Pfad konstruiert werden kann, ist die Zertifikatsprüfung fehlgeschlagen.

8.4 Pfadvalidierung

Nach der erfolgten Konstruktion wird die Validierung des Zertifizierungspfades durchgeführt. Die Anforderungen an gültige Pfade sind in [RFC 5280] nur implizit definiert, nämlich dadurch, dass ein Pfad genau dann gültig ist, wenn er den formalen Algorithmus in Abschnitt 6.1 von [RFC 5280] erfolgreich durchläuft. Eine Möglichkeit bei Erstellung einer Implementierung zur Pfadvalidierung ist es, sich strikt an diesem formalen Algorithmus zu orientieren.

Als Alternative wird hier eine stärker gegliederte Spezifikation der Pfadvalidierung gegeben, indem einzelne Aspekte der Prüfung in separaten Algorithmen bearbeitet werden. Dies kann dazu beitragen, das Verständnis und die Übersicht beim Umgang mit den Vorgaben zu erhöhen. In Abschnitt 8.4.1 wird zunächst die Pfadvalidierung mit dem in diesem Dokument empfohlenen Mindestumfang an Funktionalität spezifiziert. Implementierungen, welche die Pfadvalidierung mit diesem Funktionsumfang unterstützen, sind für viele Einsatzzwecke bereits ausreichend.

In den nachfolgenden Abschnitten wird dieser Funktionsumfang um die erweiterte Verarbeitung von Zertifikatsrichtlinien (Certificate Policies) in Abschnitt 8.4.2 und Namensbeschränkungen (Name Constraints) in Abschnitt 8.4.3 erweitert. In diesen Abschnitten erfolgt allerdings keine vollständige explizite Spezifikation der dazu notwendigen Prüfungen, sondern es wird erläutert, welche Schritte aus dem formalen Algorithmus integriert werden müssen, um die Funktionalität umzusetzen.

Der formale Algorithmus zur Pfadvalidierung aus [RFC 5280] definiert eine Reihe von Eingabeparametern. Tabelle 52 listet diejenigen Parameter, die im Zusammenhang mit der Verarbeitung von Zertifikatsrichtlinien und Namensbeschränkungen stehen, und gibt jeweils an, ob die Verwendung des Parameters von der Pfadvalidierung mit Mindestfunktionalität, wie in Abschnitt 8.4.1 beschrieben, oder den Varianten der Pfadvalidierung mit erweitertem Funktionsumfang aus den daran anschließenden Abschnitten unterstützt wird. Ein Eintrag „–“ bedeutet, dass die entsprechende Funktionserweiterung die Verarbeitung des Parameters nicht beeinflusst. Alle anderen von [RFC 5280] definierten Eingabeparameter werden von der Pfadvalidierung mit Mindestfunktionalität unterstützt. Diese werden im entsprechenden Unterabschnitt vollständig aufgelistet.

Parameter	Funktion	Mindest-funktionalität	Erweiterung „Policy-Verarbeitung“	Erweiterung „Name Constraints“
user-initial-policy-set	Eine Menge von Policies, deren Unterstützung der Benutzer von dem zu prüfenden Zertifikat verlangt. Falls der Benutzer hier keine Einschränkungen vorgibt, enthält diese Menge genau die anyPolicy.	ja	–	–
initial-policy-	Kann die Werte „wahr“ oder „falsch“ annehmen. Im Falle von „wahr“ werden	nein	ja	–

mapping-inhibit	die Effekte von Policy Mappings Erweiterungen in dem Pfad unterdrückt.			
initial-explicit-policy	Kann die Werte „wahr“ oder „falsch“ annehmen. Im Falle von „wahr“ muss bei der Pfadprüfung eine der Policies im user-initial-policy-set für das Zielzertifikat gültig sein.	ja	ja	–
initial-any-policy-inhibit	Kann die Werte „wahr“ oder „falsch“ annehmen. Im Fall von „wahr“, wird die spezielle anyPolicy bei der Pfadvalidierung nicht als gültige Policy betrachtet und ignoriert.	nein	ja	–
initial-permitted-subtrees	Hiermit werden Einschränkungen an erlaubte Namen in subject oder in der Subject Alternative Name Erweiterung definiert. Wenn im Zertifizierungspfad Zertifikate vorkommen, deren entsprechende Felder diese Einschränkungen nicht erfüllen, schlägt die Validierung fehl.	nein	–	ja
initial-excluded-subtrees	Hiermit werden, analog zu der Funktion von initial-permitted-subtrees, Einschränkungen an erlaubte Namen in Form von explizit verbotenen Namensräumen spezifiziert. Wenn im Zertifizierungspfad Zertifikate vorkommen, deren entsprechende Felder diese Einschränkungen nicht erfüllen, schlägt die Validierung fehl.	nein	–	ja

Tabelle 52: Unterstützung der Pfadvalidierungsparameter nach [RFC 5280]

8.4.1 Pfadvalidierung mit Unterstützung der Mindestfunktionalität

Im Folgenden wird die Pfadvalidierung mit Unterstützung der essentiellen Funktionalitäten beschrieben: Darunter fallen alle Prüfungen der Zertifikatsfelder und der Anforderungen an die Erweiterungen für CA-Zertifikate, sowie die Auswertung der Certificate Policies Erweiterung. Die hierfür notwendigen Prüfungen werden selbstkonsistent beschrieben. Es wird angegeben, welche Erweiterungen auf welche Weise verarbeitet werden müssen. Es wird ferner beschrieben, wie mit Zertifikaten umzugehen ist, die Erweiterungen enthalten, die nicht unterstützt werden.

Wie oben erläutert, ist es von der Implementierung abhängig, ob diese das Zertifikat eines Vertrauensankers als Teil des Pfades den gleichen Prüfungen unterzieht, die für die Ausstellerzertifikate im Pfad vorgegeben sind. Es wird allerdings von dieser Spezifikation empfohlen, dass die diesbezügliche Einstufung des Vertrauensankers einheitlich für alle Prüfungen erfolgen muss. Wenn also in der folgenden Spezifikation Bezug genommen wird auf „alle Zertifikate im Pfad“, beinhaltet dies entweder immer den Vertrauensanker, oder niemals. Falls der Vertrauensanker in einer Prüfung explizit ausgeschlossen werden soll, so wird dies entsprechend angegeben.

8.4.1.1 Eingabeparameter

Die Eingabeparameter für die Pfadvalidierung sind die folgenden:

- Der Zertifizierungspfad, inklusive Vertrauensanker am Anfang und Zielzertifikat am Ende, welcher validiert werden soll.
- Das aktuelle Datum mit Uhrzeit.
- Die formalen Parameter nach [RFC 5280] (siehe Tabelle 52)
  - *user-initial-policy-set*
  - *initial-explicit-policy*
- Optional eine Menge von Sperrlisten, eine Menge von OCSP-Antworten.

Bezüglich des letzten Punktes ist zu bemerken, dass bestimmte Zertifikatserweiterungen URLs enthalten können, von denen die unter diesem Punkt aufgeführten Objekte abgerufen werden können. Somit ist es im Allgemeinen notwendig, dass die Routinen zur Pfadvalidierung direkt oder indirekt auf diese URLs zugreifen können. Dieser Aspekt wird bei der Vorstellung der Pfadkonstruktion näher erläutert.

8.4.1.2 Syntaktische Prüfung

Das Dekodieren eines Zertifikats impliziert eine syntaktische Überprüfung des ASN.1 Formats. Es ist wichtig, dass hierbei gilt:

- Die Routinen sind robust gegenüber falsch kodierten Zertifikaten und auch in diesem Fall ist eine sichere Verarbeitung der Daten gewährleistet (siehe Abschnitt 8.1.2).
- In dem Fall, dass das Zertifikat nicht in dem geforderten Format vorliegt, muss die Verarbeitung mit einem Fehler abgebrochen werden.

8.4.1.3 Prüfung der Version

Bezeichnung	Prüfung der Zertifikatsversion	
Beschreibung	Es muss geprüft werden, ob das Zertifikat eine von der Implementierung unterstützte Version hat, und ob dessen Struktur konform zu den Vorgaben für die jeweilige Version ist.	
Anzuwenden auf	alle Zertifikate im Pfad	
Referenzen	Abschnitt 4.1.2.1 in [RFC 5280]	
Typ des Eingabeparameters	**Verwendete Parameter**	**Verwendung**
Basisfelder des Zertifikats	*version*	Prüfung auf gültige und unterstützte Version
Zertifikatserweiterungen	Feld *extensions*	Falls das Feld *extensions* vorhanden ist: Prüfung ob dieses von der ermittelten Version unterstützt wird
Benutzerparameter	–	
Eigenschaften des Zertifizierungspfads	–	
Sonstige Eingabeparameter	–	

Tabelle 53: Prüfung der Zertifikatsversion – Übersicht

Für das Feld *version* muss geprüft werden, ob es eine der erlaubten Versionen ist, und ob die Version konsistent ist mit weiteren Zertifikatseigenschaften. Erlaubte Werte sind dabei

- Version 1, kodiert als der Integer 0. Die Unterstützung dieser Version ist optional. In diesem Fall darf das Zertifikat keine Zertifikatserweiterungen enthalten.
- Version 2, kodiert als der Integer 1. Die Unterstützung dieser Version ist optional. In diesem Fall darf das Zertifikat keine Zertifikatserweiterungen enthalten.
- Version 3, kodiert als der Integer 2. Die Unterstützung dieser Version ist notwendig.

Es wird empfohlen, sofern möglich, nur Version-3-Zertifikate zu unterstützen. Im Fall von Version-1- und Version-2-Zertifikaten gibt es keinen generischen Weg, ein Zertifikat als CA-Zertifikat zu klassifizieren, was eine erhebliche Sicherheitsproblematik mit sich bringt.

8.4.1.4 Prüfung der kritischen Zertifikatserweiterungen

Bezeichnung	Prüfung der kritischen Zertifikatserweiterungen	
Beschreibung	Es muss sichergestellt werden, dass das Zertifikat keine als kritisch markierte Erweiterung enthält, deren Verarbeitung von der Implementierung nicht unterstützt wird.	
Anzuwenden auf	alle Zertifikate im Pfad	
Referenzen	Abschnitt 4.2 in [RFC 5280]	
Typ des Eingabeparameters	**Verwendete Parameter**	**Verwendung**
Basisfelder des Zertifikats	–	
Zertifikatserweiterungen	Feld *extensions*	Falls das Feld *extensions* vorhanden ist, muss die Prüfung der Kritikalität und Unterstützung durch die Implementierung für jede darin enthaltene Erweiterung durchgeführt werden
Benutzerparameter	–	
Eigenschaften des Zertifizierungspfads	–	
Sonstige Eingabeparameter	–	

Tabelle 54: Prüfung der kritischen Zertifikatserweiterungen – Übersicht

Für jedes Zertifikat im Pfad muss überprüft werden, dass es keine als kritisch markierte Zertifikatserweiterung enthält, die nicht von der Implementierung unterstützt wird. Dieses Verhalten ist im Allgemeinen am besten dadurch zu realisieren, dass bei der Verarbeitung des Zertifikats über alle Erweiterungen iteriert wird, für unterstützte Erweiterungen der jeweilige Code-Abschnitt zu deren Verarbeitung angesprungen wird und in dem Sonderfall einer nicht unterstützten Erweiterung, die als kritisch markiert ist, ein Fehler angezeigt wird. Von einer expliziten, separaten Liste mit unterstützten und nicht unterstützten Erweiterungen wird abgeraten, da ein solches Vorgehen leicht zu Inkonsistenzen führen kann.

8.4.1.5 Prüfung der Namensverkettung

Bezeichnung	Prüfung der Namensverkettung	
Beschreibung	Es muss geprüft werden, ob die Verkettung von *subject* und *issuer* aller Zertifikate im Pfad korrekt ist.	
Anzuwenden auf	alle Zertifikate im Pfad	
Referenzen	Abschnitt 4.1.2.4 in [RFC 5280]	
Typ des Eingabeparameters	**Verwendete Parameter**	**Verwendung**
Basisfelder des Zertifikats	*subject*, *issuer*	Prüfung der korrekten Namensverkettung
Zertifikatserweiterungen	–	
Benutzerparameter	–	
Eigenschaften des Zertifizierungspfads	–	
Sonstige Eingabeparameter	–	

Tabelle 55: Prüfung der Namensverkettung – Übersicht

Die korrekte Verkettung der Namen in den Zertifikaten des Pfades muss geprüft werden. In dem zu prüfenden Pfad müssen *subject* in einem Zertifikat und *issuer* in dem nachfolgenden Zertifikat binär identisch sein. Alternativ darf die Implementierung die Regeln zur Validierung von internationalisierten Namen laut [RFC 5280], Abschnitt 7.1, anwenden. Diese Prüfung kann auch bereits in der Pfadkonstruktion erfolgen, so dass sie für die Pfadvalidierung unnötig wird.

8.4.1.6 Prüfung der Zertifikatssignaturen

Bezeichnung	Prüfung der Zertifikatssignaturen	
Beschreibung	Es muss geprüft werden, ob die Signaturen aller Zertifikate im Pfad korrekt sind.	
Anzuwenden auf	alle Zertifikate im Pfad außer dem Vertrauensanker	
Referenzen	Abschnitt 6.1 in [RFC 5280]	
Typ des Eingabeparameters	**Verwendete Parameter**	**Verwendung**
Basisfelder des Zertifikats	*signature, signatureValue, subjectPublicKeyInfo*	Prüfung, ob die Signatur in *signatureValue* mit dem in *signature* angegebenen Verfahren und dem öffentlichen Schlüssel des Ausstellers in dessen *subjectPublicKeyInfo* verifiziert werden kann
Zertifikatserweiterungen	–	

Benutzerparameter	–	
Eigenschaften des Zertifizierungspfads	–	
Sonstige Eingabeparameter	–	

Tabelle 56: Prüfung der Zertifikatssignaturen – Übersicht

Für jedes Zertifikat im Pfad außer dem Vertrauensanker muss die Signatur mit dem öffentlichen Schlüssel des Ausstellers geprüft werden. Dazu wird zunächst der Signaturalgorithmus im *signature*-Feld ausgewertet. Anschließend wird die Signatur im *signatureValue*-Feld mit dem ermittelten Signaturverfahren und dem öffentlichen Schlüssel des Ausstellers geprüft. Der öffentliche Schlüssel ist im Allgemeinen im *subjectPublicKeyInfo*-Feld des jeweiligen Zertifikats enthalten. Da der Vertrauensanker, wie in Abschnitt 8.2.1 dargelegt, nicht zwangsläufig im Zertifikatsformat vorliegt, kann hier der öffentliche Schlüssel in einem anderen Format gegeben sein.

8.4.1.7 Prüfung der zeitlichen Gültigkeit

Bezeichnung	Prüfung der zeitlichen Gültigkeit	
Beschreibung	Es muss geprüft werden, ob der aktuelle Zeitpunkt der Zertifikatsprüfung innerhalb der Gültigkeitsdauer eines Zertifikats liegt.	
Anzuwenden auf	alle Zertifikate im Pfad	
Referenzen	Abschnitt 6.1 in [RFC 5280]	
Typ des Eingabeparameters	**Verwendete Parameter**	**Verwendung**
Basisfelder des Zertifikats	*validity*	Bestimmung des Gültigkeitszeitraums
Zertifikatserweiterungen	–	
Benutzerparameter	–	
Eigenschaften des Zertifizierungspfads	–	
Sonstige Eingabeparameter	Aktueller Zeitpunkt	Prüfung gegen den Gültigkeitszeitraum des Zertifikats

Tabelle 57: Prüfung der zeitlichen Gültigkeit – Übersicht

Für jedes Zertifikat im Pfad muss überprüft werden, ob es zum Zeitpunkt der Prüfung gültig ist. Diese Prüfung erfolgt, indem der aktuelle Zeitpunkt mit den Gültigkeitsdaten im Zertifikat verglichen wird. Es werden folgende Prüfungen durchgeführt:

1. Falls der aktuelle Zeitpunkt vor *notBefore* liegt, so ist das Zertifikat ungültig.
2. Falls der aktuelle Zeitpunkt nach *notAfter* liegt, so ist das Zertifikat ungültig.

Es sei darauf hingewiesen, dass ein Fehler bei der Prüfung der zeitlichen Gültigkeit eines Zertifikats zu erheblichen Sicherheitsbeeinträchtigungen führen kann. Dies ist auch dann der Fall, wenn ein Zertifikat nur

in einem geringfügig zum tatsächlichen Gültigkeitszeitraum verschobenen Zeitfenster fälschlicherweise als gültig geprüft wird. Der Grund dafür ist, dass ein Zertifikat sowohl im Falle der Verwendung von Sperrlisten als auch bei Verwendung des OCSP Protokolls zuverlässig nur innerhalb seines Gültigkeitszeitraums von diesen Diensten explizit als gesperrt gekennzeichnet wird. Somit besteht im Falle einer in diesem Punkt fehlerhaften Implementierung die Gefahr, dass gesperrte Zertifikate fälschlicherweise als nicht gesperrt geprüft werden. Allerdings sehen sowohl OCSP mit der „Archive Cutoff" Erweiterung (Abschnitt 4.4.4. in [RFC 6960]) als auch Sperrlisten mit der „Expired certificates on CRL" Erweiterung (Abschnitt 9.5.2.8 in [X.509/16]) Mechanismen vor, um Zertifikate auch nach Ablauf der Gültigkeit weiter zu berücksichtigen.

8.4.1.8 Prüfung der CA-Zertifikate

Bezeichnung	CA-Zertifikatsprüfung	
Beschreibung	Es wird für alle CA-Zertifikate im Pfad geprüft, ob diese die für CA-Zertifikate notwendigen Kennzeichen tragen	
Anzuwenden auf	alle Zertifikate im Pfad außer dem Zielzertifikat	
Referenzen	Abschnitte 4.2.1.3 und 4.2.1.9 in [RFC 5280]	
Typ des Eingabeparameters	**Verwendete Parameter**	**Verwendung**
Basisfelder des Zertifikats	*version*	Prüfung ob es sich um ein Version-3-Zertifikat handelt und somit die auf den Zertifikatsinhalten basierenden Prüfungen möglich sind
Zertifikatserweiterungen	Basic Constraints	Prüfung, ob das Zertifikat in der Erweiterung als CA-Zertifikat gekennzeichnet ist
	Key Usage	Prüfung, ob das Zertifikat in der Erweiterung als Aussteller von Zertifikaten gekennzeichnet ist
Benutzerparameter	–	
Eigenschaften des Zertifizierungspfads	–	
Sonstige Eingabeparameter	–	

Tabelle 58: CA-Zertifikatsprüfung – Übersicht

Für jedes CA-Zertifikat im Pfad außer dem Zielzertifikat muss geprüft werden, ob es sich um ein gültiges CA-Zertifikat handelt. Für die jeweiligen X.509 Versionen müssen jeweils die aufgelisteten Prüfungen durchgeführt werden. Schlägt eine der Prüfungen fehl, so ist die Pfadvalidierung fehlgeschlagen.

1. Prüfung von Version 1 oder Version 2 Zertifikaten. Sofern die Implementierung diese Version von X.509 Zertifikaten unterstützt, muss auf durch den Einsatzkontext vorgegebenen Wegen sichergestellt werden, dass es sich um ein CA-Zertifikat handelt. Ein möglicher Ansatz ist es, nur solche Version-1- oder Version-2-Zertifikate als CA-Zertifikate zu klassifizieren, die zuvor über einen kryptographischen Hashwert des Zertifikats als solche in der Implementierung hinterlegt wurden.

2. Prüfung von Version-3-Zertifikaten. Es müssen die nachfolgend aufgelisteten Eigenschaften geprüft werden.
   a. Das Zertifikat enthält eine Basic Constraints Erweiterung.
   b. In der Basic Constraints Erweiterung ist das Feld *cA* auf den Wert *true* gesetzt.
   c. Falls das Zertifikat eine Key Usage Erweiterung enthält, dann ist in dieser der Verwendungszweck *keyCertSign* gesetzt.

8.4.1.9 Prüfung der Pfadlängenbeschränkung

Bezeichnung	Prüfung der Pfadlängenbeschränkung	
Beschreibung	Es muss für jede innerhalb des Zertifizierungspfades vorkommende Basic Constraints Erweiterung geprüft werden, ob die ggf. darin definierte Pfadlängenbeschränkung erfüllt ist.	
Anzuwenden auf	alle Zertifikate im Pfad außer dem Zielzertifikat	
Referenzen	Abschnitt 4.2.1.9 und 6.1 in [RFC 5280]	
Typ des Eingabeparameters	**Verwendete Parameter**	**Verwendung**
Basisfelder des Zertifikats	–	
Zertifikatserweiterungen	Basic Constraints	Bestimmung der Pfadlängenbeschränkung, die für im Pfad nachfolgende Zertifikate erzwungen werden muss
Benutzerparameter	–	
Eigenschaften des Zertifizierungspfads	Anzahl der dem Zertifikat im Pfad nachfolgenden CA-Zertifikate, die nicht selbst ausgestellt (self-issued) sind	
Sonstige Eingabeparameter	–	

Tabelle 59: Prüfung der Pfadlängenbeschränkung – Übersicht

Um die Erfüllung aller Pfadlängenbeschränkungen in dem zu validierenden Pfad zu prüfen, ist wie folgt vorzugehen.

Für jedes Zertifikat im Pfad außer dem Zielzertifikat muss geprüft werden, ob für das Zertifikat eine Pfadlängenbeschränkung gesetzt ist, und ob diese von dem vorliegenden Pfad verletzt wird.

Für jedes dieser Zertifikate wird die geltende Pfadlängenbeschränkung *d* ermittelt. Im Falle des Vertrauensankers kann diese Information je nach Implementierung entweder über die Basic Constraints Erweiterung des Zertifikats des Vertrauensankers oder über mit dem Vertrauensanker assoziierte Informationen gegeben sein, allerdings sind Implementierungen nach [RFC 5280] nicht gezwungen, die Verwendung dieser Inhalte eines Vertrauensankers bzw. diesem zugeordneter Parameter zu unterstützen.

Für die Zertifikate im Pfad ist diese Information jeweils der Basic Constraints Erweiterung zu entnehmen. Fehlt diese Erweiterung bei einem V3-Zertifikat, muss das Zertifikat als CA-Zertifikat abgelehnt werden. Eine Pfadlängenbeschränkung gilt dann, wenn in dieser Erweiterung das Feld *pathLenConstraint* gesetzt ist. In diesem Fall wird *d* der dort angegebene Wert zugewiesen, andernfalls der besondere Wert unbeschränkt.

Nun wird ermittelt, ob die Pfadlängenbeschränkungen von allen zu prüfenden Zertifikaten im Pfad erfüllt werden. Zusätzlich zu den genannten Fällen, müssen Zertifikate, für die der Wert *d* unbeschränkt ist, ebenfalls nicht auf die Erfüllung der Pfadlängenbeschränkung geprüft werden. Um die Erfüllung der Pfadlängenbeschränkungen für ein Zertifikat zu überprüfen, wird die Zahl *z* ermittelt, welche die Anzahl der Zertifikate angibt, die dem in Prüfung befindlichen Zertifikat im Pfad folgen. Selbst ausgestellte (self-issued) Zertifikate werden dabei nicht zur Zahl z dazugerechnet. Wenn z größer als *d+1* ist, dann ist die Pfadlängenbeschränkung für das in Prüfung befindliche Zertifikat verletzt, andernfalls ist sie erfüllt.

8.4.1.10 Überprüfung der Zertifikatsrichtlinien

Bezeichnung	Eingeschränkte Prüfung der Zertifikatsrichtlinien	
Beschreibung	Prüfung der Konsistenz der in den Certificate Policies Erweiterungen der Zertifikate im Pfad ausgewiesenen Richtlinien gegen die vom Benutzer vorgegebenen Bedingungen. Die hier beschriebene eingeschränkte Prüfung bezieht die Verarbeitung folgender Zertifikatserweiterungen nicht mit ein: Policy Constraints, Inhibit anyPolicy, Policy Mappings. Siehe dazu Abschnitt 8.4.2.	
Anzuwenden auf	alle Zertifikate im Pfad	
Referenzen	Abschnitt 4.2.1.4 in [RFC 5280]	
Typ des Eingabeparameters	**Verwendete Parameter**	**Verwendung**
Basisfelder des Zertifikats	–	
Zertifikatserweiterungen	Certificate Policies	Prüfung, ob vom Benutzer geforderte Policy enthalten ist
		Prüfung, ob als kritisch markiert. Wenn ja, Prüfung, ob alle enthaltenen Policies inklusive Qualifiern interpretiert werden können
Benutzerparameter	*initial-explicit-policy*	Falls *initial-explicit-policy* den Wert „wahr“ hat, muss *user-initial-policy-set* mindestens eine Policy enthalten, die von jedem Zertifikat im Pfad erfüllt wird
	user-initial-policy-set	
Eigenschaften des Zertifizierungspfads	–	
Sonstige Eingabeparameter	–	

Tabelle 60: Eingeschränkte Prüfung der Zertifikatsrichtlinien – Übersicht

Die Prüfung der Zertifikatsrichtlinien, welche ein optionaler Bestandteil in Form der Certificate Policies Erweiterung sind, ist notwendig, wenn eine der beiden folgenden Bedingungen zutrifft:

- Beim Aufruf der Pfadvalidierung hat der Benutzer die Variable *initial-explicit-policy* gesetzt.
- Eine Zertifikatsrichtlinie ist als kritisch markiert.

Im Folgenden werden diese beiden Prüfungen, die unabhängig voneinander durchgeführt werden müssen, erläutert.

1. Prüfung der vom Benutzer vorgegebenen Policies

Wenn der Benutzer in der Variablen *initial-explicit-policy* für die Pfadvalidierung den Wert „wahr“ gesetzt hat, dann müssen folgende Prüfungen durchgeführt werden:

- Jedes Zertifikat im Pfad, mit Ausnahme des Vertrauensankers, enthält die Certificate Policies Erweiterung.
- Es muss mindestens eine Policy aus dem *user-initial-policy-set*, referenziert über einen entsprechenden OID, geben, die in jedem Zertifikat im Pfad, mit Ausnahme des Vertrauensankers, in dessen Certificate Policies Erweiterung enthalten ist. Dabei wird die *anyPolicy* als gleich zu jeder beliebigen Policy betrachtet.

2. Prüfung von kritischen Zertifikatsrichtlinien

Falls das Zertifikat eine als kritisch markierte Certificate Policies Erweiterung enthält, dann muss die Implementierung in der Lage sein, die Erweiterung vollständig zu interpretieren. Dies bedeutet, dass auch alle potentiell vorhandenen Policy Qualifier interpretiert werden können. Falls dies nicht der Fall ist, muss das Zertifikat abgelehnt werden.

Von [RFC 5280] werden zwei grundlegende Policy Qualifier definiert: der „CPS Pointer“ und die „User Notice“. Ersterer bezieht sich auf das „Certification Practice Statement“, ein Dokument, welches die Handhabung und Richtlinien der jeweiligen CA erläutert. Der CPS Pointer verweist auf dieses Dokument. Letzterer beinhaltet eine Meldung, die dem Benutzer vor der Verwendung des validierten Zertifikats angezeigt werden soll. Wie die Verarbeitung dieser beiden Arten von Feldern zu erfolgen hat, hängt davon ab in welchem Anwendungskontext die Implementierung eingesetzt wird.

8.4.2 Pfadvalidierung mit erweiterter Funktionalität zur Verarbeitung von Zertifikatsrichtlinien

Die in Abschnitt 8.4.1 beschriebene Pfadvalidierung kann die im Zusammenhang mit Zertifikatsrichtlinien (Certificate Policies) stehenden Zertifikatserweiterungen Policy Mappings, Policy Constraints und Inhibit anyPolicy nicht verarbeiten. Falls eine Unterstützung dieser Erweiterungen in der Implementierung notwendig ist, so muss die Erstellung der Implementierung anhand des formalen Algorithmus zur Pfadvalidierung aus [RFC 5280], Abschnitt 6.1 erfolgen. Lediglich die Schritte zur Verarbeitung der Felder *permittedSubtrees* und *excludedSubtrees* der Name Constraints Erweiterung brauchen nicht umgesetzt zu werden, solange die Erweiterung des Algorithmus zur Pfadvalidierung entsprechend Abschnitt 8.4.3 nicht umgesetzt wird. Dabei kann auch eine Unterstützung der relevanten Benutzerparameter entsprechend Tabelle 52 umgesetzt werden.

8.4.3 Pfadvalidierung mit erweiterter Funktionalität zur Verarbeitung der Name Constraints Zertifikatserweiterung

Die in Abschnitt 8.4.1 beschriebene Pfadvalidierung unterstützt nicht die Verarbeitung der Name Constraints Erweiterung. Falls diese Funktionalität in einer Implementierung benötigt wird, so müssen die entsprechenden Schritte des formalen Algorithmus in [RFC 5280] umgesetzt werden. Diese umfassen die Schritte (b) und (c) unter Abschnitt 6.1.3 sowie (g).(1) und (g).(2) unter Abschnitt 6.1.4 jeweils in [RFC 5280]. Ferner muss die korrekte Initialisierung der beteiligten formalen Variablen *permitted_subtrees* und *excluded_subtrees* zu Beginn der Pfadvalidierung erfolgen. Dabei kann auch eine Unterstützung der relevanten Benutzerparameter entsprechend Tabelle 52 umgesetzt werden.

8.5 Prüfung des Revokationsstatus

Für jedes Zertifikat im Pfad muss dessen Revokationsstatus geprüft werden. Die Prüfung des Revokationsstatus eines Zertifikats kann prinzipiell auf zwei Wegen erfolgen:

- über Sperrlisten (CRLs) oder
- über OCSP-Antworten.

Im Falle von Sperrlisten gibt es noch die Unterscheidung zwischen vollständigen Sperrlisten und Delta-Sperrlisten (Delta CRLs). In allen Fällen erfolgt die Identifikation eines Zertifikats in den Sperrinformationen über dessen Ausstellernamen (*issuer*) und seine Seriennummer (*serialNumber*).

Im nachfolgenden Abschnitt 8.5.2 wird nur eine eingeschränkte Prüfung unter Verwendung von vollständigen Sperrlisten beschrieben, was die Mindestanforderung von [RFC 5280] an Implementierungen darstellt. Allerdings wird gegenüber den Anforderungen von [RFC 5280] für die Prüfung mittels vollständiger Sperrlisten eine Vereinfachung vorgenommen, durch welche aber die typischen Anwendungsfälle nicht beeinträchtigt werden. Die nicht unterstützte Funktionalität betrifft die Verarbeitung von Sperrlisten, die Sperrinformationen nur für eine Untermenge der möglichen Sperrgründe enthalten. In diesem Fall wäre die Kombination mehrerer Sperrlisten, die zusammengenommen alle Sperrgründe abdecken, notwendig. Die Ausstellung von Sperrlisten mit dieser Eigenschaft ist jedoch untypisch, und von [RFC 5280] wird in Abschnitt 4.2.1.13 gefordert („MUST“), dass eine CRL Distribution Points Erweiterung in einem Zertifikat mindestens auf eine Sperrliste verweist, welche alle Sperrgründe abdeckt. Die in Abschnitt 8.5.2 spezifizierte Revokationsprüfung kann nicht mit Zertifikaten umgehen, deren CRL Distribution Points Erweiterung in diesem Sinne nicht konform ist. Anders ausgedrückt wird diese Revokationsprüfung fehlschlagen, also das Zertifikat abweisen, wenn nicht mindestens eine alle Sperrgründe abdeckende Sperrliste für das Zertifikat verfügbar ist.

Wie die Implementierung anzupassen ist, um auch die Verwendung von Sperrlisten für eine Untermenge der möglichen Sperrgründe, von Delta-Sperrlisten und OCSP-Antworten zu unterstützen, wird in den Abschnitten 8.5.3, 8.5.4, und 8.5.5 beschrieben.

Zunächst wird eine Erläuterung des Formats von Sperrlisten gegeben.

8.5.1 Das Format von Sperrlisten

Das Format von Sperrlisten ist, wie auch das für X.509 Zertifikate, von [RFC 5280] vorgegeben. Es hat auf oberster Ebene die folgende Struktur, die analog zu der von X.509 Zertifikaten ist.

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

Das Feld *tbsCertList* hat folgenden Aufbau:

```
TBSCertList ::= SEQUENCE {
    version                 Version OPTIONAL, -- if present, MUST be v2
    signature               AlgorithmIdentifier,
    issuer                  Name,
    thisUpdate              Time,
    nextUpdate              Time OPTIONAL,
    revokedCertificates     SEQUENCE OF SEQUENCE {
        userCertificate         CertificateSerialNumber,
        revocationDate          Time,
        crlEntryExtensions      Extensions OPTIONAL -- if present, version MUST be v2
                                } OPTIONAL,
```

```
    crlExtensions        [0]  EXPLICIT Extensions OPTIONAL -- if present, version MUST be v2
}
```

Die Felder *version*, *signature* und *issuer* haben die gleiche Bedeutung wie die entsprechend benannten Felder in einem X.509 Zertifikat. Das Feld *thisUpdate* kennzeichnet den Zeitpunkt, an dem die Sperrliste ausgestellt wurde und auf den sich die Sperrinformationen beziehen. Der Zeitpunkt, zu dem spätestens eine neue Sperrliste verfügbar sein wird, ist in *nextUpdate* angegeben. Das Feld *revokedCertificates* enthält eine Liste mit Einträgen, von denen jeder ein gesperrtes Zertifikat identifiziert, und zwar über das darin enthaltene Feld *userCertificate*, welches die Seriennummer des betroffenen Zertifikats angibt. Ferner enthält jeder Eintrag das Feld *revocationDate*, welches den Revokationszeitpunkt angibt. Optional kann ein Eintrag auch das Feld *crlEntryExtensions* (Sperrlisteneintragserweiterungen) enthalten, welches weitere Informationen zum gesperrten Zertifikat, wie z.B. den Sperrgrund, angeben kann. Das Feld *crlExtensions* kann analog zu den Zertifikatserweiterungen in dem Feld *extensions* in einem X.509 Zertifikat sogenannte Sperrlistenerweiterungen enthalten. Für die Prüfung relevante Sperrlistenerweiterungen sind die Issuing Distribution Point und die Delta CRL Indicator Erweiterung.

Die Issuing Distribution Point Sperrlistenerweiterung (IDP) stellt das Gegenstück zur CRL Distribution Points Zertifikatserweiterung (CDP, siehe Abschnitt 2.3.5) dar und erlaubt eine Zuordnung der Sperrliste zu einem der Distribution Points in der CDP.

Die Delta CRL Indicator Sperrlistenerweiterung kennzeichnet die Sperrliste als eine Delta-Sperrliste, die nur differentielle Sperrinformationen zu einer vollständigen Sperrliste, die den Normalfall darstellt, enthält. Die Verarbeitung von Delta-Sperrlisten wird in Abschnitt 8.5.4 behandelt.

Sowohl Sperrlistenerweiterungen als auch Sperrlisteneintragserweiterungen können wie Zertifikatserweiterungen als kritisch markiert sein. Eine Sperrliste, die eine von der Implementierung nicht unterstützte kritische Erweiterung einer der beiden Arten enthält, darf nicht zur Bestimmung des Revokationsstatus irgendeines Zertifikats verwendet werden.

8.5.2 Prüfung des Revokationsstatus mit Mindestfunktionalität

Bezeichnung	Überprüfung des Revokationsstatus mittels vollständiger Sperrlisten, die alle Sperrgründe abdecken	
Beschreibung	Der Revokationsstatus aller Zertifikate wird bestimmt, indem lokal vorhandene und von ausgewiesenen Speicherorten zu ladende vollständige Sperrlisten ausgewertet werden.	
Anzuwenden auf	alle Zertifikate im Pfad (jedoch nicht auf den Vertrauensanker)	
Referenzen	Abschnitt 6.3 in [RFC 5280]	
Typ des Eingabeparameters	**Verwendete Parameter**	**Verwendung**
Basisfelder des Zertifikats	*serialNumber, issuer*	Prüfung, ob das über die Seriennummer und den Aussteller identifizierte Zertifikat auf einer Sperrliste vorhanden ist
Zertifikatserweiterungen	CRL Distribution Points	Bestimmung der Speicherorte von aktuellen Sperrlisten, die in der Prüfung verwendet werden bzw. Zuordnung einer Sperrliste zu einem Distribution Point

		Bestimmung des Sperrlistenausstellers im Falle von indirekten Sperrlisten
		Bestimmung, ob es sich um eine Sperrliste für eine Untermenge von Sperrgründen handelt
	Basic Constraints	Bestimmung, ob es sich um ein CA-Zertifikat handelt oder nicht
	Key Usage	Bestimmung, ob der Aussteller der Sperrliste den korrekten Schlüsselverwendungszweck im Zertifikat eingetragen hat
Benutzerparameter	–	
Eigenschaften des Zertifizierungspfads	–	
Sonstige Eingabeparameter	Lokal vorhandene Sperrlisten	Prüfung des Revokationsstatus des Zertifikats
	Zeitpunkt der Zertifikatsprüfung	Prüfung ob die verwendeten Sperrlisten gültig sind

Tabelle 61: Prüfung des Revokationsstatus – Übersicht

Im Rahmen der Zertifikatsprüfung muss jedes Zertifikat im Pfad, ausgenommen der Vertrauensanker, darauf überprüft werden, ob es gesperrt ist oder nicht. Zu diesem Zweck muss der Algorithmus aus [RFC 5280], Abschnitt 6.3, für jedes Zertifikat im Pfad, mit Ausnahme des Vertrauensankers durchgeführt werden. In diesem Abschnitt wird allerdings eine vereinfachte Variante des dortigen Algorithmus spezifiziert, welche für die überwiegende Zahl der Anwendungsszenarien ausreichend ist. Die in dieser Variante realisierten Vereinfachungen sind:

- Keine Unterstützung von Sperrlisten für eine Untermenge von Sperrgründen. Sofern für ein Zertifikat zu [RFC 5280] konforme Sperrlisten ausgestellt werden, wird die Revokationsprüfung durch diese Vereinfachung nicht beeinträchtigt.
- Keine Unterstützung von Delta-Sperrlisten. Bei Delta-Sperrlisten handelt es sich um ein wenig verbreitetes Format für Sperrinformationen, deren Verarbeitung in [RFC 5280] als optional spezifiziert ist.

Wie diese Vereinfachungen gegenüber dem in [RFC 5280], Abschnitt 6.3, spezifizierten, im Folgenden auch als „RFC Sperrstatus Algorithmus“ bezeichnet, genau umzusetzen sind, wird im Unterabschnitt 8.5.2.1 definiert. In den Abschnitten 8.5.3 und 8.5.4 werden Varianten der Revokationsprüfung spezifiziert, die diese Vereinfachungen jeweils aufheben.

Nach der Spezifikation der Vereinfachungen werden in Abschnitt 8.5.2.2 wichtige Sicherheitshinweise zur Implementierung der Revokationsprüfung gegeben.

8.5.2.1 Umsetzung der Vereinfachungen

- Die Vereinfachungen des Algorithmus zur Prüfung des Revokationsstatus mit Mindestfunktionalität gegenüber dem RFC Sperrstatus Algorithmus werden wie folgt umgesetzt. Keine Unterstützung von Sperrlisten für eine Untermenge von Sperrgründen. Hierzu werden die Schritte (d), (e) und (l) des RFC

Sperrstatus Algorithmus ersetzt. Statt wie in diesen Schritten spezifiziert, die Kombination der Sperrgründe von den einzelnen verarbeiteten Sperrlisten zu berechnen, wird stattdessen
  - jede Sperrliste von der Verarbeitung ausgeschlossen, in deren IDP das Feld *onlySomeReasons* nur eine Untermenge aller möglichen Sperrgründe spezifiziert
  - und jede Sperrliste von der Verarbeitung ausgeschlossen, deren zugeordnetes DistributionPoint-Element in der CDP im Zertifikat nur eine Untermenge aller möglichen Sperrgründe spezifiziert. Eine Möglichkeit dies umzusetzen besteht darin, auf den Zugriff auf eine solche Sperrliste zu verzichten.
- Keine Unterstützung von Delta-Sperrlisten. Dies wird umgesetzt, indem der formale Eingabeparameter des RFC Sperrstatus Algorithmus *use-deltas* auf den Wert „falsch“ gesetzt wird.

8.5.2.2 Zusätzliche Prüfungen zum Algorithmus aus RFC 5280

Im Folgenden werden Hinweise zur Implementierung des RFC Sperrstatus Algorithmus, bzw. der hier spezifizierten vereinfachten Variante, gegeben. Die in diesem Zusammenhang vorgegebenen Prüfungen sind teilweise an anderer Stelle von [RFC 5280] gefordert, finden sich jedoch nicht explizit in der algorithmischen Beschreibung, und sind teilweise in [RFC 5280] gar nicht erwähnt.

1. Überprüfung der Key Usage Zertifikatserweiterung des Sperrlistenausstellers. Falls im Zertifikat des Sperrlistenausstellers, welches in Schritt (f) des RFC Sperrstatus Algorithmus geprüft wird, die Key Usage Erweiterung enthalten ist, so muss geprüft werden, dass in dieser mindestens der Verwendungszweck *cRLSign* gesetzt ist.
2. Die Delta CRL Indicator Sperrlistenerweiterung darf in einer vom RFC Sperrstatus Algorithmus geprüften Sperrliste nicht enthalten sein. Diese Einschränkung folgt aus der Tatsache, dass von der hier spezifizierten Variante keine Delta-Sperrlisten verarbeitet werden.
3. Der aktuelle Zeitpunkt für die Revokationsprüfung muss als identisch mit demjenigen gewählt werden, der auch bei der zeitlichen Gültigkeitsprüfung des Zertifikats in der Pfadvalidierung (siehe Abschnitt 8.4.1.7) verwendet wurde.
4. Das Feld *nextUpdate* muss in der Sperrliste vorhanden sein und wie im RFC Sperrstatus Algorithmus vorgegeben geprüft werden. In der Spezifikation der ASN.1 Struktur einer Sperrliste ist dieses zwar als optional gekennzeichnet, von [RFC 5280] wird in Abschnitt 5.1.2.5 das Vorhandensein dieses Feldes allerdings als zwingend vorgegeben.
5. Im Falle einer indirekten Sperrliste, d.h. bei einer Sperrliste, deren Aussteller nicht mit dem Aussteller des Zertifikats übereinstimmt, muss berücksichtigt werden, dass diese Sperrliste Zertifikate enthalten kann, die von verschiedenen Ausstellern ausgestellt wurden. Um den Revokationsstatus eines Zertifikats richtig zu bestimmen, müssen die Regeln für die Zuordnung eines Sperreintrags zu einem Zertifikatsaussteller beachtet werden. Diese Regeln, in [RFC 5280], Abschnitt 5.3.3 dargelegt, sind wie folgt: Jeder Sperreintrag (ein Element von revokedCertificates), der die Sperrlisteneintragserweiterung Certificate Issuer enthält, spezifiziert den Zertifikatsaussteller für das darin referenzierte Zertifikat. Für jeden Sperreintrag, der diese Erweiterung nicht enthält, gilt der zuletzt in einem vorhergehenden Eintrag festgelegte Aussteller. Falls der erste Sperreintrag die Certificate Issuer Erweiterung nicht enthält, dann gilt für diesen und die nachfolgenden Einträge bis zum ersten Auftreten dieser Erweiterung der Sperrlistenaussteller (Feld *issuer* der Sperrliste) als Zertifikatsaussteller.

8.5.3 Erweiterung der Revokationsprüfung um die Verarbeitung von Sperrlisten für eine Untermenge der möglichen Sperrgründe

Der Algorithmus aus Abschnitt 8.5.2 verarbeitet keine Sperrlisten, die in der Issuing Distribution Point Sperrlistenerweiterung im Feld *onlySomeReasons* oder in dem ihnen zugeordneten DistributionPoint-Element in der CRL Distribution Points Zertifikatserweiterung nicht alle Sperrgründe ausweisen. Für die korrekte Verarbeitung solcher Sperrlisten muss der Algorithmus um die Funktionalität erweitert werden, zu erkennen, wann von den bisher im Rahmen der Prüfung des Revokationsstatus verarbeiteten Sperrlisten in Kombination alle Sperrgründe abgedeckt werden.

Die Verwendung dieser Funktionalität ist dann sinnvoll, wenn der Umgang mit nicht-konformen Sperrlistenausstellern notwendig ist, die sich nicht an die Vorgabe aus [RFC 5280] halten, für jedes Zertifikat mindestens eine Sperrliste auszustellen, welche alle Sperrgründe abdeckt.

Der Algorithmus zur Sperrlistenverarbeitung aus [RFC 5280], Abschnitt 6.6.3, beinhaltet die Funktionalität zur Verarbeitung von Sperrlisten, die nur eine Untermenge von Sperrgründen abdecken. Eine Implementierung muss diese Funktionalität wie dort spezifiziert umsetzen.

8.5.4 Erweiterung der Revokationsprüfung um die Verarbeitung von Delta-Sperrlisten

Delta-Sperrlisten stellen ein alternatives Format von Sperrlisten dar. Der Vorteil von Delta-Sperrlisten gegenüber vollständigen Sperrlisten ist der, dass Delta-Sperrlisten nur differentielle Sperrinformationen bezogen auf eine bestimmte vollständige Sperrliste des gleichen Sperrlistenausstellers enthalten. Im Falle einer großen Zahl von gesperrten Zertifikaten sind die von Delta-Sperrlisten gebildeten Dateien deutlich kleiner als die von vollständigen Sperrlisten, was den Ladevorgang und deren Verarbeitung unter bestimmten Bedingungen vereinfacht.

Um diese Funktionalität in einer Implementierung umzusetzen, wird der in [RFC 5280], Abschnitt 6.3 spezifizierte Algorithmus umgesetzt, wobei der formale Eingabeparameter *use-deltas* den Wert „wahr“ annimmt. Ferner ist Abschnitt 5.2.4 aus [RFC 5280] zu berücksichtigen.

Bezogen auf den in Abschnitt 8.5.2 spezifizierten, vereinfachten Algorithmus zur Bestimmung des Revokationsstatus ergibt sich bei der Unterstützung von Delta-Sperrlisten folgende Änderung. Statt wie unter Punkt 2 in Abschnitt 8.5.2.2 gefordert Delta-Sperrlisten zurückzuweisen, wird das Vorhandensein der Delta CRL Indicator Sperrlistenerweiterung zur Klassifizierung der vorliegenden Sperrliste als Delta-Sperrliste verwendet. Es ist essentiell, dass diese Klassifizierung bei jeder geladenen Sperrliste nur anhand der Delta CRL Indicator Sperrlistenerweiterung vorgenommen wird, und nicht etwa implizit über die Quelle, von welcher diese geladen wurde.

Die Freshest CRL Erweiterung kann sowohl in Zertifikaten (siehe Abschnitt 2.4.1) enthalten sein als auch in Sperrlisten. Sie wird in [RFC 5280], Abschnitt 4.2.1.15 definiert und ist völlig analog aufgebaut zu der CRL Distribution Points Erweiterung. Der einzige Unterschied ist der, dass diese Erweiterung auf Delta-Sperrlisten verweist statt auf vollständige Sperrlisten. Sofern Delta-Sperrlisten verarbeitet werden, so muss diese Erweiterung in Zertifikaten und Sperrlisten ausgewertet werden um die Speicherorte für Delta-Sperrlisten zu bestimmen.

8.5.5 Erweiterung der Revokationsprüfung um die Verarbeitung von OCSP Antworten

Das Online Certificate Status Protocol (OCSP) [RFC 6960] stellt eine Alternative für die Revokationsprüfung eines Zertifikats gegenüber Sperrlisten dar. Es handelt sich um ein einfaches Protokoll, bei dem eine Anwendung, welche ein Zertifikat prüft, eine Anfrage an einen OCSP-Responder nach dem Revokationsstatus stellt. Als Antwort erhält sie entweder eine Fehlermeldung oder eine kryptographisch signierte definitive Aussage über den Revokationsstatus des Zertifikats.

Die Verwendung von OCSP-Antworten für die Revokationsprüfung setzt voraus, dass die Adresse eines OCSP-Responders bekannt ist, der Auskünfte für das geprüfte Zertifikat erteilt. Eine solche Adresse kann das Zertifikat in der Authority Information Access Erweiterung angeben (siehe Abschnitt 2.3.9), die im Falle der Unterstützung von Revokation über OCSP ausgewertet werden muss.

Bei der Verwendung von OCSP für die Revokationsprüfung muss ein zu [RFC 6960] konformer OCSP-Client zum Einsatz kommen. Folgende Aspekte sind bei der Verwendung des OCSP Protokolls besonders hervorzuheben:

1. Ein OCSP-Responder liefert auf eine Anfrage entweder eine definitive Antwort über den Revokationsstatus des Zertifikats oder eine Fehlerantwort. Eine definitive Antwort kennzeichnet das Zertifikat entweder als unbekannt („unknown“), gesperrt („revoked“) oder nicht gesperrt („good“) und trägt eine mittels dem OCSP-Signer-Zertifikat verifizierbare Signatur. Eine Fehlerantwort ist dagegen nicht signiert. Eine Anwendung darf ein Zertifikat nur dann als nicht gesperrt betrachten, wenn sie die signierte Antwort „good“ erhalten hat.
2. Vom OCSP-Responder signierte Antworten können von dem gleichen Schlüssel signiert sein wie dem, welcher das geprüfte Zertifikat ausstellt. Andernfalls muss der Zertifikatsaussteller diese Aufgabe an einen anderen Schlüssel delegieren, indem er ein entsprechendes OCSP-Signer-Zertifikat ausstellt. Ein solches Zertifikat ist dadurch gekennzeichnet, dass es die Extended Key Usage Erweiterung mit dem Zweck *id-kp-OCSPSigning* enthält (Abschnitt 4.2.2.2, [RFC 6960]).
3. Falls die OCSP-Antworten nicht direkt von dem Zertifikatsaussteller signiert werden, muss prinzipiell auch der Revokationsstatus des OCSP-Signer-Zertifikats geprüft werden. Hierbei gibt es folgende Möglichkeiten (siehe [RFC 6960], Abschnitt 4.2.2.2.1):
    a. In dem OCSP-Signer-Zertifikat kann die Zertifikatserweiterung *id-pkix-ocsp-nocheck* gesetzt sein. Damit ist festgelegt, dass keine Prüfung des Revokationsstatus dieses Zertifikats erfolgen soll.
    b. Falls das OCSP-Signer-Zertifikat die CRL Distribution Points Erweiterung enthält, dann muss dessen Revokationsstatus ermittelt werden, und die Verwendung der darin spezifizierten Sperrlisten ist mindestens eine der Möglichkeiten dafür.
    c. Falls das OCSP-Signer-Zertifikat die Authority Information Access Erweiterung enthält, und dort eine OCSP Adresse spezifiziert wird, dann muss der Revokationsstatus ermittelt werden, und die Verwendung des entsprechenden OCSP-Responders ist mindestens eine der Möglichkeiten dafür. Allerdings muss eine Implementierung in diesem Zusammenhang Sorge dafür tragen, dass sie durch fehlerhaft ausgestellte Zertifikate nicht in eine Endlosschleife versetzt werden kann, bei der wiederholt versucht wird, für den gleichen OCSP-Signer wiederum eine OCSP-Auskunft einzuholen.
    d. Falls keine der in den vorhergehenden Punkten aufgeführten Kriterien im OCSP-Signer-Zertifikat erfüllt sind, ist es der Anwendung überlassen, ob und wie sie dessen Revokationsstatus prüft.
4. Wie auch im Falle der Sperrlistenprüfung muss für die Prüfung von OCSP-Antworten als aktueller Zeitpunkt der gleiche Zeitpunkt verwendet werden wie derjenige, der für die Bestimmung des Gültigkeitszeitraums des geprüften Zertifikats (siehe Abschnitt 8.4.1.7) verwendet wird. Speziell muss geprüft werden, dass der Prüfungszeitpunkt später liegt als der im Feld *thisUpdate* der OCSP-Antwort ausgewiesene Zeitpunkt, aber gleichzeitig früher liegt als der Zeitpunkt in deren Feld *nextUpdate*.

Im TLS Protokoll besteht die Möglichkeit, das sogenannte „OCSP Stapling“ einzusetzen. Dies bedeutet, dass der TLS Server selbst OCSP Antworten sowohl zu seinem eigenen Zertifikat als auch zu den weiteren Zertifikaten in dessen Zertifizierungspfad vorhält und diese unter Verwendung der Certificate Status TLS Erweiterung innerhalb der TLS Nachrichten an den Client schickt [RFC 6961]. Somit wird es dem Client erspart, selbst auf den OCSP Responder zuzugreifen. Es wird empfohlen, wo immer möglich OCSP Stapling zu verwenden, um den Kommunikationsaufwand und somit Latenzen und Protokollabbrüche so gering wie möglich zu halten.

8.6 Anwendungsspezifische Zertifikatsprüfungen

Entsprechend des Verwendungszwecks des Zertifikats müssen prinzipiell weitere spezifische Prüfungen am Zertifikat vorgenommen werden.

8.6.1 Validierung von TLS Server Zertifikaten

Bei der Validierung eines TLS Server Zertifikats muss die Eignung des im Zertifikat enthaltenen Schlüssels für das in dem TLS Handshake zum Einsatz kommende Schlüsselaustauschverfahren überprüft werden und eine Überprüfung des Namens des Servers erfolgen.

8.6.1.1 Prüfung der Schlüsselverwendungszwecke

Falls die Key Usage Erweiterung in dem Zertifikat des Servers gesetzt ist, muss diese im Falle von TLS 1.2, entsprechend den Vorgaben von [RFC 5246] geprüft werden. Eine Übersicht über die Voraussetzungen für die Verwendung eines Zertifikats in einem bestimmten Schlüsselaustauschverfahren ist in Tabelle 33 zu finden. Im Fall von TLS 1.3 [RFC 8446] gilt die Anforderung, dass im Falle einer gesetzten Key Usage Erweiterung der Verwendungszweck *digitalSignature* gesetzt sein muss.

Falls im Zertifikat die Extended Key Usage Erweiterung im Zertifikat gesetzt ist, so muss geprüft werden, dass darin mindestens der Verwendungszweck *serverAuth* oder *anyExtendedKeyUsage* gesetzt ist.

8.6.1.2 Prüfung des DNS Namens

Um sicherzustellen, dass das geprüfte Zertifikat tatsächlich für den Server ausgestellt wurde, zu dem die TLS Verbindung aufgebaut werden soll, muss dessen Name gegen die Zertifikatsinhalte geprüft werden.

Die dafür relevanten Regeln sind in [RFC 6125], Abschnitt 6, angegeben. Wir geben im Folgenden eine Zusammenstellung der dabei wichtigsten Aspekte:

1. Falls in dem Zertifikat die Subject Alternative Name Erweiterung enthalten ist und mindestens einen Eintrag
   a. vom Typ *dNSName*,
   b. vom Typ *iPAddress*,
   c. vom Typ *otherName*, welcher einen *SRVName*[6] beinhaltet,
   d. oder vom Typ *uniformResourceIdentifier*, welcher sowohl einen Protokoll- als auch Hostanteil enthält,

   so kann der Inhalt eines beliebigen dieser Einträge verwendet werden, um den Server zu identifizieren. Falls im Anwendungskontext andere oder weitere Typen von Einträgen als zur Identifikation eines Servers zulässig definiert sind, so können auch diese verwendet werden.
2. Der Server gilt als identifiziert, wenn der Inhalt eines der zuvor aufgezählten Typen von Einträgen in der Subject Alternative Name enthalten ist und mit dem Namen des Servers übereinstimmt.
3. Nur wenn keiner der zuvor aufgezählten Typen von Einträgen in der Subject Alternative Name Erweiterung enthalten ist, und auch kein anderer im jeweiligen Anwendungskontext interpretierbarer Eintrag darin enthalten ist, darf die Anwendung den Inhalt des Feldes *subject* des Zertifikats zur Identifikation des Servers verwenden. Es wird allerdings empfohlen, die Verwendung des Feldes *subject* zur Identifikation des Servers nicht zu unterstützen.
4. Der Vergleich von Namen muss bei allen oben genannten möglichen Identifikationsmethoden unempfindlich gegenüber Groß-/Kleinschreibung erfolgen.
5. Falls in dem Anwendungskontext internationalisierte Domain Namen verwendet werden, so gelten die weiteren Regeln in [RFC 6125], Abschnitt 6.4.1.
6. TLS Server Zertifikate können im DNS Namen den Platzhalter („Wildcard Character“) „*“ enthalten.
   a. Es wird empfohlen, einen Namensvergleich unter Verwendung von Platzhaltern in einer Implementierung nicht zu unterstützen. Für eine Übersicht über die Gründe sei auf [RFC 6125], Abschnitt 7.2, verwiesen. Falls eine Implementierung Platzhalter unterstützt, dann gilt die

[6] Ein SRVName bezieht sich auf einen SRV DNS Resource Record, welcher Dienste zu einer Domain auflistet [RFC 2782]. Die Verwendung von *SRVName* in X.509 Zertifikaten wird in [RFC 4985] definiert.

Empfehlung, dass dieser nur als erstes Zeichen eines DNS Namens in *dNSName* oder in *subject* vorkommen soll und nur ein einziges Label repräsentieren kann.

b. Ferner ist bei der Validierung eines Domain-Namen, der einen Platzhalter enthält, zu beachten, dass dabei keine Zertifikate für ICANN-Domains, unterhalb denen Nutzer eigene Domains registrieren können, akzeptiert werden. Neben den Top-Level Domains („.de“, „.com“, etc.) umfasst dies je nach Nation ggf. weitere (Beispiel. „.co.uk“). Für private Domains, welche als Public Suffix registriert sind, sind Wildcards erlaubt.[7]

7. In IP-Adressen müssen immer vollständige Adressen spezifiziert sein, Adressbereiche dürfen nicht zur Identifikation eines Servers verwendet werden.

8.6.2 Validierung von TLS Client Zertifikaten

Bei der Validierung von TLS Client Zertifikaten müssen die im Zertifikat gesetzten Schlüsselverwendungszwecke geprüft werden.

Falls die Key Usage Erweiterung in dem Zertifikat gesetzt ist, muss diese im Falle von TLS 1.2 entsprechend den Vorgaben von [RFC 5246] umgesetzt werden. Eine Übersicht über die Voraussetzungen für die Verwendung eines Zertifikats in einem bestimmten Schlüsselaustauschmechanismus ist in Tabelle 38 zu finden.

Im Fall von TLS 1.3 werden grundsätzliche keine Anforderungen an die Key Usage Erweiterung im Client Zertifikat gestellt. Jedoch besteht in dieser Protokollversion die Möglichkeit, dass der Server von sogenannten OID Filtern Gebrauch macht, wie in Abschnitt 5.1.2 ausgeführt. Dadurch können auch Anforderungen an die in der Key Usage Erweiterung des Client Zertifikats gesetzten Schlüsselverwendungszwecke gestellt werden, die dann vom Server entsprechend überprüft werden müssen.

Falls im Zertifikat die Extended Key Usage Erweiterung gesetzt ist, so wird empfohlen zu prüfen, dass darin mindestens der Verwendungszweck *clientAuth* oder *anyExtendedKeyUsage* gesetzt ist.

8.6.3 Validierung von S/MIME Zertifikaten

Bei der Validierung von S/MIME Zertifikaten muss geprüft werden, dass der im Zertifikat gesetzte Verwendungszweck des öffentlichen Schlüssels zum intendierten Einsatzzweck des Zertifikats (Verschlüsselung oder Signatur) passt und dass das Zertifikat dem richtigen E-Mail Konto zugeordnet ist.

8.6.3.1 Prüfung der Schlüsselverwendungszwecke

Falls das Zertifikat die Key Usage Erweiterung enthält, und für die Verifikation von S/MIME Signaturen verwendet werden soll, dann muss es mindestens einen der Schlüsselverwendungszwecke *digitalSignature* oder *nonRepudiation* gesetzt haben ([RFC 5750], Abschnitt 4.4.2).

Falls das Zertifikat die Key Usage Erweiterung enthält, und für die Verschlüsselung von S/MIME Nachrichten verwendet werden soll, dann gelten keine bindenden Regeln für die erforderlichen gesetzten Schlüsselverwendungszwecke. Falls die Implementierung die Zertifikatsverwendung auch im Falle der Nachrichtenverschlüsselung einschränken soll, so kann sie mindestens einen der Schlüsselverwendungszwecke *keyAgreement* oder *keyEncipherment* fordern. In jedem Fall sollten für eine Verschlüsselungsoperation bei mehreren zur Auswahl stehenden Zertifikaten, die der entsprechenden E-Mail-Adresse zugeordnet sind, diejenigen bevorzugt werden, welche einen der Schlüsselverwendungszwecke *keyAgreement* oder *keyEncipherment* gesetzt haben.

[7] Eine nicht offizielle, aber weithin genutzte Liste mit solchen Domains, deren Vollständigkeit nicht garantiert ist, ist unter https://publicsuffix.org/ verfügbar.

Falls im Zertifikat die Extended Key Usage Erweiterung gesetzt ist, so muss diese sowohl für Verschlüsselung als auch Signatur von S/MIME Nachrichten mindestens einen der Zwecke *emailProtection* oder *anyExtendedKeyUsage* gesetzt haben.

8.6.3.2 Prüfung der E-Mail-Adresse

Die E-Mail-Adresse des Benutzers, von dem eine signierte Nachricht verifiziert oder an den eine Nachricht verschlüsselt werden soll, kann auf zwei verschiedene Arten im Zertifikat bezeichnet werden: Entweder explizit oder über Namensauflösung.

Explizite E-Mail-Adressen können im Zertifikat in der Subject Alternative Name Erweiterung in einem Eintrag des Typs *rfc822Name* ([RFC 5750], Abschnitt 4.4.3) oder im Feld *subject* in einem PKCS#9 *emailAddress* Attribut ([RFC 5750], Abschnitt 3) enthalten sein.

Falls keine explizite E-Mail-Adresse im Zertifikat gesetzt ist, kann die Implementierung auf externe Informationen zurückgreifen, über welche dem im Feld *subject* des Zertifikats enthaltenen Namen zuverlässig eine E-Mail-Adresse zugeordnet werden kann. Dies kann beispielsweise über ein authentisches Verzeichnis geschehen.

Die so aus dem Zertifikat bestimmte E-Mail-Adresse muss im Falle einer Signaturverifikation mit dem Inhalt des „From“ oder „Sender“ Feldes des E-Mail-Headers übereinstimmen. Im Falle einer Nachrichtenverschlüsselung muss sie mit der Empfängeradresse übereinstimmen, für welche die Nachricht erzeugt wird. Im Falle mehrerer Empfänger muss die Zertifikatsprüfung für jede Empfängeradresse vorgenommen werden.

8.6.4 Validierung von Zertifikaten für IPsec Knoten

Bei der Validierung von Zertifikaten für IPsec Knoten müssen sowohl die Serveradresse als auch die Schlüsselverwendungszwecke überprüft werden. Ferner legt IPsec gegenüber [RFC 5280] engere Regeln fest, wie nicht unterstützte kritische Zertifikatserweiterungen verarbeitet werden müssen.

8.6.4.1 Prüfung der Schlüsselverwendungszwecke

Sofern die Key Usage Erweiterung im Zertifikat gesetzt ist, muss für die Verwendung des Zertifikats innerhalb von IKE oder IKEv2 geprüft werden, dass mindestens einer der beiden Verwendungszwecke *digitalSignature* oder *nonRepudiation* [RFC 4945] in der Key Usage Erweiterung gesetzt ist.

8.6.4.2 Prüfung des Namens des Peers

Die Prüfung der Adresse des Peers, zu dem die Verbindung aufgebaut wird, kann in IPsec über verschiedene ID Types, die im IPsec ID Payload verschickt werden dürfen, erfolgen [RFC 4945]. Tabelle 62 dokumentiert den Zusammenhang zwischen den verschiedenen ID Typen und dem jeweils zugeordneten Zertifikatsinhalt, mit dem der Inhalt der ID übereinstimmen muss.

ID Type	PKIX Attribut	Übereinstimmung im Zertifikat	Weitere Prüfungen
IP*_ADDR	SAN: iPAddress	ID muss binär mit der IP-Adresse im Zertifikat übereinstimmen. Falls in der SAN mehrere IP-Adressen eingetragen sind, so muss eine übereinstimmen.	IP-Adresse muss mit derjenigen im IP Header übereinstimmen.
FQDN	SAN: dNSName	ID muss unempfindlich gegenüber Groß-/ Kleinschreibung mit dNSName verglichen werden. Dabei darf kein auf „Substring“, „Wildcard“ oder regulären Ausdrücken basierender Vergleich durchgeführt werden.	-

ID Type	PKIX Attribut	Übereinstimmung im Zertifikat	Weitere Prüfungen
USER_FQDN	SAN: rfc822Name	ID muss unempfindlich gegenüber Groß-/ Kleinschreibung mit rfc822Name verglichen werden. Dabei darf kein auf „Substring“, „Wildcard“ oder regulären Ausdrücken basierender Vergleich durchgeführt werden.	-
DN	*subject*	ID muss binär mit dem Inhalt von *subject* übereinstimmen. Das Feld *subject* darf nicht leer sein.	-

Tabelle 62: Zusammenhang von IPsec ID Payload und Zertifikatsinhalten

Die ID Typen aus den ersten drei Zeilen müssen jeweils mit dem in der zweiten Spalte angegebenen Eintrag der Subject Alternative Name Erweiterung (SAN) des Zertifikats übereinstimmen. Die vierte Zeile behandelt eine ID vom Typ DN, welche mit dem Inhalt des Feldes *subject* des Zertifikats binär übereinstimmen muss. Für die ID Typen in den ersten drei Zeilen muss für eine erfolgreiche Prüfung eine vollständige Übereinstimmung des Namens erzielt werden. Der Vergleich auf Basis der Gleichheit von Teilen der Namen („Substring“), unter Verwendung von Platzhaltern („Wildcard“) oder regulären Ausdrücken ist nicht erlaubt.

8.6.4.3 Verarbeitung der Zertifikatserweiterungen

Gegenüber [RFC 5280] spezifiziert IPsec in [RFC 4945] in Abschnitt 5.1.3 für einige Fälle genauer, wie die Reaktion auf von der Implementierung nicht unterstützte Zertifikatserweiterungen erfolgen muss. Hier gilt zunächst, dass ein Zertifikat wie in [RFC 5280] gefordert abgelehnt werden muss. Darüber hinaus müssen Zertifikate aber auch dann abgelehnt werden, wenn sie nicht unterstützte Erweiterungen enthalten, die zwar im Zertifikat nicht als kritisch markiert sind, aber laut den Anforderungen aus [RFC 5280] als kritisch markiert sein müssten.

8.7 Übersicht über die Verarbeitung von Zertifikatserweiterungen bei der Zertifikatsprüfung

Tabelle 63 listet alle in diesem Dokument behandelten Zertifikatserweiterungen auf und gibt jeweils an, von welchem Teil der Zertifikatsprüfung diese verarbeitet werden bzw. unterstützt werden. Ein Eintrag „ja“ bedeutet, dass die entsprechende Zertifikatserweiterung in dem jeweiligen Algorithmus verarbeitet wird. Ein Eintrag „–“ bedeutet, dass die entsprechende Erweiterung für den Algorithmus nicht relevant ist. Dabei sind die Basisalgorithmen zur Pfadkonstruktion, Pfadvalidierung mit Mindestfunktionalität, Überprüfung des Revokationsstatus mit Mindestfunktionalität und die anwendungsspezifischen Prüfungen als zentrale Prüfungen jeweils grau hinterlegt. In diesen Spalten wird, sofern von den jeweils geltenden Standards die Verarbeitung einer Erweiterung in dem entsprechenden Teilvorgang vorgesehen ist, aber von den hier spezifizierten Algorithmen diese nicht verarbeitet wird, „nein“ eingetragen. Ein Eintrag „ja“ in einer der darauffolgenden (d.h. weiter rechts stehenden) weiß hinterlegten Spalten mit erweiterten Algorithmen zu dem vorherigen Basisalgorithmus bedeutet, dass durch Verwendung der erweiterten Algorithmen die Funktionalität abgedeckt ist.

Zertifikatserweiterung	Pfadkonstruktion (Abschnitt 8.3)	Pfadvalidierung mit Mindestfunktionalität (Abschnitt 8.4.1)	Erweiterte Pfadvalidierung „Policy-Verarbeitung“ (Abschnitt 8.4.2)	Erweiterte Pfadvalidierung „Name Constraints“ (Abschnitt 8.4.3)	Revokationsprüfung mit Mindestfunktionalität (Abschnitt 8.5.2)	Revokationsprüfung mit Delta-Sperrlisten (Abschnitt 8.5.4)	Revokationsprüfung mit OCSP (Abschnitt 8.5.5)	Anwendungsspezifische Prüfung
Basic Constr.	–	ja	–	–	ja	–	–	–
Key Usage	–	ja	–	–	ja	–	–	ja
Ext. Key Usage	–	–	–	–	–	–	ja	ja
Subj. Alt. Name	–	–	–	–	–	–	–	ja
CRL Distrib. Points	–	–	–	–	ja	–	–	–
Auth. Key Id.	ja	–	–	–	–	–	–	–
Subj. Key Id.	ja	–	–	–	–	–	–	–
Authority Inf. Access	ja[8]	–	–	–	nein	–	ja[9]	–
Freshest CRL	–	–	–	–	nein	ja	–	–
Cert. Policies	–	ja	–	–	–	–	–	–
Policy Mappings	–	nein	Ja	–	–	–	–	–
Policy Constraints	–	nein	Ja	–	–	–	–	–
Inhibit Any Policy	–	nein	ja	–	–	–	–	–
Name Constraints	–	nein	–	ja	–	–	–	–
Subj. Dir. Attrib.	–	nein[10]	–	–	–	–	–	–

Tabelle 63: Unterstützung der Zertifikatserweiterungen in der Zertifikatsprüfung

9 Das Certification Path Validation Test Tool

In diesem Abschnitt wird erläutert, wie das Testwerkzeug „Certification Path Validation Test Tool“ [11] (CPT) dazu verwendet werden kann, um Implementierungen der Zertifizierungspfadvalidierung hinsichtlich der in diesem Dokument spezifizierten Teilprüfungen zu testen.

Bei dem CPT handelt es sich im Kern um ein Tool zur Erzeugung von Testfällen in Form von Zertifizierungspfaden. In den im Lieferumfang des CPT enthalten Testfällen finden sich sowohl fehlerfreie Zertifikatspfade, die von einer korrekten Implementierung akzeptiert werden müssen, als auch fehlerbehaftete Pfade, die entsprechend zurückzuweisen sind. Die Testfälle des CPT, d.h. im Wesentlichen die Zertifikate, welche die zu prüfenden Pfade bilden, werden in Form von XML-Dateien spezifiziert. Aufgabe des Basis-Tools des CPT ist die Erstellung konkreter Testzertifikate aus diesen Daten. Ferner existieren Erweiterungen des Tools zur Durchführung der Tests gegen bestimmte Software-Bibliotheken, welche die Pfadvalidierung implementieren, sowie gegen TLS-, IPsec-, und E-Mail-Anwendungen. Das Tool unterstützt ferner die Prüfung von Sperrinformationen mittels Sperrlisten und OCSP.

Die Spezifikation der mit dem Tool mitgelieferten Testfälle ist beschrieben in [CPT-S]. Auf der Webseite zum CPT finden sich Verweise auf weitere Dokumente zum CPT und den zugehörigen Erweiterungen.

In Tabelle 64 bis Tabelle 68 werden für die einzelnen Zertifikatstypen in den ersten beiden Spalten mittels Abschnittsnummer und -titel die einzelnen Teilschritte der Pfadvalidierung referenziert. In der dritten Spalte werden jeweils die Testfälle aus den mitgelieferten Testfällen des CPT aufgelistet, welche die entsprechende Funktionalität verifizieren. Dabei bedeutet „*“ einen Platzhalter für sämtliche vorkommenden Testfallnummern.

Einzig für die Teilprüfung der Namensverkettung (Abschnitt 8.4.1.5) innerhalb der Pfadvalidierung existiert kein Test. Ferner existieren auch keine Tests für die beiden folgenden Varianten der Revokationsprüfungen:

- Verarbeitung von Sperrlisten für eine Untermenge von Sperrgründen (Abschnitt 8.5.3)
- Verarbeitung von Delta-Sperrlisten (Abschnitt 8.5.4)

Abschnitt	Überschrift	Testfälle
8.4.1.2	Syntaktische Prüfung	CERT_PATH_COMMON_05
8.4.1.3	Prüfung der Version	CERT_PATH_COMMON_11
		CERT_PATH_EXT_01
		CERT_PATH_EXT_02
8.4.1.4	Prüfung der kritischen Zertifikatserweiterungen	CERT_PATH_EXT_04
8.4.1.5	Prüfung der Namensverkettung	CERT_PATH_COMMON_02
8.4.1.6	Prüfung der Zertifikatssignaturen	CERT_PATH_CRYPT_01
		CERT_PATH_CRYPT_02
		CERT_PATH_COMMON_03
		CERT_PATH_COMMON_04

[11] Webseite des CPT:
https://www.bsi.bund.de/DE/Themen/Kryptografie_Kryptotechnologie/Kryptografie/CPT/cpt_node.html

Abschnitt	Überschrift	Testfälle
8.4.1.7	Prüfung der zeitlichen Gültigkeit	CERT_PATH_COMMON_07
		CERT_PATH_COMMON_08
		CERT_PATH_COMMON_09
		CERT_PATH_COMMON_10
8.4.1.8	Prüfung der CA-Zertifikate	CERT_PATH_EXT_06
		CERT_PATH_EXT_07
		CERT_PATH_EXT_11
8.4.1.9	Prüfung der Pfadlängenbeschränkung	CERT_PATH_EXT_08
8.4.1.10	Überprüfung der Zertifikatsrichtlinien	CERT_PATH_EXT_12
8.4.2	Pfadvalidierung mit erweiterter Funktionalität zur Verarbeitung von Zertifikatsrichtlinien	CERT_PATH_EXT_13
		CERT_PATH_EXT_14
		CERT_PATH_EXT_17
		CERT_PATH_EXT_18
		CERT_PATH_EXT_19
8.4.3	Pfadvalidierung mit erweiterter Funktionalität zur Verarbeitung der Name Constraints Zertifikatserweiterung	CERT_PATH_EXT_1
		CERT_PATH_EXT_16
		CERT_PATH_TLS_CLIENT_06
		CERT_PATH_IPSEC_06
		CERT_PATH_EMAIL_05
8.5.2	Prüfung des Revokationsstatus mit Mindestfunktionalität	CERT_PATH_CRL_*
8.5.3	Erweiterung der Revokationsprüfung um die Verarbeitung von Sperrlisten für eine Untermenge der möglichen Sperrgründe	–
8.5.4	Erweiterung der Revokationsprüfung um die Verarbeitung von Delta-Sperrlisten	–
8.5.5	Erweiterung der Revokationsprüfung um die Verarbeitung von OCSP Antworten	CERT_PATH_OCSP_*

Tabelle 64: Zuordnung von Testfällen zu Teilschritten der Pfadvalidierung – allgemein

Abschnitt	Überschrift	Testfälle
8.6.1.1	Prüfung der Schlüsselverwendungszwecke	CERT_PATH_TLS_CLIENT_04
		CERT_PATH_TLS_CLIENT_05

Abschnitt	Überschrift	Testfälle
8.6.1.2	Prüfung des DNS Namens	CERT_PATH_TLS_CLIENT_02
		CERT_PATH_TLS_CLIENT_03

Tabelle 65: Zuordnung von Testfällen zu Teilschritten der Pfadvalidierung – TLS Server Zertifikate

Abschnitt	Überschrift	Testfälle
8.6.2	Validierung von TLS Client Zertifikaten	CERT_PATH_TLS_SERVER_*

Tabelle 66: Zuordnung von Testfällen zu Teilschritten der Pfadvalidierung – TLS Client Zertifikate

Abschnitt	Überschrift	Testfälle
8.6.3.1	Prüfung der Schlüsselverwendungszwecke	CERT_PATH_EMAIL_03
		CERT_PATH_EMAIL_04
8.6.3.2	Prüfung der E-Mail-Adresse	CERT_PATH_EMAIL_02

Tabelle 67: Zuordnung von Testfällen zu Teilschritten der Pfadvalidierung – S/MIME Zertifikate

Abschnitt	Überschrift	Testfälle
8.6.4.1	Prüfung der Schlüsselverwendungszwecke	CERT_PATH_IPSEC_04
		CERT_PATH_IPSEC_05
8.6.4.2	Prüfung des Namens des Peers	CERT_PATH_IPSEC_03
8.6.4.3	Verarbeitung der Zertifikatserweiterungen	CERT_PATH_IPSEC_02

Tabelle 68: Zuordnung von Testfällen zu Teilschritten der Pfadvalidierung – IPsec Zertifikate

Literaturverzeichnis

[CPT-S] Certification Path Validation Test Tool – Test Specification, Version 1.1 vom 23.10.2018. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/CPT/CPT-Test-Tool-Specification_v1_1.pdf

[CPT-T] Certification Path Validation Test Tool, https://www.bsi.bund.de/CPT

[RFC 2119] S. Bradner: Key words for use in RFCs to Indicate Requirement Levels (RFC 2119), März 1997

[RFC 2585] R. Housley, P. Hoffman: Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP (RFC 2585), Mai 1999

[RFC 2782] Gulbrandsen, et al.: A DNS RR for specifying the location of services (DNS SRV) (RFC 2782), Februar 2000

[RFC 2818] E. Rescorla: HTTP Over TLS (RFC 2818), Mai 2000

[RFC 3279] W. Polk, R. Housley, L. Bassham: Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (RFC 3279), April 2002

[RFC 3280] R. Housley, W. Polk, W. Ford, D. Solo: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (RFC 3280), April 2002

[RFC 3647] S. Chokhani, W. Ford, R. Sabett, C. Merrill, S. Wu: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework (RFC 3647), November 2003

[RFC 4055] J. Schaad, B. Kaliski, R. Housley: Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (RFC 4055), Juni 2005

[RFC 4158] M. Cooper, Y. Dzambasow, P. Hesse, S. Joseph, R. Nicholas: Internet X.509 Public Key Infrastructure: Certification Path Building (RFC 4158), September 2005

[RFC 4262] S. Santesson: X.509 Certificate Extension for Secure/Multipurpose Internet Mail Extensions (S/MIME) Capabilities (RFC 4262), Dezember 2005

[RFC 4516] M. Smith, T. Howes: Lightweight Directory Access Protocol (LDAP): Uniform Resource Locator (RFC 4516), Juni 2006

[RFC 4945] B. Korver: The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX (RFC 4945), August 2007

[RFC 4985] S. Santesson: Internet X.509 Public Key Infrastructure Subject Alternative Name for Expression of Service Name (RFC 4985), August 2007

[RFC 5246] T. Dierks, E. Rescorla: The Transport Layer Security (TLS) Protocol Version 1.2 (RFC 5246), August 2008

[RFC 5280] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (RFC 5280), Mai 2008

[RFC 5750] B. Ramsdell, S. Turner: Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Certificate Handling (RFC 5750), Januar 2010

[RFC 5758] Q. Dang, S. Santesson, K. Moriarty, D. Brown, T. Polk: Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA (RFC 5758), Januar 2010

[RFC 5913] S. Turner, S. Chokhani: Clearance Attribute and Authority Clearance Constraints Certificate Extension (RFC 5913), Juni 2010

[RFC 5937] S. Ashmore, C. Wallace: Using Trust Anchor Constraints during Certification Path Processing (RFC 5937), August 2010

[RFC 6125] P. Saint-Andre, J. Hodges: Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS) (RFC 6125), März 2011

[RFC 6960] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP (RFC 6960), Juni 2013

[RFC 6961] Y. Pettersen: The Transport Layer Security (TLS) – Multiple Certificate Status Request Extension (RFC 6961), Juni 2013

[RFC 7230] R. Fielding, J. Reschke: Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing (RFC 7230), Juni 2014

[RFC 8446] E. Rescorla: The Transport Layer Security (TLS) Protocol Version 1.3 (RFC 8446), August 2018

[X.509/16] ITU-T: SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY / Directory, Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks, Oktober 2016

[X.680] ITU-T: SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY / OSI networking and system aspects – Abstract Syntax Notation One (ASN.1), Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation, August 2015

[X.690] ITU-T: SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY / OSI networking and system aspects – Abstract Syntax Notation One (ASN.1), Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER) , August 2015

[TR-02102-1] BSI: TR-02102-1 "Kryptographische Verfahren: Empfehlungen und Schlüssellängen", aktuell gültige Version ist 2020-01, 24.03.2020

[TR-02102-2] BSI: Technische Richtlinie TR-02102-2, Kryptographische Verfahren: Empfehlungen und Schlüssellängen, Teil 2 – Verwendung von Transport Layer Security (TLS), Version 2020-01, 28.02.2020

[TR-02102-3] BSI: Technische Richtlinie TR-02102-3, Kryptographische Verfahren: Empfehlungen und Schlüssellängen, Teil 3 – Verwendung von Internet Protocol Security (IPsec) und Internet Key Exchange (IKEv2), Version 2020-01, 31.01.2020