



PKI und X.509 Zertifikatsprofile

Beschreibung eines Konzepts zur Erstellung und Verwaltung von X.509 Zertifikaten

Editor: Olaf Rode

Dokumenten-ID: PKI und X.509 Zertifikatsprofile
Verantwortlich: Fraunhofer ISST
Status: Release
Version: 1.2.0.02
Letztes Update: 29. Februar 2008
Kategorie: Security and Privacy
Non-Normative

Copyright

Copyright 2008 © Fraunhofer-Institut für Software- and Systemtechnik (ISST), Asklepios Kliniken Verwaltungsgesellschaft mbH, Charité - Universitätsmedizin Berlin, Deutsche Krankenhausgesellschaft e.V., HELIOS Kliniken GmbH, Klinikum Dortmund gGmbH, Rhön-Klinikum AG, Sana e.med GmbH, Städtisches Klinikum München GmbH, Universitätsklinikum Aachen, Universitätsklinikum Tübingen and Vivantes GmbH Berlin. Alle Rechte vorbehalten.

Dieses Dokument und Übersetzungen, die davon angefertigt wurden, dürfen kopiert und weitergegeben werden und abgeleitete Werke, die es kommentieren, erklären, oder Hilfestellung bei der Implementierung leisten, dürfen vorbereitet, kopiert, veröffentlicht und verteilt werden, als Ganzes oder in Teilen, ohne dass hierbei Einschränkungen in irgendeiner Form bestehen; vorausgesetzt, dass die obige Urheberrechtserklärung und dieser Absatz in allen Kopien und abgeleiteten Werken enthalten sind. Dieses Dokument selbst darf nur mit schriftlichem Einverständnis der Urheber modifiziert werden. Die beschränkten Rechte, die durch obige Aussage gewährt werden, sind dauerhaft und werden von den oben genannten Urhebern, ihren Nachfolgeorganisationen und Rechtsnachfolgern nicht zurückgezogen werden. Dieses Dokument und die hierin enthaltene Information werden ohne Mängelgewähr zur Verfügung gestellt.

DAS FRAUNHOFER-INSTITUT FÜR SOFTWARE- AND SYSTEMTECHNIK (ISST), DIE ASKLEPIOS KLINIKEN VERWALTUNGSGESELLSCHAFT MBH, DIE CHARITÉ - UNIVERSITÄTSMEDIZIN BERLIN, DIE DEUTSCHE KRANKENHAUSGESELLSCHAFT E.V., DIE HELIOS KLINIKEN GMBH, DIE KLINIKUM DORTMUND GGMBH, DIE RHÖN-KLINIKUM AG, DIE SANA E.MED GMBH, DIE STÄDTISCHES KLINIKUM MÜNCHEN GMBH, DAS UNIVERSITÄTSKLINIKUM AACHEN, DAS UNIVERSITÄTSKLINIKUM TÜBINGEN UND DIE VIVANTES GMBH BERLIN UND DIE AN DER ERSTELLUNG DIESES DOKUMENTS BETEILIGTEN MITARBEITER DER GENANNTEN EINRICHTUNGEN SCHLIESSEN JEDE FORM DER HAFTUNG, OB GEÄUßERT ODER VERMUTET; AUS, DAFÜR DASS DIE VERWENDUNG DER INFORMATIONEN IN DIESEM DOKUMENT KEINE RECHTE VERLETZT; DASS SIE GEBRAUCHSTAUGLICH SIND ODER SICH FÜR EINEN SPEZIELLEN ZWECK EIGNEN.

Diese Spezifikation ist unter <http://www.fallakte.de> verfügbar.



Änderungsübersicht

Version	Datum	Seite	Bemerkungen	Bearbeiter
1.1.9.01	14.02.07	Alle	Erste Version	OR
1.1.9.02	22.03.07	Alle	Erweiterung um Dienstzertifikate	OR
1.1.9.03	27.04.07	Alle	Abgleich mit Referenzimplementierung	OR
1.1.9.04	05.06.07	Alle	Integration der Dokumente zu Profilen und PKI	JC
1.1.9.05	05.06.07	Alle	Freigabe zur Kommentierung	JC
1.2	29.02.08	Alle	Einarbeiten der Kommentare / Anpassung an Architekturänderungen	OR

Statushistorie

Status	Datum	Bemerkungen	Bearbeiter
In Erstellung	14.02.07	Erste Version	OR
Draft	05.06.07	Fertigstellung für Review	JC
PreFinal	29.02.08	PreFinal	OR
Final	29.02.08	Fertigstellung für Veröffentlichung	OR

Inhalt

1	Einleitung	6
1.1	Zielgruppe des Dokuments	6
1.2	Genutzte Standards	6
1.3	Konventionen	6
2	Zertifikatsprofile	8
2.1	CA-Zertifikat	8
2.1.1	Allgemeine Anforderungen	8
2.1.2	Zertifikatsprofil - Allgemeiner Teil	8
2.1.3	Zertifikatsprofil - Zertifikatserweiterungen	9
2.1.4	Struktur Beispielzertifikat	11
2.1.5	Zertifikatsgenerierung	12
2.2	Clientauthentisierungszertifikate	13
2.2.1	Allgemeine Anforderungen	13
2.2.2	Zertifikatsprofil - Allgemeiner Teil	13
2.2.3	Zertifikatsprofil - Zertifikatserweiterungen	15
2.2.4	Struktur Beispielzertifikat	17
2.2.5	Zertifikatsgenerierung	17
2.3	SSL-Serverzertifikate	18
2.3.1	Allgemeine Anforderungen	18
2.3.2	Zertifikatsprofil - Allgemeiner Teil	18
2.3.3	Zertifikatsprofil - Zertifikatserweiterungen	19
2.3.4	Struktur Beispielzertifikat	22
2.3.5	Zertifikatsgenerierung	22
2.4	eCR-Servicezertifikate	23
2.4.1	Allgemeine Anforderungen	23
2.4.2	Zertifikatsprofil - Allgemeiner Teil	23
2.4.3	Zertifikatsprofil - Zertifikatserweiterungen	24
2.4.4	Struktur Beispielzertifikat	25
2.4.5	Zertifikatsgenerierung	26
2.5	Sperrlisten	27
2.5.1	Allgemeine Anforderungen	27
2.5.2	Sperrlistenprofil - Allgemeiner Teil	27
2.5.3	Sperrlistenprofil - Sperrlistenerweiterungen	28
2.5.4	Struktur Beispielsperrliste	28
2.5.5	Sperrlistengenerierung	28
3	Anforderungen an die PKI	30
3.1	Allgemeine Anforderung an die Sicherheit einer PKI	30
3.1.1	Offlinebetrieb	30
3.1.2	Infrastruktursicherheit	31

3.1.3	Schlüsselbackup/-wiederherstellung	31
3.1.4	Vier-Augen-Prinzip	31
3.2	Bereitstellen einer Ausstellererklärung (Certificate Practice Statement)	31
3.3	Bereitstellung einer Zertifizierungsrichtlinie (Certificate Policy)	32
3.4	Anforderungen an Zertifikatnehmer	32
3.5	Umgang mit Schlüsseln	32
3.6	Zertifizierungsvorgang	33
3.7	Bereitstellung von Zertifikaten	33
3.8	Sperrungen von Zertifikaten	33
3.9	Bereitstellen von Sperr-/Statusinformationen	34
3.10	Gültigkeitszeiträume	34
4	PKI-Minimallösung	35
4.1	Ausprägung	35
4.2	Umsetzung von Standardfunktionalitäten	36
4.2.1	Anforderungen an die Sicherheit	36
4.2.2	Bereitstellung einer Ausstellererklärung/Zertifizierungsrichtlinie	36
4.2.3	Umgang mit Schlüsseln	37
4.2.4	Zertifizierungsvorgang	37
4.3	Vorteile	38
4.4	Nachteile	38
5	PKI-Optimallösung	39
5.1	Ausprägung	39
5.2	Umsetzung von Standardfunktionalität	39
5.2.1	Anforderung an die Sicherheit	39
5.2.2	Bereitstellung einer Ausstellererklärung/Zertifizierungsrichtlinie	40
5.2.3	Umgang mit Schlüsseln	40
5.2.4	Zertifizierungsvorgang	41
5.3	Vorteile	42
5.4	Nachteile	43
A	Struktur der Beispielzertifikate	44
B	Aufbau der Konfigurationsdatei (openssl.cnf)	49
C	Abkürzungsverzeichnis	54
D	Literatur	55

1 Einleitung

Um eine sichere Kommunikation zwischen Nutzern und Diensten sowie zwischen Diensten verschiedener Provider sicherzustellen, werden bei der Umsetzung elektronischer Fallakten X.509 Zertifikate zur Authentisierung und Nachrichtensicherung eingesetzt. In diesem Dokument werden der Aufbau der verwendeten Zertifikate sowie deren Verwaltung über eine PKI beschrieben.

1.1 Zielgruppe des Dokuments

Dieses Dokument richtet sich vorrangig an die Systemadministratoren der eCR-Provider, die für die Erstellung, Auslieferung und Pflege der X.509-Zertifikate sowie den Betrieb der zugehörigen PKI zuständig sind.

1.2 Genutzte Standards

Die in diesem Dokument entworfenen Zertifikatsprofile richten sich streng nach den Vorgaben des standardisierten Kodierungsschemata für X.509v3-Zertifikate [*RFC 3280*] und der ISIS-MTT-Spezifikation für interoperable PKI-Anwendungen [*ISIS MTT 1.1*]. Ziel ist die Erstellung X.509v3- und ISIS-MTT-konformer Zertifikate zur Authentisierung von Nutzern und Diensten der elektronischen Fallakte (electronic Case Records – eCR).

Die konkrete Ausgestaltung der SSL-Client- und SSL-Serverzertifikate orientiert sich zusätzlich an der vom BSI entwickelten PKI-1-Verwaltung [*BSI PKI-1*]. Die Inhalte wurden entsprechend den Anforderungen und Bedürfnissen der eCR-Spezifikation angepasst.

1.3 Konventionen

Die Syntax der im Text verwendeten Befehle, Attribute und Parameter bezieht sich im Wesentlichen auf die Eigenschaften von OpenSSL [*openssl*]. Daher ist beim Einsatz anderer Software darauf zu achten, dass die Konfiguration unter Umständen nicht in der beschriebenen Form übernommen werden kann. Allgemein beschriebene Anforderungen, die sich aus den entwickelten Zertifikatsprofilen ableiten, sollten jedoch auch in anderen Produkten umsetzbar sein.



Die Wörter "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", "MAY" in den Profilen sind entsprechend [RFC 2119] zu interpretieren.

2 Zertifikatsprofile

Um eine möglichst breite Produkt- und Herstellerunabhängigkeit zu gewährleisten, ist es erforderlich, sogenannte Zertifikatsprofile zu erstellen. Sie definieren vergleichbar einer Vorlage die genaue Ausgestaltung eines bestimmten Zertifikattyps mit entsprechend zu verwendenden Zertifikatserweiterungen (Attributen) und ihnen zugeordneten Werten.

2.1 CA-Zertifikat

Das CA-Zertifikat ist Ausgangspunkt für die Erstellung von End-Entity-Zertifikaten, die später für eine Vielzahl von Zwecken zum Einsatz kommen können (z. B. beidseitige SSL-Authentisierung). Das folgende Profil beschreibt sowohl die Anforderungen an Root-CA-Zertifikate als auch an Sub-CA-Zertifikate.

2.1.1 Allgemeine Anforderungen

CA-Zertifikate müssen ISIS-MTT konform sein. **[MUST]**

Die zu verwendende Schlüssellänge für CA-Zertifikate sollte mindestens 4096 Bit betragen. **[SHOULD]** Verbindliche Mindestlängen sind [eCR_CC-1.2] zu entnehmen. **[MUST]**

2.1.2 Zertifikatsprofil - Allgemeiner Teil

Version

Es müssen v3-Zertifikate verwendet werden. **[MUST]**

Signaturalgorithmus

Als Signaturalgorithmus muss SHA1, in Zukunft einer der Nachfolger aus der SHA2-Familie zum Einsatz kommen. Verbindliche Angaben sind [eCR-CC-1.2] zu entnehmen. **[MUST]**

Seriennummer

Bei der Seriennummer muss es sich um einen eindeutigen positiven Integerwert von maximal 20 Byte Länge handeln. **[MUST]**

Gültigkeit von/bis

CA-Zertifikate sollten eine Gültigkeit von max. 4 Jahren besitzen. **[SHOULD]**

IssuerUniqueID

Das Feld "IssuerUniqueID" darf nicht verwendet werden. **[MUST NOT]**

SubjectUniqueID

Das Feld "SubjectUniqueID" darf nicht verwendet werden. **[MUST NOT]**

Subject

Für den DName (Distinguished Name) müssen mindestens die Bestandteile C (Country), O (Organisation) und CN (Common Name) verwendet werden. **[MUST]**

Der Bestandteil OU (Organizational Unit) kann verwendet werden. **[MAY]**

Folgende Stringlängenbeschränkungen gelten **[MUST]**:

C (Country) → 2 Byte (ISO 3166 code)

O (Organisation) → max. 64 Byte

CN (Common Name) → max. 64 Byte

OU (Organizational Unit) → max. 64

Die Verwendung von weiteren DName-Attributen (z.B. EE (E-Mail)) ist zu vermeiden. **[SHOULD NOT]**
Sollten diese Attribute dennoch verwendet werden, gelten die Stringlängenbeschränkungen nach ISIS-MTT. **[MUST]**

Handelt es sich um die "oberste" CA, sollte der Namensbestandteil CN (Common Name) nach Möglichkeit den Bestandteil "Root CA" oder "PCA" (Policy CA) beinhalten. **[SHOULD]**

Der DName-Strings muss UTF8 kodiert werden. **[MUST]**

Die Verwendung eines Subsets (Unicode Latin-1 page - ANSI/ISO 8859-1) wird empfohlen. **[SHOULD]**

C=DE, O=[Name der Organisation], CN=[Name der CA]

Issuer

Der DName muss mit dem Subject-DName des Issuer-Zertifikates identisch sein. **[MUST]**

2.1.3 Zertifikatsprofil - Zertifikatserweiterungen

Der folgende Abschnitt stellt Zertifikatserweiterungen nach X.509v3 dar, die innerhalb der vorliegenden Spezifikation berücksichtigt werden müssen. Die Ausgestaltung orientiert sich streng an den Vorgaben von ISIS-MTT.

Zusätzlich zu den hier dargestellten dürfen zwar weitere Erweiterungen aufgenommen werden, diese müssen dann jedoch zwingend den ISIS-MTT-Vorgaben entsprechen. Es empfiehlt sich die Auswahl auf die hier dargestellten Erweiterungen zu beschränken.

SubjectKeyIdentifier (non-critical)

"SubjectKeyIdentifier" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**

Der SHA-1-Hashwert über den "subjectPublicKey" sollte verwendet werden. **[SHOULD]**

```
[ subjectKeyIdentifier = hash ]
```

AuthorityKeyIdentifier (non-critical)

"AuthorityKeyIdentifier" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**

Handelt es sich um ein selbstsigniertes CA-Zertifikat so gilt: **[MUST]**

```
authorityKeyIdentifier.keyIdentifier = subjectkeyIdentifier
```

Handelt es sich um ein von einer übergeordneten CA signiertes Zertifikat, so ist deren "subjectKeyIdentifier" anzugeben. **[MUST]**

KeyUsage (critical)

"KeyUsage" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**

Die Erweiterung ist stets als kritisch zu kennzeichnen. **[MUST]**

Für CA-Zertifikate muss die Nutzungsart „keyCertSign“ angegeben werden. **[MUST]**

Wird der zum Zertifikat gehörende Schlüssel zusätzlich für die Signatur von Sperrlisten genutzt, kann zusätzlich "crlSign" angegeben werden. **[MAY]**

Weitere Nutzungsarten dürfen nicht angegeben werden. **[MUST NOT]**

```
[ keyUsage = critical, keyCertSign, crlSign ]
```

SubjectAltNames (non-critical)

"SubjectAltNames" kann als Erweiterung im Zertifikat vorhanden sein. **[MAY]**

Bei Verwendung der Erweiterung wird empfohlen eine entsprechende LDAP-URL zu hinterlegen, von der das Zertifikat abgerufen werden kann. **[SHOULD]**

HTTP- und FTP-URLs, die auf das Zertifikat verweisen, können ebenfalls hinterlegt werden. **[MAY]**

E-Mail-Adressen (RFC822-Name) können ebenfalls hinterlegt werden. **[MAY]**

```
[ subjectAltName = URI:ldap://xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx ]
```

IssuerAltNames (non-critical)

"IssuerAltNames" kann als Erweiterung im Zertifikat vorhanden sein. **[MAY]**

Bei Verwendung der Erweiterung wird empfohlen eine entsprechende LDAP-URL zu hinterlegen, von der das Issuer-Zertifikat abgerufen werden kann. **[SHOULD]**

HTTP- und FTP-URLs, die auf das Zertifikat verweisen, können ebenfalls hinterlegt werden. **[MAY]**

```
[ issuerAltName = URI:ldap://xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx ]
```

BasicConstraints (critical)

"BasicConstraints" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**
Die Erweiterung ist stets als kritisch zu kennzeichnen. **[MUST]**

Die Erweiterung muss für „ca“ den Wert TRUE annehmen. **[MUST]**

"PathLen"-Constraint" gibt an wie viele CA-Zertifikate im Pfad unterhalb dieser CA existieren dürfen. Stellt die CA nur End-Entity-Zertifikate aus, so gilt "pathLen=0". Ein entsprechender Wert sollte angegeben werden. **[SHOULD]**

```
[ basicConstraints = critical, CA:TRUE, pathLen=X ]
```

CRLDistributionPoints (non-critical)

"CRLDistributionPoints" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**

In dieser Erweiterung ist die LDAP-Adresse anzugeben, unter der die vollständige Sperrliste der ausstellenden Zertifizierungsinstantz zu beziehen ist. **[MUST]**

Optional dürfen zusätzlich URLs (auch HTTP und FTP) angegeben werden, unter denen die Sperrliste alternativ bezogen werden kann. Weitere Informationen dürfen nicht in der Erweiterung enthalten sein. **[MAY]**

Stellt die CA gleichzeitig auch die CRL aus (direkte CRL) so darf das cRLIssuer-Feld nicht vorhanden sein. **[MUST NOT]**

```
[ crlDistributionPoints = URI:ldap://xxxxxxxxxxxxxxxxxxxx ]
```

CertificatePolicies (non-critical)

"CertificatePolicies" sollte als Erweiterung im Zertifikat vorhanden sein. **[SHOULD]**

Policyinformation sollte ausschließlich eine OID beinhalten. **[SHOULD]**

```
[ certificatePolicies = x.x.x.x.x.x ]
```

Authority Info Access (non-critical)

"AuthorityInfoAccess" sollte als Erweiterung im Zertifikat vorhanden sein. **[SHOULD]**

Wenn die ausstellende CA einen OCSP-Dienst anbietet, so muss dessen HTTP-URI in der Erweiterung enthalten sein. **[MUST]**

```
[ authorityInfoAccess = OCSP;URI:http://pki.klinik-xyz.de/ocsp ]
```

2.1.4 Struktur Beispielzertifikat

Die Struktur des Beispielzertifikates kann dem Anhang A.1 entnommen werden.

2.1.5 Zertifikatsgenerierung

CA-Keys und CA-Zertifikat generieren

```
openssl req -config /pfad/zu/openssl.cnf
            -new
            -x509
            -extensions v3_ca
            -sha1
            -newkey rsa:4096
            -keyout ca_key.pem
            -out ca_cert.pem
            -days 1095
```

Weitere wesentliche Parameter (z.B. Festlegung der Zertifikatserweiterungen) werden durch die Datei openssl.cnf vorgegeben. Eine entsprechende Anpassung der Konfiguration an die Bedürfnisse des Unternehmens ist in jedem Fall notwendig. Eine beispielhafte Konfiguration kann dem Anhang B entnommen werden.



2.2 Clientauthentisierungszertifikate

Clientauthentisierungszertifikate werden, wie ihr Name bereits andeutet, auf Clientsystemen (meist APCs) installiert und verwendet. Man benötigt sie für den Aufbau einer Verbindung (z.B. SSL/TLS) sobald seitens der Serversoftware eine Clientauthentisierung verlangt wird. Mit diesem Zertifikat authentisiert sich der Client gegenüber dem Server/Service.

Im vorliegenden Szenario dienen die ausgestellten Clientzertifikate ausschließlich diesem einen Zweck. Eine Beschränkung der Verwendung erfolgt über die Zertifikatserweiterungen und den ihnen zugeordneten Werten.

Der folgende Abschnitt definiert das Profil, also den Aufbau eines solchen Zertifikats. Sämtliche von einer CA auszustellenden Clientzertifikate haben sich an diesen Vorgaben zu orientieren.

2.2.1 Allgemeine Anforderungen

Clientzertifikate müssen ISIS-MTT konform sein. **[MUST]**

Die zu verwendende Schlüssellänge für Client-Zertifikate sollte mindestens 2048 Bit betragen. **[SHOULD]** Verbindliche Mindestlängen sind [eCR_CC-1.2] zu entnehmen. **[MUST]**

2.2.2 Zertifikatsprofil - Allgemeiner Teil

Version

Es müssen v3-Zertifikate verwendet werden. **[MUST]**

Signaturalgorithmus

Als Signaturalgorithmus muss SHA1, in Zukunft einer der Nachfolger aus der SHA2-Familie zum Einsatz kommen. Verbindliche Angaben sind [eCR-CC-1.2] zu entnehmen. **[MUST]**

Seriennummer

Bei der Seriennummer muss es sich um einen eindeutigen positiven Intergerwert von maximal. 20 Byte Länge handeln. **[MUST]**

Gültigkeit von/bis

SSL-Clientzertifikate sollten eine Gültigkeit von max. 2 Jahren besitzen. **[SHOULD]**

IssuerUniqueID

Das Feld "IssuerUniqueID" darf nicht verwendet werden. **[MUST NOT]**

SubjectUniqueID

Das Feld "SubjectUniqueID" darf nicht verwendet werden. **[MUST NOT]**

Subject

Der Subject-DName muss über die gesamte Lebenszeit der CA hinweg eindeutig sein. **[MUST]**

Für den DName müssen mindestens die Bestandteile C (Country), O (Organization) und CN (Common Name) verwendet werden. **[MUST]**

Ferner sollten die Bestandteile T (Titel), G (Given Name) und SN (Surname) enthalten sein und die

entsprechenden Informationen zum Zertifikatsinhabers (Verantwortlicher) beinhalten. **[SHOULD]**

Die Bestandteile OU (Organizational Unit) und S (Serialnumber) können verwendet werden. **[MAY]**

Folgende Stringlängenbeschränkungen gelten **[MUST]**:

C (Country) → 2 Byte (ISO 3166 code)

O (Organization) → max. 64 Byte

CN (Common Name) → max. 64 Byte

T (Titel) → max. 64 Byte

G (Given Name) → max. 64. Byte

SN (Surname) → max. 64 Byte

OU (Organizational Unit) → max. 64 Byte

S (Serialnumber) → max. 64 Byte

Die Verwendung von weiteren DName-Attributen (z.B. E (E-Mail)) ist zu vermeiden. **[SHOULD NOT]**

Sollten diese Attribute dennoch verwendet werden, gelten die Stringlängenbeschränkungen nach ISIS-MTT. **[MUST]**

Der DName-String muss UTF8 kodiert werden. **[MUST]**

Die Verwendung eines Subsets (Unicode Latin-1 page - ANSI/ISO 8859-1) wird empfohlen. **[SHOULD]**

Clientzertifikate dienen der Authentisierung von Organisationen oder natürlichen Personen. Dies sollte im "Subject" bei der Festlegung des DName (Distinguished Name) entsprechend berücksichtigt werden. Für organisationsbezogene Zertifikate sollte es immer einen im Zertifikat hinterlegten Ansprechpartner geben. **[SHOULD]**

Werden im CN Pseudonyme verwendet, so muss dies mit dem Suffix ":PN" gekennzeichnet werden.

[MUST]

C=DE, O=[Name der Issuer Organisation], CN= [Name der Institution], T=[Titel Ansprechpartner],

G=[Vorname(n) Ansprechpartner], SN=[Nachname(n) Ansprechpartner]

Issuer

Der DName muss mit den subject-DName des Issuer-Zertifikates identisch sein. **[MUST]**



2.2.3 Zertifikatsprofil - Zertifikatserweiterungen

Der folgende Abschnitt stellt Zertifikatserweiterungen nach X.509v3 dar, die innerhalb der vorliegenden Spezifikation berücksichtigt werden müssen. Die Ausgestaltung orientiert sich streng an den Vorgaben von ISIS-MTT.

Zusätzlich zu den hier dargestellten dürfen zwar weitere Erweiterungen aufgenommen werden, diese müssen dann jedoch zwingend den ISIS-MTT-Vorgaben entsprechen. Es empfiehlt sich die Auswahl auf die hier dargestellten Erweiterungen zu beschränken.

AuthorityKeyIdentifier (non-critical)

"Authority KeyIdentifier" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**

Der "SubjectKeyIdentifier" der signierenden CA muss verwendet werden. **[MUST]**

```
[ authorityKeyIdentifier = keyid(, issuer:always) ]
```

SubjectKeyIdentifier (non-critical)

"SubjectKeyIdentifier" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**

Der SHA-1-Hashwert über den "subjectPublicKey" sollte verwendet werden. **[SHOULD]**

```
[ subjectKeyIdentifier = hash ]
```

KeyUsage (critical)

"KeyUsage" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**

Die Erweiterung ist stets als kritisch zu kennzeichnen. **[MUST]**

Für SSL-Clientzertifikate darf ausschließlich die Nutzungsart „digitalSignature“ angegeben werden. **[MUST]**

```
[ keyUsage = critical, digitalSignature ]
```

IssuerAltNames (non-critical)

"IssuerAltNames" kann als Erweiterung im Zertifikat vorhanden sein. **[MAY]**

Bei Verwendung der Erweiterung wird empfohlen eine entsprechende LDAP-URL zu hinterlegen, von der das Issuer-Zertifikat abgerufen werden kann. **[SHOULD]**

HTTP- und FTP-URLs, die auf das Zertifikat verweisen, können ebenfalls hinterlegt werden. **[MAY]**

```
[ issuerAltName = URI:ldap://xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx ]
```

SubjectAltNames (non-critical)

"SubjectAltNames" kann als Erweiterung im Zertifikat vorhanden sein. **[MAY]**

Bei Verwendung der Erweiterung wird empfohlen eine entsprechende LDAP-URL zu hinterlegen, von der das Zertifikat abgerufen werden kann. **[SHOULD]**

HTTP- und FTP-URLs, die auf das Zertifikat verweisen, können ebenfalls hinterlegt werden. **[MAY]**
E-Mail-Adressen (RFC822-Name) können ebenfalls hinterlegt werden. **[MAY]**

```
[ subjectAltName = URI:ldap://xxxxxxxxxxxxxxxxxxx ]
```

BasicConstraints (critical)

"BasicConstraints" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**
Die Erweiterung ist stets als kritisch zu kennzeichnen. **[MUST]**

Die Erweiterung muss für "ca" den Wert FALSE annehmen. **[MUST]**

```
[ basicConstraints = critical, CA:FALSE ]
```

ExtendedKeyUsage (non critical)

"ExtendedKeyUsage" sollte als Zertifikatserweiterung im Zertifikat vorhanden sein. **[SHOULD]**

Falls diese Erweiterung im Zertifikat enthalten ist, darf sie ausschließlich den Wert „ClientAuth“ (OID 1.3.6.1.5.5.7.3.2) annehmen. **[MUST]**

```
[ extendedKeyUsage = 1.3.6.1.5.5.7.3.2 ]
```

CRLDistributionPoints (non-critical)

"CRLDistributionPoints" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**

In dieser Erweiterung ist die LDAP-Adresse anzugeben, unter der die vollständige Sperrliste der ausstellenden Zertifizierungsinstanz zu beziehen ist. **[MUST]**

Optional dürfen zusätzlich URLs (auch HTTP und FTP) angegeben werden, unter denen die Sperrliste alternativ bezogen werden kann. Weitere Informationen dürfen nicht in der Erweiterung enthalten sein. **[MAY]**

```
[ crlDistributionPoints = URI:ldap://xxxxxxxxxxxxxxxxxxx ]
```

CertificatePolicies (non-critical)

"CertificatePolicies" sollte als Erweiterung im Zertifikat vorhanden sein. **[SHOULD]**

Policyinformation sollte ausschließlich eine OID beinhalten. **[SHOULD]**

```
[ certificatePolicies = x.x.x.x.x.x.x ]
```

Authority Info Access (non-critical)

"AuthorityInfoAccess" sollte als Erweiterung im Zertifikat vorhanden sein. **[SHOULD]**

Wenn die ausstellende CA einen OCSP-Dienst anbietet, so muss dessen HTTP-URI in der Erweiterung enthalten sein. **[MUST]**

```
[ authorityInfoAccess = OCSP;URI:http://pki.klinik-xyz.de/ocsp ]
```



2.2.4 Struktur Beispielzertifikat

Die Struktur des Beispielzertifikates kann dem Anhang A.1 entnommen werden.

2.2.5 Zertifikatsgenerierung

Keys und Request generieren

```
openssl req -config /path/to/openssl.cnf
            -new
            -newkey rsa:2048
            -sha1
            -keyout newkey.pem
            -out newreq.pem
            -days 730
            -nameopt utf8
```

Zertifikat ausstellen

```
openssl ca -config /path/to/openssl.cnf
           -policy policy_client
           -days 730
           -extension client_auth_ext
           -out newcert.pem
           -infile newreq.pem
```

Personal Security Environment (PKCS#12) exportieren

```
openssl pkcs12 -export
              -in /path/to/newcert.pem
              -inkey /path/to/newkey.pem
              -certfile /path/to/ca_cert.pem
              -out /path/to/newpse.p12
```

Weitere wesentliche Parameter (z.B. Festlegung der Zertifikatserweiterungen) werden durch die Datei `openssl.cnf` vorgegeben. Eine entsprechende Anpassung der Konfiguration an die Bedürfnisse des Unternehmens ist in jedem Fall notwendig. Eine beispielhafte Konfiguration kann dem Anhang B entnommen werden.

2.3 SSL-Serverzertifikate

Im Gegensatz zu Clientzertifikaten finden Serverzertifikate auf Systemen Einsatz, die den Clients einen entsprechenden Service (z.B. https) anbieten. Mit dem Serverzertifikat authentisiert sich der Server gegenüber dem Client. Erst bei erfolgreicher Authentisierung kann der Aufbau einer SSL-Verbindung erfolgen.

Genau wie für die Clientzertifikate gelten auch für die Serverzertifikate bestimmte über die Zertifikatserweiterungen definierte Begrenzungen, die den Zertifikatszweck ausschließlich auf den Aufbau einer SSL-Verbindung beschränken soll.

Der folgende Abschnitt definiert das Profil, also den Aufbau eines solchen Zertifikats. Sämtliche von der CA auszustellende Serverzertifikate haben sich an diesen Vorgaben zu orientieren.

2.3.1 Allgemeine Anforderungen

SSL-Serverzertifikate müssen ISIS-MTT konform sein. **[MUST]**

Die zu verwendende Schlüssellänge für Server-Zertifikate sollte mindestens 2048 Bit betragen. **[SHOULD]** Verbindliche Mindestlängen sind [eCR_CC-1.2] zu entnehmen. **[MUST]**

2.3.2 Zertifikatsprofil - Allgemeiner Teil

Version

Es müssen v3-Zertifikate verwendet werden. **[MUST]**

Signaturalgorithmus

Als Signaturalgorithmus muss SHA1, in Zukunft einer der Nachfolger aus der SHA2-Familie zum Einsatz kommen. Verbindliche Angaben sind [eCR-CC-1.2] zu entnehmen. **[MUST]**

Seriennummer

Bei der Seriennummer muss es sich um einen eindeutigen positiven Intergerwert von maximal. 20 Byte Länge handeln. **[MUST]**

Gültigkeit von/bis

SSL-Serverzertifikate sollten eine Gültigkeit von max. 1 Jahr besitzen. **[SHOULD]**



IssuerUniqueID

Das Feld "IssuerUniqueID" darf nicht verwendet werden. **[MUST NOT]**

SubjectUniqueID

Das Feld "SubjectUniqueID" darf nicht verwendet werden. **[MUST NOT]**

Subject

Der Subject-DName muss über die gesamte Lebenszeit der CA hinweg eindeutig sein. **[MUST]**

Für den DName müssen mindestens die Bestandteile C (Country), O (Organization) und CN (Common Name) verwendet werden. **[MUST]**

Ferner sollten die Bestandteile T (Titel), G (Given Name) und SN (Surname) enthalten sein und entsprechende Informationen zum Zertifikatsverantwortlichen enthalten. **[SHOULD]**

Der Bestandteil OU (Organizational Unit) kann verwendet werden. **[MAY]**

Folgende Stringlängenbeschränkungen gelten **[MUST]**:

C (Country) → 2 Byte (ISO 3166 code)

O (Organization) → max. 64 Byte

CN (Common Name) → max. 64 Byte

T (Titel) → max. 64 Byte

G (Given Name) → max. 64 Byte

SN (Surname) → max. 64 Byte

OU (Organizational Unit) → max. 64

Die Verwendung von weiteren DName-Attributen (z.B. E (E-Mail)) ist zu vermeiden. **[SHOULD NOT]**

Sollten diese Attribute dennoch verwendet werden, gelten die Stringlängenbeschränkungen nach ISIS-MTT. **[MUST]**

Der DName-String muss UTF8 kodiert werden. **[MUST]**

Die Verwendung eines Subsets (Unicode Latin-1 page - ANSI/ISO 8859-1) wird empfohlen. **[SHOULD]**

Serverzertifikate dienen der Authentisierung von Servern/Diensten. Dies muss im Subject bei der Festlegung des „Distinguished Name“ entsprechend berücksichtigt werden. Für Serverzertifikate sollte es immer einen im Zertifikat hinterlegten Ansprechpartner geben. **[SHOULD]**

Der CN (Common Name) muss den DNS-Servernamen beinhalten, da der Client per default das Serverzertifikat sonst als nicht vertrauenswürdig einstuft. **[MUST]**

C=DE, O=[Name der Organisation](, OU=[Bereich]), CN= [DNS-Servername], T=[Titel], G=[Vorname(n) Ansprechpartner], SN=[Nachname(n) Ansprechpartner]

Issuer

Der DName muss mit den subject-DName des Issuer-Zertifikates identisch sein. **[MUST]**

2.3.3 Zertifikatsprofil - Zertifikatserweiterungen

Der folgende Abschnitt stellt Zertifikatserweiterungen nach X.509v3 dar, die innerhalb der vorliegenden Spezifikation berücksichtigt werden müssen. Die Ausgestaltung orientiert sich streng an den Vorgaben von ISIS-MTT.

Zusätzlich zu den hier dargestellten dürfen zwar weitere Erweiterungen aufgenommen werden, diese müssen dann jedoch zwingend den ISIS-MTT-Vorgaben entsprechen. Es empfiehlt sich die Auswahl auf die hier dargestellten Erweiterungen zu beschränken.

AuthorityKeyIdentifier (non-critical)

"AuthorityKeyIdentifier" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**

Der "SubjectKeyIdentifier" der signierenden CA muss verwendet werden. **[MUST]**

```
[ authorityKeyIdentifier = keyid(, issuer:always) ]
```

SubjectKeyIdentifier (non-critical)

"SubjectKeyIdentifier" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**

Der SHA-1-Hashwert über den "subjectPublicKey" sollte verwendet werden. **[SHOULD]**

```
[ subjectKeyIdentifier = hash ]
```

KeyUsage (critical)

"KeyUsage" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**

Die Erweiterung ist stets als kritisch zu kennzeichnen. **[MUST]**

Für SSL-Serverzertifikate darf ausschließlich die Nutzungsart „keyEncipherment“ angegeben werden. **[MUST]**

```
[ keyUsage = critical, keyEncipherment ]
```

IssuerAltNames (non-critical)

"IssuerAltNames" kann als Erweiterung im Zertifikat vorhanden sein. **[MAY]**

Bei Verwendung der Erweiterung wird empfohlen eine entsprechende LDAP-URL zu hinterlegen, von der das Issuer-Zertifikat abgerufen werden kann. **[SHOULD]**

HTTP- und FTP-URLs, die auf das Zertifikat verweisen können, ebenfalls hinterlegt werden. **[MAY]**

```
[ issuerAltName = URI:ldap://xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx ]
```

SubjectAltNames (non-critical)

"SubjectAltNames" kann als Erweiterung im Zertifikat vorhanden sein. **[MAY]**

Bei Verwendung der Erweiterung wird empfohlen eine entsprechende LDAP-URL zu hinterlegen, von der das Zertifikat abgerufen werden kann. **[SHOULD]**

HTTP- und FTP-URLs, die auf das Zertifikat verweisen, können ebenfalls hinterlegt werden. **[MAY]**

E-Mail-Adressen (RFC822-Name) können ebenfalls hinterlegt werden. **[MAY]**

```
[ subjectAltName = URI:ldap://xxxxxxxxxxxxxxxxxxxxxxxx ]
```



BasicConstraints (critical)

"BasicConstraints" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**

Die Erweiterung ist stets als kritisch zu kennzeichnen. **[MUST]**

Die Erweiterung muss für "ca" den Wert FALSE annehmen. **[MUST]**

```
[ basicConstraints = critical, CA:FALSE ]
```

ExtendedKeyUsage (non critical)

"ExtendedKeyUsage" sollte als Zertifikatserweiterung im Zertifikat vorhanden sein. **[SHOULD]**

*Falls diese Erweiterung im Zertifikat enthalten ist, darf sie ausschließlich den Wert „ServerAuth“ (OID 1.3.6.1.5.5.7.3.1) annehmen. **[MUST]***

```
[ extendedKeyUsage = 1.3.6.1.5.5.7.3.1 ]
```

CRLDistributionPoints (non-critical)

"CRLDistributionPoints" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**

In dieser Erweiterung ist die LDAP-Adresse anzugeben, unter der die vollständige Sperrliste der ausstellenden Zertifizierungsinstantz zu beziehen ist. **[MUST]**

Optional dürfen zusätzlich URLs (auch HTTP und FTP) angegeben werden, unter denen die Sperrliste alternativ bezogen werden kann. Weitere Informationen dürfen nicht in der Erweiterung enthalten sein. **[MAY]**

```
[ crlDistributionPoints = URI:ldap://xxxxxxxxxxxxxxxxxxxx ]
```

CertificatePolicies (non-critical)

"CertificatePolicies" sollte als Erweiterung im Zertifikat vorhanden sein. **[SHOULD]**

Policyinformation sollte ausschließlich eine OID beinhalten. **[SHOULD]**

```
[ certificatePolicies = x.x.x.x.x.x ]
```

Authority Info Access (non-critical)

"AuthorityInfoAccess" sollte als Erweiterung im Zertifikat vorhanden sein. **[SHOULD]**

Wenn die ausstellende CA einen OCSP-Dienst anbietet, so muss dessen HTTP-URI in der Erweiterung enthalten sein. **[MUST]**

```
[ authorityInfoAccess = OCSP;URI:http://pki.klinik-xyz.de/ocsp ]
```

2.3.4 Struktur Beispielzertifikat

Die Struktur des Beispielzertifikates kann dem Anhang A.1 entnommen werden.

2.3.5 Zertifikatsgenerierung

Keys und Request generieren

```
openssl req -config /path/to/openssl.cnf
            -new
            -newkey rsa:2048
            -sha1
            -keyout newkey.pem
            -out newreq.pem
            -days 365
            -nameopt utf8
```

Zertifikat ausstellen

```
openssl ca -config /path/to/openssl.cnf
           -policy policy_service
           -days 365
           -extensions ssl_server_ext
           -out newcert.pem
           -infiles newreq.pem
```

Personal Security Environment (PKCS#12) exportieren

```
openssl pkcs12 -export
              -in /path/to/newcert.pem
              -inkey /path/to/newkey.pem
              -certfile /path/to/ca_cert.pem
              -out /path/to/newpse.p12
```

Weitere wesentliche Parameter (z.B. Festlegung der Zertifikatserweiterungen) werden durch die Datei openssl.cnf vorgegeben. Eine entsprechende Anpassung der Konfiguration an die Bedürfnisse des Unternehmens ist in jedem Fall notwendig. Eine beispielhafte Konfiguration kann dem Anhang B entnommen werden.



2.4 eCR-Servicezertifikate

eCR-Servicezertifikate und die dazugehörigen Schlüssel dienen dem Signieren von Nachrichten und dem Verschlüsseln von symmetrischen Kommunikationsschlüsseln. Über diese Mechanismen kann ein sicherer Kontext generiert und Vertrauen zwischen verschiedenen Partnern aufgebaut werden.

Der folgende Abschnitt definiert das Profil, also den Aufbau eines solchen Zertifikats. Sämtliche von der CA auszustellende eCR-Servicezertifikate haben sich an diesen Vorgaben zu orientieren.

2.4.1 Allgemeine Anforderungen

eCR-Servicezertifikate müssen ISIS-MTT konform sein. **[MUST]**

Die zu verwendende Schlüssellänge für eCR-Service-Zertifikate sollte mindestens 2048 Bit betragen. **[SHOULD]** Verbindliche Mindestlängen sind [eCR_CC-1.2] zu entnehmen. **[MUST]**

2.4.2 Zertifikatsprofil - Allgemeiner Teil

Version
Vgl. SSL-Serverzertifikat
Signaturalgorithmus
Vgl. SSL-Serverzertifikat
Seriennummer
Vgl. SSL-Serverzertifikat
Gültigkeit von/bis
Vgl. SSL-Serverzertifikat
IssuerUniqueID
Vgl. SSL-Serverzertifikat
SubjectUniqueID
Vgl. SSL-Serverzertifikat

Subject

Der Subject-DName muss über die gesamte Lebenszeit der CA hinweg eindeutig sein. **[MUST]**

Für den DName müssen mindestens die Bestandteile C (Country), O (Organization) und CN (Common Name) verwendet werden. **[MUST]**

Ferner sollten die Bestandteile T (Titel), G (Given Name) und SN (Surname) enthalten sein und entsprechende Informationen zum Zertifikatsverantwortlichen enthalten. **[SHOULD]**

Der Bestandteil OU (Organizational Unit) kann verwendet werden. **[MAY]**

Folgende Stringlängenbeschränkungen gelten **[MUST]**:

C (Country) → 2 Byte (ISO 3166 code)

O (Organization) → max. 64 Byte

CN (Common Name) → max. 64 Byte

T (Titel) → max. 64 Byte

G (Given Name) → max. 64. Byte

SN (Surname) → max. 64 Byte

OU (Organizational Unit) → max. 64

Die Verwendung von weiteren DName-Attributen (z.B. E (E-Mail)) ist zu vermeiden. **[SHOULD NOT]**

Sollten diese Attribute dennoch verwendet werden, gelten die Stringlängenbeschränkungen nach ISIS-MTT. **[MUST]**

Der DName-String muss UTF8 kodiert werden. **[MUST]**

Die Verwendung eines Subsets (Unicode Latin-1 page - ANSI/ISO 8859-1) wird empfohlen. **[SHOULD]**

Servicezertifikate dienen der Authentisierung von Diensten. Dies muss im Subject bei der Festlegung des „Distinguished Name“ entsprechend berücksichtigt werden. Für Serverzertifikate sollte es immer einen im Zertifikat hinterlegten Ansprechpartner geben. **[SHOULD]**

Der CN (Common Name) muss den Servicenamen beinhalten. **[MUST]**

C=DE, O=[Name der Organisation](, OU=[Bereich]), CN= [eCR-Servicename]

Issuer

Vgl. SSL-Serverzertifikat

2.4.3 Zertifikatsprofil - Zertifikatserweiterungen

Der folgende Abschnitt stellt Zertifikatserweiterungen nach X.509v3 dar, die innerhalb der vorliegenden Spezifikation berücksichtigt werden müssen. Die Ausgestaltung orientiert sich streng an den Vorgaben von ISIS-MTT.

Zusätzlich zu den hier dargestellten dürfen zwar weitere Erweiterungen aufgenommen werden, diese müssen dann jedoch zwingend den ISIS-MTT-Vorgaben entsprechen. Es empfiehlt sich die Auswahl auf die hier dargestellten Erweiterungen zu beschränken.



AuthorityKeyIdentifier (non-critical)

Vgl. SSL-Serverzertifikat

SubjectKeyIdentifier (non-critical)

Vgl. SSL-Serverzertifikat

KeyUsage (critical)

"KeyUsage" muss als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST]**
Die Erweiterung ist stets als kritisch zu kennzeichnen. **[MUST]**

Für eCR-Servicezertifikate dürfen ausschließlich die Nutzungsarten „keyEncipherment“ und "digitalSignature" angegeben werden. **[MUST]**

```
[ keyUsage = critical, keyEncipherment, digitalSignature ]
```

IssuerAltNames (non-critical)

Vgl. SSL-Serverzertifikat

SubjectAltNames (non-critical)

Vgl. SSL-Serverzertifikat

BasicConstraints (critical)

Vgl. SSL-Serverzertifikat

CRLDistributionPoints (non-critical)

Vgl. SSL-Serverzertifikat

CertificatePolicies (non-critical)

Vgl. SSL-Serverzertifikat

Authority Info Access (non-critical)

Vgl. SSL-Serverzertifikat

ExtendedKeyUsage

"ExtendedKeyUsage" darf nicht als Zertifikatserweiterung im Zertifikat vorhanden sein. **[MUST NOT]**

2.4.4 Struktur Beispielzertifikat

Die Struktur des Beispielzertifikates kann dem Anhang A.1 entnommen werden.

2.4.5 Zertifikatsgenerierung

Keys und Request generieren

```
openssl req -config /path/to/openssl.cnf
            -new
            -newkey rsa:2048
            -sha1
            -keyout newkey.pem
            -out newreq.pem
            -days 365
            -nameopt utf8
```

Zertifikat ausstellen

```
openssl ca -config /path/to/openssl.cnf
           -policy policy_service
           -days 365
           -extensions ecr_service_ext
           -out newcert.pem
           -infiles newreq.pem
```

Personal Security Environment (PKCS#12) exportieren

```
openssl pkcs12 -export
              -in /path/to/newcert.pem
              -inkey /path/to/newkey.pem
              -certfile /path/to/ca_cert.pem
              -out /path/to/newpse.p12
```

Weitere wesentliche Parameter (z.B. Festlegung der Zertifikatserweiterungen) werden durch die Datei openssl.cnf vorgegeben. Eine entsprechende Anpassung der Konfiguration an die Bedürfnisse des Unternehmens ist in jedem Fall notwendig. Eine beispielhafte Konfiguration kann dem Anhang B entnommen werden.



2.5 Sperrlisten

Anhand von Sperrlisten kann überprüft werden, ob ein Zertifikat für ungültig erklärt und zurückgezogen wurde. Sowohl Client als auch Server/Service sollten das von der Gegenseite zur Verfügung gestellte Zertifikat gegen die Sperrliste prüfen, bevor Daten zum Kommunikationspartner übertragen werden. Ist das Zertifikat der Gegenstelle ungültig, ist die Verbindung abzubrechen.

Der folgende Abschnitt definiert das Profil, also den Aufbau einer solchen Sperrliste. Sämtliche von der CA ausgestellten Sperrlisten haben sich an diesen Vorgaben zu orientieren.

Die getroffenen Vorgaben beziehen sich auf direkte CRLs.

2.5.1 Allgemeine Anforderungen

Sperrlisten müssen ISIS-MTT konform sein. **[MUST]**

2.5.2 Sperrlistenprofil - Allgemeiner Teil

Version

Es müssen v2-Sperrlisten verwendet werden. **[MUST]**

Signaturalgorithmus

Als Signaturalgorithmus muss SHA1, in Zukunft einer der Nachfolger aus der SHA2-Familie zum Einsatz kommen. Verbindliche Angaben sind [eCR-CC-1.2] zu entnehmen. **[MUST]**

Issuer

Der DName muss mit den subject-DName des Issuer-Zertifikates identisch sein. **[MUST]**

Gültigkeit von/bis (thisUpdate/nextUpdate)

Die Gültigkeitsdauer (von/bis) der Sperrliste muss angegeben werden. **[MUST]**

2.5.3 Sperrlistenprofil - Sperrlistenerweiterungen

Auch Sperrlisten können wie Zertifikate mit beliebigen Erweiterungen versehen werden. Der folgende Abschnitt beschreibt sinnvolle Sperrlistenerweiterungen.

Zusätzlich zu den hier dargestellten dürfen zwar weitere Erweiterungen aufgenommen werden, diese müssen dann jedoch zwingend den ISIS-MTT-Vorgaben entsprechen. Es empfiehlt sich die Auswahl auf die hier dargestellten Erweiterungen zu beschränken.

AuthorityKeyIdentifier

Der KeyIdentifier muss angegeben werden. **[MUST]**

```
[ authorityKeyIdentifier = keyid ]
```

CRLNumber

Die laufende Nummer für eine ausgestellte CRL muss angegeben werden. **[MUST]**

Es muss sich um einen eindeutigen positiven Integerwert von maximal 20 Byte Länge handeln. **[MUST]**

```
[ crlnumber = /path/to/crlnumber ]
```

IssuerAltNames

"IssuerAltNames" kann als Erweiterung in der Sperrliste vorhanden sein. **[MAY]**

Bei Verwendung der Erweiterung wird empfohlen eine entsprechende LDAP-URL zu hinterlegen, von der das Issuer-Zertifikat abgerufen werden kann. **[SHOULD]**

HTTP- und FTP-URLs, die auf das Zertifikat verweisen können ebenfalls hinterlegt werden. **[MAY]**

```
[ issuerAltName = URI:ldap://xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx ]
```

2.5.4 Struktur Beispielsperrliste

Die Struktur der Beispielsperrliste kann dem Anhang A.1 entnommen werden.

2.5.5 Sperrlistengenerierung

Zertifikat sperren

```
openssl ca -revoke /path/to/certificate.pem
           -config /path/to/openssl.cnf
```



Sperrliste generieren

```
openssl ca -genctrl  
           -md sha1  
           -config /path/to/openssl.cnf  
           -out /path/to/ctrl.pem
```

Weitere wesentliche Parameter (z.B. Festlegung der Sperrlistenerweiterungen) werden durch die Datei `openssl.cnf` vorgegeben. Eine entsprechende Anpassung der Konfiguration an die Bedürfnisse des Unternehmens ist in jedem Fall notwendig. Eine beispielhafte Konfiguration kann dem Anhang B entnommen werden.

3 Anforderungen an die PKI

Dieses und die folgenden Kapitel befassen sich primär mit der Spezifikation der Anforderung an die Organisation der Zertifikatserstellung und -verwaltung im Rahmen des Betriebs einer PKI. In diesem Zusammenhang wird davon ausgegangen, dass der Betrieb der PKI hausintern erfolgt und keine Zertifikate von autorisierten Zertifizierungsinstanzen (GeoTrust, Thawte, Verisign, etc.) bezogen werden.

Zwei Lösungen werden vorgestellt und verglichen, die in ihrer Struktur teilweise erhebliche Unterschiede aufweisen. Auf Grundlage der Quantität der auszustellenden Zertifikate muss entschieden werden, welche Lösung letztendlich in welcher Phase des Projektes zum Einsatz kommt.

3.1 Allgemeine Anforderung an die Sicherheit einer PKI

3.1.1 Offlinebetrieb

Ein erfolgreicher Angriff auf die PKI ermöglicht einem potentiellen Angreifer unter Umständen diese vollständig zu kompromittieren. Alle auf dieser PKI basierenden Anwendungen (z.B. beidseitige SSL-Authentisierung, Dienstauthentisierung) wären somit ebenfalls kompromittiert und würden bis zur Neuausstellung sämtlicher Zertifikate als unsicher gelten. Die CA stellt daher eine hochschutzbedürftige Anwendung dar.

Unter der Annahme, dass die meisten Anwendungen im Unternehmen maximal mittelschutzbedürftig sind, schlägt das BSI vor zu prüfen, ob es nicht sinnvoll ist, die wenigen hochschutzbedürftigen Anwendungen auf isolierte Systeme auszulagern. Dem hohen Schutzbedarf kann ansonsten nur durch unverhältnismäßig hohen (finanziellen) Aufwand begegnet werden.

Für die im Rahmen der elektronischen Fallakte aufzubauende PKI wird eine Offline-CA angestrebt. Ein Datenaustausch (z.B. Certificate Signing Request, Zertifikate, Personal Security Environment) mit der CA erfolgt daher ausschließlich über Wechselmedien. Die Anbindung an ein Netzwerk ist untersagt.



3.1.2 Infrastruktursicherheit

Das CA-System muss auch aus infrastruktureller Sicht vor dem Zugriff durch unberechtigte Personen geschützt werden. Es bietet sich an, dies über eine räumliche Abgrenzung zu erreichen. Je nach gewählter Lösung kann dafür bereits ein Safe ausreichen oder aber ein eigener Raum verwendet werden. Abschnitt 4.2.1 und 5.2.1 beleuchten diese Problematik etwas genauer.

3.1.3 Schlüsselbackup/-wiederherstellung

Unabhängig von der gewählten Lösung muss auch nach einem Havariefall gewährleistet sein, dass die CA in ihrer vollen Funktionalität wiederhergestellt werden kann. Der Schlüssel muss dazu auf einem Backupmedium in gesicherter Form (verschlüsselt) abgelegt sein. Es sollte darauf geachtet werden, dass das Backupmedium hinreichend lange lesbar ist und räumlich getrennt von der eigentlichen CA aufbewahrt wird. Hierfür bietet sich beispielsweise ein Safe in einem anderen Brandabschnitt an. Backup und Wiederherstellung der CA haben nach dem 4-Augen-Prinzip zu erfolgen (vgl. Abschnitt 3.1.4).

3.1.4 Vier-Augen-Prinzip

Der Zugriff auf den geheimen Schlüssel der CA und somit auf all ihre Funktionen darf nicht durch eine einzelne Person allein möglich sein. Alle Aufgaben müssen nach dem Vier-Augen-Prinzip realisiert werden.

Die Abschnitte 4.2.1 und 5.2.1 stellen je nach Lösungsansatz verschiedene Möglichkeiten der Realisierung dieses Prinzips dar.

Stellvertreterregelungen sollten mit in die konkrete Ausgestaltung einer entsprechenden Organisation einfließen.

3.2 Bereitstellen einer Ausstellererklärung (Certificate Practice Statement)

Die Ausstellererklärung beschreibt die Vorgehensweise beim Betrieb der CA. Insbesondere die Thematik der Organisation des Ausstellens von Zertifikaten wird hier sehr genau behandelt.

Die Ausgestaltung der Ausstellererklärung sollte sich an den Vorgaben von RFC2527 orientieren.

3.3 Bereitstellung einer Zertifizierungsrichtlinie (Certificate Policy)

Die Zertifizierungsrichtlinie ist eine Zusammenstellung von Regeln, die die Anwendbarkeit eines Zertifikates für bestimmte Nutzergruppen/Applikationen mit gemeinsamen Anforderungen hinsichtlich der Sicherheit festlegt.

Die Ausgestaltung der Zertifizierungsrichtlinie sollte sich an den Vorgaben von RFC2527 orientieren.

Wurde sich innerhalb des Unternehmens oder zwischen Unternehmen auf die Festlegung einer Zertifizierungsrichtlinie (Policy) und deren Befolgung verständigt, so sollte die OID des Dokuments zumindest über die in den Zertifikatsprofilen definierten Zertifikatserweiterungen (CertificatePolicies) bekannt gemacht werden.

3.4 Anforderungen an Zertifikatnehmer

Clientzertifikate sind Gruppen- oder Personenzertifikate. Sie werden für Organisationen (z.B. Arztpraxen) oder natürliche Personen (z.B. Mitarbeiter oder niedergelassene Ärzte) ausgestellt. Zertifikatnehmer ist jedoch in jedem Fall der im Zertifikat benannte Ansprechpartner bzw. die im Zertifikat benannte Person.

Server-/Servicezertifikate sind Gruppensertifikate. Sie werden ausschließlich für Dienste auf IT-Systemen ausgestellt. Es gibt immer einen zuständigen Ansprechpartner. In diesem Fall gilt dieser als Zertifikatnehmer.

Der Zertifikatnehmer muss der Ausstellererklärung sowie den geltenden Zertifizierungsrichtlinien und der Veröffentlichung des Zertifikats zustimmen. Er sorgt für einen sicheren Umgang mit dem zum Zertifikat gehörenden geheimen Schlüssel (vgl. Abschnitt 3.5) und veranlasst umgehend die Sperrung des Zertifikats (vgl. Abschnitt 3.8) sobald er eine missbräuchliche Nutzung nicht mehr ausschließen kann.

3.5 Umgang mit Schlüsseln

Der geheime Schlüssel der CA darf bis auf ein gültiges Backup die CA nicht verlassen. Zusätzlich dazu muss er immer in verschlüsselter Form vorliegen. Mindestens zwei Personen werden zu seiner Freischaltung benötigt. (Vier-Augen-Prinzip)

Sämtliche zu Zertifikatnehmern gehörende geheime Schlüssel müssen verschlüsselt abgespeichert werden.



Der geheime Schlüssel muss sicher verwahrt und vor dem Zugriff durch Dritte geschützt werden. Er darf nicht weitergegeben werden. Das Vier-Augen-Prinzip findet für End-Entity-Zertifikate und zugehörige Schlüssel jedoch keine Anwendung.

Die Schlüssellängen haben sich an den Vorgaben der definierten Zertifikatsprofile zu orientieren.

3.6 Zertifizierungsvorgang

Der eigentliche Zertifizierungsvorgang ist abhängig von der Art der gewählten PKI-Lösung. Besonders das Generieren von Schlüsseln, Requests und Zertifikaten wird zum Teil sehr unterschiedlich gehandhabt. Nähere Erläuterung hierzu liefern die Abschnitte 4.2.4 und 5.2.4.

Die ausgestellten Zertifikate müssen den definierten Zertifikatsprofilen entsprechen. Vor der Ausstellung bzw. Übergabe hat sich der Zertifikatnehmer mit einem gültigen Ausweisdokument (Personalausweis oder Reisepass) gegenüber der Zertifizierungsstelle bzw. Registrierungsstelle zu identifizieren.

Eine eindeutig definierte Vorgehensweise sollte je nach Ausgestaltung der PKI im Rahmen der Ausstellererklärung/Zertifizierungsrichtlinie vereinbart und von allen beteiligten Parteien anerkannt werden.

3.7 Bereitstellung von Zertifikaten

Sämtliche Zertifikate werden sowohl in einer lokalen Datenbank auf dem dedizierten und isolierten Rechner der CA, als auch in einem zentralen für alle Nutzer zugänglichen Verzeichnisdienst gespeichert.

Dem Zertifikatnehmer wird sein Zertifikat je nach gewählter PKI-Lösung ggf. auch in anderer Form (z.B. PKCS#12-Datei auf Datenträger) zur Verfügung gestellt. Entsprechende Regelungen sind in Abhängigkeit von der gewählten Lösung zu definieren und in der Ausstellererklärung/Zertifizierungsrichtlinie festzuhalten.

3.8 Sperren von Zertifikaten

Erteilte Zertifikate können jederzeit seitens der Zertifizierungsstelle vor Ablauf der Gültigkeitsdauer gesperrt werden. Ursachen für die Sperrung können beispielsweise sein:

- Ausscheiden eines Zertifikatnehmers
- Namensänderungen
- Missbräuchliche Handlungen eines Zertifikatnehmers
- Nichtbefolgen der Zertifizierungsrichtlinie

Jeder Zertifikatnehmer kann ohne Angabe von Gründen die Sperrung seines Zertifikats verlangen. Gründe hierfür könnten sein:

- Bekanntwerden der missbräuchlichen Nutzung des geheimen Schlüssel des Zertifikatnehmers
- Der Zertifikatnehmer hat das Passwort für den privaten Schlüssel vergessen und kann ihn so nicht mehr benutzen.
- Der geheime Schlüssel ist verloren oder unwiederbringlich zerstört.

Dem Verlangen der Sperrung seitens des Zertifikatnehmers ist nachzukommen, sobald sich durch geeignete Schritte davon überzeugt wurde, dass der Antrag vom Zertifikatnehmer selbst stammt bzw. von ihm autorisiert ist. (z.B. durch Anruf und Abfrage eines zuvor vereinbarten Passworts oder persönliches Erscheinen und Ausweisen) Eine rückwirkende Sperrung ist nicht möglich. Einmal gesperrte Zertifikate können nicht erneuert werden. Jedoch hat jeder Teilnehmer die Möglichkeit, ein neues Zertifikat anzufordern.

3.9 Bereitstellen von Sperr-/Statusinformationen

Die gesperrten Zertifikate werden unverzüglich auf einer Sperrliste - der sogenannten Certificate Revocation List (CRL) - veröffentlicht, damit die entsprechenden Schlüssel nicht irrtümlicherweise benutzt werden. Gesperrte Zertifikate bleiben solange auf der CRL, bis die ursprüngliche Zertifikatgültigkeitsdauer überschritten wurde.

Die Sperrliste entspricht dem definierten Sperrlistenprofil und wird unverzüglich an den in den Zertifikatserweiterungen angegebenen Orten (z.B. LDAP-Verzeichnis) veröffentlicht.

Statusinformationen sollten zusätzlich über einen OCSP-Responder (Online Certificate Status Protocol – [RFC 2560]) zur Verfügung gestellt werden.

3.10 Gültigkeitszeiträume

Die Gültigkeitszeiträume sind den Zertifikatsprofilen zu entnehmen. Zertifikate können nicht verlängert werden. Läuft ihre Gültigkeit ab, müssen auf der Grundlage eines neu generierten Schlüsselpaares neue Zertifikate ausgestellt werden.

4 PKI-Minimallösung

Hält sich die Menge der von einer PKI auszustellenden Zertifikate in einem überschaubaren Rahmen (≤ 100), sollte nicht zuletzt aus Aufwands- und Kostengründen darüber nachgedacht werden, welchen Umfang und welche Ausgestaltung die zu wählende Lösung haben muss.

Gerade die Testphase mit ihrem sehr eingeschränkten Nutzerkreis stellt noch keine hohen Anforderungen an Automatismen und dient in erster Hinsicht dem Sammeln von Erfahrungen auf deren Grundlage eine endgültige Entscheidung für eine zukünftige Lösung getroffen werden kann.

In diesem Zusammenhang würde sich der Einsatz einer der folgenden PKI-Minimallösungsvarianten also durchaus anbieten.

4.1 Ausprägung

Bei allen der drei vorgestellten Varianten der Minimallösung kommt OpenSSL zum Einsatz. Die Steuerung des Tools kann entweder direkt über die Kommandozeile, selbstentwickelte Skripte oder speziell angepasste Tools (z.B. TinyCA) erfolgen. Es handelt sich hierbei um praktikable und zum Teil bewährte Lösungen, die trotz ihrer Einfachheit sämtliche gestellten Sicherheitsanforderungen erfüllen können. Voraussetzung dafür ist jedoch die Einbettung in eine entsprechende Organisationsstruktur.

Variante 1 - Offline-PC in einem geschützten Raum

Eine relativ aufwendige Variante ist das Aufstellen eines dedizierten Offline-PCs in einem eigens dafür vorgesehenen und verschließbaren Raum. Die Daten der Zertifizierungsstelle werden auf der ggf. verschlüsselten Festplatte des Gerätes abgelegt.

Variante 2 - Offline-Laptop in einem Safe

Steht kein Raum für das Aufstellen eines Offline-PCs zur Verfügung, lässt sich die Zertifizierungsstelle auch auf einem Laptop installieren. Dieser lässt sich relativ problemlos in einem Safe verwahren. Wird er benötigt kann er an einem beliebigen Ort betrieben werden. Wie bei der ersten Variante werden auch hier die Daten der Zertifizierungsstelle auf der Festplatte gespeichert.

Variante 3 - CD mit Live-Betriebssystem und zusätzlichem Wechseldatenträger in einem Safe

Kann kein Gerät dauerhaft als Zertifizierungsstelle dienen, besteht die Möglichkeit, eine CD mit Live-Betriebssystem zu generieren und von dieser nach dem Booten auf den verschlüsselten Wechseldatenträger zuzugreifen, wo die relevanten Daten der Zertifizierungsstelle gespeichert sind. Zusätzlich zur Flexibilität dieser Lösung wird durch den fehlenden Schreibzugriff eine absichtliche oder unabsichtliche Manipulation des Systems verhindert.

4.2 Umsetzung von Standardfunktionalitäten

Weicht die Funktionalität der Minimallösung von der in den generellen Anforderungen beschriebenen ab bzw. ergänzt diese, wird dies im folgenden Abschnitt kenntlich gemacht und dargestellt.

4.2.1 Anforderungen an die Sicherheit

Dass der Betrieb des Zertifizierungsstellensystems offline erfolgt, muss durch geeignete Maßnahmen (z.B. Deinstallation des Netzwerkkartentreibers) zu jeder Zeit sichergestellt werden.

Die infrastrukturelle Sicherheit kann durch geeignete Verfahren (Zutrittsberechtigung/Schlüsselvergabe) in allen drei Varianten gewährleistet werden. Das Aufbewahren der Zertifizierungsstelle in einem Safe (Variante 2 + 3) bietet jedoch noch ein deutlich höheres Schutzniveau im Vergleich zur ersten Variante.

Das Schlüssel- und Zertifikatsbackup der Zertifizierungsstelle erfolgt einmalig nach deren Erstellung. Das Backup wird in verschlüsselter Form auf einem Datenträger gespeichert und an einem sicheren Ort verwahrt (z.B. weiterer Safe in einem anderen Brandabschnitt). Die Indexdateien der Zertifizierungsstelle und sämtliche ausgestellten Zertifikate werden nach jeder Verwendung des Systems gesichert.

Das 4-Augen-Prinzip findet in einer relativ einfachen Ausführung seine Anwendung. Eine Möglichkeit besteht im sogenannten Passwordsharing. Dabei wird das den geheimen Schlüssel schützende Passwort geteilt. Jede Person kennt nur einen ganz bestimmten Teil. Zur Freischaltung des geheimen Schlüssels und damit zur Nutzung der CA müssen beide Teile eingegeben werden. Eine weitere Möglichkeit besteht in der doppelten Verschlüsselung des Schlüssels. Somit kennt die eine Person das Passwort für die erste "Hülle" und die Andere für die Zweite.

4.2.2 Bereitstellung einer Ausstellererklärung/Zertifizierungsrichtlinie

Auf die Bereitstellung einer Ausstellererklärung/Zertifizierungsrichtlinie kann im Testbetrieb einer PKI durchaus verzichtet werden. Wird später in den Produktivbetrieb gewechselt, sollten die Dokumente jedoch vorhanden sein und in den Zertifikatserweiterungen auch auf sie verwiesen werden.



4.2.3 Umgang mit Schlüsseln

Die PKI in der Minimallösung übernimmt gleichzeitig die Rolle eines Trust-centers. Das heißt, dass in der CA neben dem Zertifikat auch das asymmetrische Schlüsselpaar erzeugt wird. Eine Generierung des Schlüsselpaars über OpenSSL durch den Zertifikatnehmer wäre zwar auch denkbar, aber sollte aufgrund der Komplexität und fehlender Tools vermieden werden.

Nach der geeigneten Übergabe von Schlüsselpaar und Zertifikat an den Zertifikatnehmer wird der geheime Schlüssel sicher gelöscht. Die langfristige Speicherung zu Zwecken des Key-Backups ist nicht vorgesehen.

4.2.4 Zertifizierungsvorgang

Für die Funktion der CA als Trust-Center, welche das asymmetrische Schlüsselpaar zentral für den Zertifikatnehmer erstellt, sind folgende Schritte vorzunehmen:

- 1 Der Zertifikatnehmer meldet sich mit dem Zertifizierungswunsch bei einem ihm benannten Mitarbeiter der Zertifizierungsstelle.
- 2 Die Zertifizierungsstelle überprüft, ob der Zertifikatnehmer den in Abschnitt 3.4 formulierten Anforderungen entspricht.
- 3 Die Zertifizierungsstelle erzeugt für den Zertifikatnehmer eine sogenannte Personal Security Environment (PSE), indem sie ein asymmetrisches Schlüsselpaar generiert und den öffentlichen Teil des Schlüsselpaares zertifiziert.
- 4 Die Zertifizierungsstelle überträgt die durch eine Transport-PIN gesicherte PSE als PKCS#12-Datei auf einen Datenträger (USB-Stick, Diskette, o. ä.).
- 5 Die Zertifizierungsstelle überzeugt sich von der Identität des Zertifikatnehmers (in der Regel Personalausweis oder Reisepass) und händigt dem Zertifikatnehmer die PSE gemeinsam mit einer umfassenden Dokumentation zu deren Installation aus.
- 6 Der Zertifikatnehmer bestätigt handschriftlich die Teilnehmererklärung der Zertifizierungsstelle, in der der Empfang der PSE bestätigt und der Zertifizierungsrichtlinie der Zertifizierungsstelle zustimmt wird. Dabei erhält er zusätzlich den öffentlichen Zertifizierungsschlüssel der Zertifizierungsstelle (Zertifikat Fingerprint).

- 7 Der Transport-PIN wird dem Zertifikatnehmer auf geeignetem Weg mitgeteilt (z. B. per Post in einem verschlossenen Umschlag).
- 8 Die Zertifizierungsstelle vernichtet bei sich die PSE des Zertifikatnehmers nachhaltig.
- 9 Die Zertifizierungsstelle liefert das Zertifikat für die Übernahme in den zentralen Verzeichnisdienst ab (z. B. als LDIF-File).

4.3 Vorteile

Die aufgezeigte Minimallösung zeichnet sich durch außerordentlich **geringe Kosten** bei der Implementierung aus. Sie können jedoch je nach gewählter Variante leicht variieren.

Die relativ **geringe Komplexität** der Lösung garantiert einen sehr **kurzen Umsetzungszeitraum**.

Die dargestellte Lösung kann auf **beliebigen Systemen** umgesetzt werden.

4.4 Nachteile

Durch **fehlende Automatismen** kann von einem relativ **hohen Anteil manueller Schritte** ausgegangen werden. Dies kann beispielsweise bei der Anbindung an Verzeichnisdienste ein Problem darstellen.

Ferner kann davon ausgegangen werden, dass es durch unsachgemäße Bedienung bzw. das Fehlen von Mechanismen, die diese verhindern, zu einer **höheren Fehleranfälligkeit** (z. B. falsch ausgestellte Zertifikate) der Lösung kommt.



5 PKI-Optimallösung

Übersteigt die Menge der von einer PKI auszustellenden Zertifikate einen bestimmten Rahmen (>200), sollte nicht zuletzt aus Gründen eines effizienten PKI-Betriebs darüber nachgedacht werden, von einer Minimallösung Abstand zu nehmen und einen größeren Aufwand bei der Implementierung zu akzeptieren.

5.1 Ausprägung

Es gibt eine ganze Reihe von Lösungen, sowohl kostenpflichtiger als auch kostenloser (OpenCA, OpenXPKI, EJBCA) Natur, die umfassende PKI-Funktionalitäten zur Verfügung stellen. Allen gemeinsam ist eine sehr klare Trennung zwischen den Funktionsbereichen der PKI.

Einen wesentlichen Unterschied zur Minimallösung stellt die Unterscheidung in Certification Authority (CA) und Registration Authority (RA) dar. Die Darstellung des Zertifizierungsvorgangs (vgl. Abschnitt 5.2.4) wird dies noch deutlicher machen.

Das Ergebnis bleibt jedoch entsprechend der Minimallösung das Gleiche. Es wird ein Zertifikat für den Zertifikatnehmer ausgestellt. Nur der Weg zu diesem Ziel ändert sich.

Ein möglicher Aufbau könnte sich wie folgt darstellen:

5.2 Umsetzung von Standardfunktionalität

Weicht die Funktionalität der Optimallösung von der in den generellen Anforderungen beschriebenen ab bzw. ergänzt diese, wird dies im folgenden Abschnitt kenntlich gemacht und dargestellt.

5.2.1 Anforderung an die Sicherheit

Es muss sichergestellt werden, dass das für die Ausstellung der Zertifikate verwendete System immer offline betrieben wird. Dies muss durch geeignete Maßnahmen (z.B. Deinstallation des Netzwerkkartentreibers) zu jeder Zeit sichergestellt werden. Ein Datenaustausch mit dem CA-System darf ausschließlich über mobile Datenträger erfolgen.

Die infrastrukturelle Sicherheit kann durch geeignete Verfahren (Zutrittsberechtigung/Schlüsselvergabe) sichergestellt werden. Handelt es sich beim CA-System um einen Laptop, so kann dieser - wie schon bei der Minimallösung - in einem Safe aufbewahrt werden. Dadurch erhöht sich das erreichte Schutzniveau nochmals deutlich.

Für das Schlüsselbackup gelten die gleichen Anforderungen wie für die Minimallösung.

Die meisten Systeme bieten eine gute Implementierung des Vier-Augen-Prinzips. Zur Freischaltung des geheimen Schlüssels müssen die Passwörter zweier Personen eingegeben werden. Systeme im Hochsicherheitsbereich können zusätzlich dazu den geheimen Schlüssel auch auf speziell geschützter Hardware (HSM) verwahren. Zur Freischaltung des geheimen Schlüssels wird dann beispielsweise eine Zwei-Faktoren-Authentisierung (Smartcard + PIN) direkt auf dem Gerät durchgeführt.

5.2.2 Bereitstellung einer Ausstellererklärung/Zertifizierungsrichtlinie

Im Rahmen eines Regelbetriebs sollte auf die Bereitstellung einer Ausstellererklärung/Zertifizierungsrichtlinie großer Wert gelegt werden. Es ist jedoch darauf zu achten, dass in ihr nur wirklich wesentliche Punkte erörtert werden. Zu lange und komplexe Dokumente werden nicht gelesen. Eine Auseinandersetzung mit dem Inhalt erfolgt dementsprechend nicht.

Entsprechend der in den Zertifikatsprofilen getroffenen Festlegungen, sollte die CertificatePolicies-Extension auf die Zertifizierungsrichtlinie verweisen.

5.2.3 Umgang mit Schlüsseln

Im Gegensatz zur Minimallösung können Schlüsselpaare sowohl im Browser des Clients, als auch durch die CA direkt generiert werden.

Erfolgt die Generierung in der CA gelten die Anforderungen entsprechend der Minimallösung.

Die Schlüsselgenerierung durch den Client stellt jedoch eine deutlich attraktivere Lösung dar. Der geheime Schlüssel verbleibt im Clientsystem und muss nicht auf gesichertem Wege von der CA zum Client gelangen. Lediglich ein Certificate Signing Request (CSR) wird vom Client versendet. Dieser enthält jedoch nicht den geheimen Schlüssel.



Somit kommt die Zertifizierungsstelle zu keiner Zeit mit dem geheimen Schlüssel des Zertifikatnehmers in Kontakt und kann damit auch nicht für dessen Kompromittierung verantwortlich gemacht werden.

5.2.4 Zertifizierungsvorgang

Je nach eingesetzter PKI-Lösung wird sich die konkrete Ausgestaltung der unten angegebenen Schritte leicht unterscheiden. Der hier beschriebene Ablauf sollte sich jedoch in allen Lösungen umsetzen lassen.

Fungiert die Zertifizierungsstelle wie schon in der Minimallösung als Trust-center ergibt sich folgende Vorgehensweise:

- 1 Der Zertifikatnehmer meldet sich mit dem Zertifizierungswunsch bei seiner Registration Authority (RA).
- 2 Die RA überprüft, ob der Zertifikatnehmer den in Abschnitt 3.3 formulierten Anforderungen entspricht.
- 3 Die RA übermittelt alle für die Erstellung einer Personal Security Environment (PSE) notwendigen Angaben an die CA.
- 4 Die CA erzeugt auf Grundlage dieser Angaben die PSE, indem sie ein asymmetrisches Schlüsselpaar generiert und den öffentlichen Teil des Schlüsselpaares zertifiziert.
- 5 Die CA überträgt die durch einen Transport-PIN gesicherte PSE als PKCS#12-Datei auf einen Datenträger (USB-Stick, Diskette, o. ä.) und übergibt diesen an die zuständige RA.
- 6 Die RA überzeugt sich von der Identität des Zertifikatnehmers (in der Regel Personalausweis oder Reisepass) und händigt dem Zertifikatnehmer die PSE gemeinsam mit einer umfassenden Dokumentation zu deren Installation aus.
- 7 Der Zertifikatnehmer bestätigt handschriftlich die Teilnehmererklärung der CA, in der der Empfang der PSE bestätigt und der Zertifizierungsrichtlinie der CA zugestimmt wird. Dabei erhält er zusätzlich den öffentlichen Zertifizierungsschlüssel der CA (Zertifikat Fingerprint).
- 8 Die RA übermittelt die Teilnehmererklärung an die CA, die daraufhin dem Zertifikatnehmer die Transport-PIN zusendet (z. B. per Post in einem verschlossenen Umschlag).

- 9 Die CA vernichtet bei sich die PSE des Zertifikatnehmers nachhaltig.
- 10 Die CA liefert das Zertifikat für die Übernahme in den zentralen Verzeichnisdienst ab (z. B. als LDIF-File).

Für die Funktion der CA als Zertifizierungsinstanz, die ausschließlich Zertifizierungsanforderungen von Zertifikatnehmern zertifiziert, sind folgende Schritte vorzunehmen:

- 11 Die RA erhält vom Zertifikatnehmer den Zertifizierungsrequest, der beispielsweise über eine Weboberfläche generiert wurde.
- 12 Die RA überprüft, ob die Angaben in der Zertifizierungsanforderung stimmig sind.
- 13 Die RA überzeugt sich von der Identität des Zertifikatnehmers (in der Regel Personalausweis oder Reisepass).
- 14 Der Zertifikatnehmer hat vor der Zertifizierung seines öffentlichen Schlüssels durch die CA handschriftlich die Teilnehmererklärung der CA, in der der Zertifizierungsrichtlinie der CA zugestimmt wird, zu unterzeichnen. Dabei erhält er den öffentlichen Zertifizierungsschlüssel der CA (Zertifikat Fingerprint).
- 15 Die RA überprüft, ob der Zertifizierungsrequest über eine definierte Schlüsselmindestlänge verfügt. (vgl. Zertifikatsprofile)
- 16 Die RA übermittelt die Zertifizierungsanforderung an die CA.
- 17 Die CA zertifiziert die Zertifizierungsanforderung und übermittelt die Zertifizierungsantwort (PKCS#7) an den Zertifikatnehmer, ggf. über seine RA.
- 18 Die CA liefert das Zertifikat für die Übernahme in den zentralen Verzeichnisdienst ab (z. B. als LDIF-File).

5.3 Vorteile

Integrierte PKI-Systeme bieten einen relativ **hohen Automatisierungsgrad** in Bezug auf die üblichen Funktionalitäten einer Zertifizierungsstelle (Einspielen der Zertifikate in Verzeichnisdienste, Veröffentlichung von Sperrlisten, etc.).



Zusätzliche Funktionalitäten wie z. B. die Schlüsselerzeugung im Client-system oder übersichtlich gestaltete Verwaltungsoberflächen erleichtern die tägliche Arbeit gerade wenn eine große Anzahl an Zertifikaten ausgestellt werden muss.

Durch automatische Prüfroutinen können im Vergleich zur Minimallösung **geringere Fehlerquoten** bei der Erstellung der Zertifikate erreicht werden.

5.4 Nachteile

Im Gegensatz zur Minimallösung entsteht ein relativ **hoher Planungs-, Installations- und Konfigurationsaufwand**. Kommt keine OpenSource-Lösung zum Einsatz kann zusätzlich mit hohen Kosten für die Zertifizierungsstellensoftware gerechnet werden.

Der Einsatz von mehreren IT-Systemen wird aufgrund der funktionalen Trennung unvermeidbar. Damit verbunden ist ein **hoher Aufwand für die Sicherung der Infrastruktur** (zusätzliche Räume, Zutrittsregelungen, etc.).

A Struktur der Beispielzertifikate

```

Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      cf:26:70:f0:dc:7c:e7:15
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=DE, O=Klinik XYZ AG, CN=Klinik XYZ Root CA 1
    Validity
      Not Before: Mar 12 10:59:02 2007 GMT
      Not After : Mar 11 10:59:02 2010 GMT
    Subject: C=DE, O=Klinik XYZ AG, CN=Klinik XYZ Root CA 1
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (4096 bit)
      Modulus (4096 bit):
        00:a5:2b:e9:2c:5b:91:a7:62:03:97:d0:fb:17:10:
        37:b7: ...
        ... :5c:ad:c8:23:e0:9d:91:ae:2c:de:90:
        65:21:39:96:30:98:a6:62:2a:40:c7:e3:91:1e:be:
        9e:c0:47
      Exponent: 65537 (0x10001)
    X509v3 extensions:
      X509v3 Subject Key Identifier:
        4C:01:74:8B:AF:07:12:24:E9:1B:3D:41:C5:AC:EC:28:C7:2D:A7:75
      X509v3 Authority Key Identifier:
        keyid:4C:01:74:8B:AF:07:12:24:E9:1B:3D:41:C5:AC:EC:28:C7:2D:A7:75
      X509v3 Key Usage: critical
        Certificate Sign, CRL Sign
      X509v3 Basic Constraints: critical
        CA:TRUE
      X509v3 Certificate Policies:
        Policy: 1.2.3.4.5.6.7.8.9.10.11
      X509v3 CRL Distribution Points:
        URI:ldap://ldap.klinik-xyz.de:389/CN=CRL,O=Klinik XYZ AG,C=DE,dc=ldap,dc=klinik-xyz,
        dc=de?certificateRevocationList;binary?base?objectClass=cRLDistributionPoint
      Authority Information Access:
        OCSP - URI:http://pki.klinik-xyz.de/ocsp
    Signature Algorithm: sha1WithRSAEncryption
    43:7e:7d:1e:1b:42:9e:76:eb:b1:1b:b5:f4:61:27:96:b5:70:
    ac:2f: ...
    ... :94:68:ab:b5:67:b5:59:0b:e1:b0:07:6b:dd:69:
    4a:a5:ab:30:0f:a4:12:f9
  
```

CA-Zertifikat

```

Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 2 (0x2)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=DE, O=Klinik XYZ AG, CN=Klinik XYZ Root CA 1
    Validity
      Not Before: Mar 12 11:57:53 2007 GMT
      Not After : Mar 11 11:57:53 2008 GMT
    Subject: C=DE, O=Klinik XYZ AG, CN=eCR Identity Provider, GN=Hubert, SN=Holle
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (2048 bit)
        Modulus (2048 bit):
          00:c8:3b:48:a7:85:cf:55:cc:c8:7e:0e:70:6c:ae:
          61:6b: ...
          ... :f9:96:91:d5:d0:d0:b3:c8:04:a4:8d:
          e7:40:20:4c:c2:e0:46:d9:a0:de:56:dd:ef:f3:8e:
          c0:1b
        Exponent: 65537 (0x10001)
    X509v3 extensions:
      X509v3 Subject Key Identifier:
        10:2D:81:77:A0:59:F9:DF:CA:AA:97:13:41:FB:98:0A:62:40:97:27
      X509v3 Authority Key Identifier:
        keyid:4C:01:74:8B:AF:07:12:24:E9:1B:3D:41:C5:AC:EC:28:C7:2D:A7:75
      X509v3 Key Usage: critical
        Digital Signature, Key Encipherment, Data Encipherment
      X509v3 Certificate Policies:
        Policy: 1.2.3.4.5.6.7.8.9.10.11
      X509v3 Basic Constraints: critical
        CA:FALSE
      X509v3 CRL Distribution Points:
        URI:ldap://ldap.klinik-xyz.de:389/CN=CRL,O=Klinik XYZ AG,C=DE,dc=ldap,dc=klinik-xyz,
        dc=de?certificateRevocationList;binary?base?objectClass=cRLDistributionPoint
      Authority Information Access:
        OCSP - URI:http://pki.klinik-xyz.de/ocsp

    Signature Algorithm: sha1WithRSAEncryption
    7e:5c:f7:71:59:c6:eb:30:0a:99:f4:27:1c:eb:d2:5c:ab:c2:
    7e:f6: ...
    ... :c8:52:af:e7:9a:5a:1d:2c:d3:19:48:29:d8:2b:
    f2:22:3f:51:32:9a:9d:e9
  
```

eCR-Servicezertifikat

```
Certificate:
Data:
  Version: 3 (0x2)
  Serial Number: 12 (0xc)
  Signature Algorithm: sha1WithRSAEncryption
  Issuer: C=DE, O=Klinik XYZ AG, CN=Klinik XYZ Root CA 1
  Validity
    Not Before: Mar 22 09:17:33 2007 GMT
    Not After : Mar 21 09:17:33 2009 GMT
  Subject: C=DE, O=Klinik XYZ AG, CN=Arztpraxis Dr. Muetzelmann/title=Dr., GN=Karl-Heinz, SN=Muetzelmann
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (2048 bit)
      Modulus (2048 bit):
        00:b0:96:c2:ca:bd:2f:30:96:8c:02:e7:2f:b8:98:
        1b:ad: ...
        ... :19:5b:86:43:68:72:d0:dd:b7:74:d6:
        28:18:36:84:48:f0:20:e0:da:4e:a9:ab:d9:5b:86:
        dc:17
      Exponent: 65537 (0x10001)
  X509v3 extensions:
    X509v3 Subject Key Identifier:
      14:E6:1E:0F:5D:D8:B3:06:32:EE:E3:8D:62:FA:88:8B:8B:29:BD:47
    X509v3 Authority Key Identifier:
      keyid:4C:01:74:8B:AF:07:12:24:E9:1B:3D:41:C5:AC:EC:28:C7:2D:A7:75
    X509v3 Key Usage: critical
      Digital Signature
    X509v3 Extended Key Usage:
      TLS Web Client Authentication
    X509v3 Basic Constraints: critical
      CA:FALSE
    X509v3 Certificate Policies:
      Policy: 1.2.3.4.5.6.7.8.9.10.11
    X509v3 CRL Distribution Points:
      URI:ldap://ldap.klinik-xyz.de:389/CN=CRL,O=Klinik XYZ AG,C=DE,dc=ldap,dc=klinik-xyz,
      dc=de?certificateRevocationList;binary?base?objectClass=cRLDistributionPoint
    Authority Information Access:
      OCSP - URI:http://pki.klinik-xyz.de/ocsp
  Signature Algorithm: sha1WithRSAEncryption
  09:3a:3d:30:2c:03:85:76:a5:c0:c1:5b:76:ee:b4:7d:2f:c8:
  b4:c0: ...
  ... :56:c4:99:8f:08:38:5c:6e:25:ec:f1:c6:32:fc:
  48:14:a4:26:30:7d:5a:32
```

Clientauthentisierungszertifikat

```

Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 6 (0x6)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=DE, O=Klinik XYZ AG, CN=Klinik XYZ Root CA 1
    Validity
      Not Before: Mar 12 12:02:30 2007 GMT
      Not After : Mar 11 12:02:30 2008 GMT
    Subject: C=DE, O=Klinik XYZ AG, CN=www.klinik-xyz.de, GN=Hubert, SN=Holle
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (2048 bit)
      Modulus (2048 bit):
        00:c1:6b:c0:51:a5:2b:34:21:94:9b:d3:bf:f9:56:
        e7:b3: ...
        ... :38:d4:12:b0:70:4d:3e:91:73:8c:13:
        52:83:61:39:fa:a1:d6:93:d5:25:9c:f5:72:f5:b0:
        d6:ab
      Exponent: 65537 (0x10001)
    X509v3 extensions:
      X509v3 Subject Key Identifier:
        1F:0A:34:46:08:70:AD:7C:65:B2:FE:E2:93:30:6F:73:C6:32:5D:EF
      X509v3 Authority Key Identifier:
        keyid:4C:01:74:8B:AF:07:12:24:E9:1B:3D:41:C5:AC:EC:28:C7:2D:A7:75
      X509v3 Key Usage: critical
        Key Encipherment
      X509v3 Certificate Policies:
        Policy: 1.2.3.4.5.6.7.8.9.10.11
      X509v3 Basic Constraints: critical
        CA:FALSE
      X509v3 Extended Key Usage:
        TLS Web Server Authentication
      X509v3 CRL Distribution Points:
        URI:ldap://ldap.klinik-xyz.de:389/CN=CRL,O=Klinik XYZ AG,C=DE,dc=ldap,dc=klinik-xyz,
        dc=de?certificateRevocationList;binary?base?objectClass=cRLDistributionPoint
      Authority Information Access:
        OCSP - URI:http://pki.klinik-xyz.de/ocsp
    Signature Algorithm: sha1WithRSAEncryption
      87:9b:29:2f:7c:97:8e:d4:2c:7b:86:f6:e9:3b:43:cf:0e:52:
      97:f8: ...
      ... :39:dc:09:30:3e:da:6d:bb:85:a2:d9:44:d2:f2:
      5b:3a:e3:de:11:8f:14:cb
  
```

SSL-Serverzertifikat

```
Certificate Revocation List (CRL):
  Version 2 (0x1)
  Signature Algorithm: sha1WithRSAEncryption
  Issuer: /C=DE/O=Klinik XYZ AG/CN=Klinik XYZ Root CA 1
  Last Update: Mar 13 08:56:42 2007 GMT
  Next Update: Apr 12 08:56:42 2007 GMT
  CRL extensions:
    X509v3 Authority Key Identifier:
      keyid:4C:01:74:8B:AF:07:12:24:E9:1B:3D:41:C5:AC:EC:28:C7:2D:A7:75
    X509v3 CRL Number:
      1
  Revoked Certificates:
    Serial Number: 08
    Revocation Date: Mar 13 08:55:32 2007 GMT
    Signature Algorithm: sha1WithRSAEncryption
    0e:fb:19:19:3c:bd:d2:2d:ef:a1:d1:28:35:50:94:9e:12:f4:
    b5:4d: ...
    ... :2e:09:d5:1e:07:2c:13:77:58:9e:09:1f:ec:d2:
    7b:34:31:83:fa:69:c6:b7
```

Sperrliste

B Aufbau der Konfigurationsdatei (openssl.cnf)

```
# -----
# Basiskonfiguration
# -----
HOME = .
RANDFILE = $ENV::HOME/.rnd

# -----
# Konfigurationseinstieg für CA
# -----
[ ca ]
default_ca = CA_default

# -----
# Konfiguration der CA (Directories & Co.)
# -----
[ CA_default ]

dir = C:/temp/CA
certs = $dir/certs
crl_dir = $dir/crl
database = $dir/index.txt
new_certs_dir = $dir/newcerts
certificate = $dir/cacert.pem
serial = $dir/serial
crlnumber = $dir/crlnumber
crl = $dir/crl.pem
private_key = $dir/private/cakey.pem
RANDFILE = $dir/private/.rand

unique_subject = yes
preserve = no

policy = policy_match

x509_extensions = client_auth_ext
crl_extensions = crl_ext

name_opt = ca_default
cert_opt = ca_default

default_days = 365
default_crl_days = 30
default_md = sha1

# -----
# CA Policy
# -----
[ policy_match ]
```

```
countryName = match
stateOrProvinceName = match
organizationName = match
organizationalUnitName = optional
commonName = supplied
emailAddress = optional

# -----
# Clientzertifikat Policy
# -----
[ policy_client ]
countryName = match
stateOrProvinceName = optional
localityName = optional
organizationName = match
organizationalUnitName = optional
commonName = supplied
title = optional
givenName = supplied
surname = supplied

# -----
# Server-/Servicezertifikat Policy
# -----
[ policy_service ]
countryName = match
stateOrProvinceName = optional
localityName = optional
organizationName = match
organizationalUnitName = optional
commonName = supplied
title = optional
givenName = supplied
surname = supplied

# -----
# Request Optionen
# -----
[ req ]
default_bits = 2048
default_keyfile = privkey.pem
distinguished_name = req_distinguished_name
x509_extensions = default_ext
default_md = sha1
utf8 = yes
string_mask = utf8only

# -----
# Request Distinguished Name Optionen
# -----
[ req_distinguished_name ]
countryName = Laendername (2 letter code)
countryName_default = DE
countryName_min = 2
```

```

countryName_max = 2

#stateOrProvinceName = Name des Bundeslandes
#stateOrProvinceName_default = Berlin

#localityName = Name des Ortes (z.B. Stadt)
#localityName_default = Berlin

0.organizationName = Organization Name (z.B. Firmenbezeichner)
0.organizationName_default = Klinik XYZ AG

#organizationalUnitName = Organizational Unit Name (z.B. Bereich)
#organizationalUnitName_default =

commonName = Common Name ( [Titel Vorname(n) Nachname(n) (Initialen)] / [DNS-
Servername] / [Dienstname] )
commonName_max = 64

title = Titel
title_max = 64

givenName = Vorname
givenName_max = 64

surname = Nachname
surname_max = 64

# -----
# Erweiterung für Default
# -----
[ default_ext ]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid
# subjectAltName = URI:ldap://
# issuerAltName = URI:ldap://
# keyUsage = critical,
# certificatePolicies = 1.2.3.4.5.6.7.8.9.10.11
basicConstraints = critical,CA:false
# extendedKeyUsage = 1.3.6.1.5.5.7.3.1
# crlDistributionPoints = @cdpsection
# authorityInfoAccess = OCSP;URI:http://pki.klinik-xyz.de/ocsp

# -----
# Erweiterung für SSL-Serverzertifikate
# -----
[ ssl_server_ext ]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid
# subjectAltName = URI:ldap://
# issuerAltName = URI:ldap://
keyUsage = critical, keyEncipherment
certificatePolicies = 1.2.3.4.5.6.7.8.9.10.11
basicConstraints = critical,CA:false
extendedKeyUsage = 1.3.6.1.5.5.7.3.1
crlDistributionPoints = @cdpsection
authorityInfoAccess = OCSP;URI:http://pki.klinik-xyz.de/ocsp

```

```
# -----  
# Erweiterung für eCR-Servicezertifikate  
# -----  
[ ecr_service_ext ]  
subjectKeyIdentifier = hash  
authorityKeyIdentifier = keyid  
# subjectAltName = URI:ldap://  
# issuerAltName = URI:ldap://  
keyUsage = critical, digitalSignature, keyEncipherment, dataEncipherment  
certificatePolicies = 1.2.3.4.5.6.7.8.9.10.11  
basicConstraints = critical,CA:false  
crlDistributionPoints = @cdpsection  
authorityInfoAccess = OCSP;URI:http://pki.klinik-xyz.de/ocsp  
  
# -----  
# Erweiterung für Clientauthetisierungszertifikate  
# -----  
[ client_auth_ext ]  
subjectKeyIdentifier = hash  
authorityKeyIdentifier = keyid  
# subjectAltName = URI:ldap://  
# issuerAltName = URI:ldap://  
keyUsage = critical, digitalSignature  
extendedKeyUsage = 1.3.6.1.5.5.7.3.2  
basicConstraints = critical,CA:false  
certificatePolicies = 1.2.3.4.5.6.7.8.9.10.11  
crlDistributionPoints = @cdpsection  
authorityInfoAccess = OCSP;URI:http://pki.klinik-xyz.de/ocsp  
  
# -----  
# Erweiterungen für CA-Zertifikat  
# -----  
[ v3_ca ]  
subjectKeyIdentifier = hash  
authorityKeyIdentifier = keyid  
# subjectAltName = URI:ldap://  
# issuerAltName = URI:ldap://  
keyUsage = critical, cRLSign, keyCertSign  
basicConstraints = critical,CA:true  
certificatePolicies = 1.2.3.4.5.6.7.8.9.10.11  
crlDistributionPoints = @cdpsection  
authorityInfoAccess = OCSP;URI:http://pki.klinik-xyz.de/ocsp  
  
# -----  
# Erweiterungen für CRL  
# -----  
[ crl_ext ]  
authorityKeyIdentifier=keyid  
  
# -----  
# Sektion für cRLDistributionPoints  
# -----
```

```
[ cdpsection ]
URI.1=ldap://ldap.klinik-xyz.de:389/CN=CRL,O=Klinik XYZ
AG,C=DE,dc=ldap,dc=klinik-
xyz,dc=de?certificateRevocationList;binary?base?objectClass=cRLDistributionPoint
# URI.2=http://pki.klinik-xyz.de/certs/klinik-xyz-crl.crl
```

C Abkürzungsverzeichnis

3DES	Three-fold application of the DES algorithm
AES	Advanced Encryption Standard
ASN.1	Abstract Syntax Notation 1
BER	Basic Encoding Rules (for ASN.1)
BMG	Bundesministerium für Gesundheit (GER)
BSI	Bundesamt für Sicherheit in der Informationstechnik (GER)
CA	Certification Authority
CRL	Certificate Revocation List
DER	Distinguished Encoding Rules (for ASN.1)
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
ECB	Electronic Codebook (cipher mode)
eCR	electronic Case Record
FIPS	Federal Information Processing Standard (US)
HBA	Heilberufsausweis (see HPC)
HPC	Health Professional Card
HSM	Hardware Security Module
IHE	Integrating the Healthcare Enterprise
NIST	National Institute of Standards and Technology (US)
OID	(unique) Object Identifier
PKI	Public Key Infrastructure
PSE	Personal Security Environment
RSA	Rivest/Shamir/Adleman algorithm
SAML	Security Assertion Markup Language
SHA	Secure Hash Algorithm
SMC	Secure Module Card
SSL	Secure Socket Layer
TDEA	see 3DES
TLS	Transport Layer Security

D Literatur

- [BNA 9655] Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahn: Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung (Übersicht über geeignete Algorithmen). Version vom 22.02.07
<http://www.bundesnetzagentur.de/media/archive/9655.pdf>
- [BSI PKI-1] Bundesamt für Sicherheit in der Informationstechnik; securvo Security Consulting GmbH: Zertifizierungsinfrastruktur für die PKI-1-Verwaltung: Umsetzungskonzept für die Anwendung von SSL. Version 1.4 vom 10.12.02.
<http://www.bsi.bund.de/fachthem/verwpki/dokumente/101202BSI-SSL-Umsetzungskonzeptv1-4.pdf>
- [eCR_CC-1.2] eFA Konsortium: Specification of an Architecture for the Cooperative Use of Medical Data – Cryptographic Keys and Algorithms. Version 1.2 vom 21.06.07.
- [HPC Part 2] Bundesministerium für Gesundheit: Spezifikation des elektronischen Heilberufsausweises. Teil II: HBA – Anwendungen und Funktionen. 15.09.06
- [HPC Part 3] Bundesministerium für Gesundheit: Spezifikation des elektronischen Heilberufsausweises. Teil III: SMC – Anwendungen und Funktionen. 15.09.06
- [ISIS MTT 1.1] Teletrust, T7: Common ISIS-MTT Specifications for Interoperable PKI Applications. Version 1.1 vom 16.03.04
- [BSI TR-03116] Bundesamt für Sicherheit in der Informationstechnik: Technische Richtlinie für die eCard-Projekte der Bundesregierung. Version 1.0 vom 23.03.07.
<http://www.bsi.de/literat/tr/tr03116/BSI-TR-03116.pdf>
- [ISO/IEC9834/1] ITU-T Recommendation X.660 (1992), ISO/IEC 9834-1: 1993, Information Technology – Open Systems Interconnection – Systems Management Overview – Procedures for the Operation of OSI Registration Authorities: General Procedures
- [NIST 800-57] National Institute of Standards and Technology: NIST Special Publication 800-57: Recommendation for Key Management – Part 1: General. May 2006.
<http://csrc.nist.gov/publications/nistpubs/800-57/SP800-57-Part1.pdf>



- [openssl] openssl: The Open Source Toolkit for SSL/TLS. Project homepage:
<http://www.openssl.org/>
- [RFC2119] Bradner, S.: Key words for use in RFCs to Indicate Requirement
Levels; Harvard University, Boston, Massachusetts, 1997.
- [RFC 3280] Housley, R.; Polk, W.; Ford, W.; Solo, D.: Internet X.509 Public Key
Infrastructure Certificate and Certificate Revocation List (CRL) Pro-
file. April 2002.
- [RFC3369] Housley, R.: Cryptographic Message Standard. RSA Laboratories,
2002.
- [RSA] Rivest, R.; Shamir, A.; Adleman, L.: A method for obtaining digital
signatures and public key cryptosystems, Communications of the
ACM, Vol. 21 No. 2, 1978