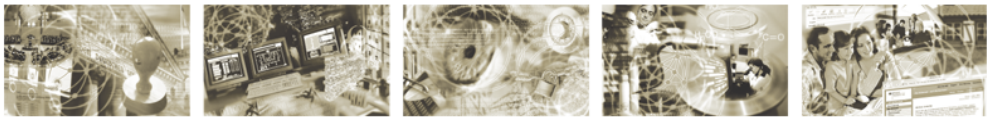
Bundesamt
für Sicherheit in der
Informationstechnik



Technische Richtlinie BSI TR-03109-4

Smart Metering PKI - Public Key Infrastruktur für Smart Meter Gateways

Version 1.0

Datum 18.03.2013

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
53133 Bonn
E-Mail: smartmeter@bsi.bund.de
Internet: https://www.bsi.bund.de

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Das Smart Meter Gateway (SMGW) ist die zentrale Kommunikationseinheit in der Infrastruktur eines intelligenten Messsystems. Das Gateway kommuniziert im lokalen Bereich beim Endkunden mit den elektronischen Zählern im Local Metrological Network (LMN), mit Geräten aus dem Home Area Network (HAN) und im Wide Area Network (WAN) mit autorisierten Marktteilnehmern. Außerdem ermöglicht das SMGW die Verbindungsaufnahme von lokalen Geräten des HANs über das WAN mit autorisierten Marktteilnehmern.

Im WAN ist für die Verbindung des SMGWs zu einem autorisierten Marktteilnehmer eine gegenseitige Authentisierung der Kommunikationspartner erforderlich. Die Kommunikation erfolgt dabei stets über einen verschlüsselten, integritätsgesicherten Kanal. Zudem werden zu sendende Daten vom SMGW zusätzlich auf Datenebene für den Endempfänger verschlüsselt und signiert.

In dem vorliegenden Dokument wird die Architektur der Smart Metering - Public Key Infrastruktur (SM-PKI) spezifiziert, mit der die Authentizität der bei dieser Kommunikation eingesetzten öffentlichen Schlüssel der Kommunikationspartner auf WAN-Ebene sichergestellt wird. Technisch wird der Authentizitätsnachweis der Schlüssel über digitale Zertifikate aus der SM-PKI realisiert. Da es sich bei der Kommunikation mit dem SMGW, um eine M2M-Kommunikation handelt, werden keine Zertifikate für die Schlüssel von Personen, sondern ausschließlich für die Schlüssel von Maschinen ausgestellt.

Des Weiteren werden in dieser Technischen Richtlinie die Mindestanforderungen an die Interoperabilität und die Sicherheit der SM-PKI aufgestellt, die in der Zertifizierungsrichtlinie (Certificate Policy, CP) für die SM-PKI berücksichtigt werden müssen. Es werden Profile für die einzusetzenden Zertifikate, Sperrlisten vorgegeben. Ferner werden Protokolle für die Beantragung und Zustellung von Zertifikaten und ein Verzeichnisdienst zur Veröffentlichung der ausgestellten Zertifikate spezifiziert.

Diese Technische Richtlinie ist wie folgt gegliedert. In Kapitel 2 wird die Architektur der SM-PKI definiert. Hierbei werden speziell die Rollen in der PKI und deren Aufgabenstellung sowie die hierfür erforderlichen Schlüssel und Zertifikate beschrieben. Das Kapitel schließt mit einer Abgrenzung der SM-PKI zu den anderen in der Infrastruktur von intelligenten Messsystemen eingesetzten Zertifikaten.

In Kapitel 3 werden die Struktur und die Gültigkeitszeit von Zertifikaten definiert. Des Weiteren wird beschrieben, wie die Gültigkeit von Zertifikaten zu validieren ist. Abschließend wird das Zertifikatsmanagement der PKI-Teilnehmer erläutert, insbesondere der Zertifikatswechsel.

Danach werden in Kapitel 4 die Sperrlisten und der Sperrdienst spezifiziert. Dabei werden insbesondere die Aktualisierungs- / Prüfzeiten von Sperrlisten definiert sowie die Prozedur der Sperrlistenvalidierung festgelgt. Abschließend wird das Sperrlistenmanagement der PKI-Teilnehmer erläutert.

In Kapitel 5 werden die Sicherheitsanforderungen an die technische Speicherung der privaten Schlüssel der PKI definiert.

Das Dokument schließt mit einem Anhang, in dem die Fein-Spezifikationen enthalten sind. Diese umfassen die Zertifikats-Profile, das CRL-Profil, das Schema für Zertifikatsrequests, die Protokolle für das Zertifikatsmanagement, das LDAP-Schema des Verzeichnisdiensts sowie die von dieser Spezifikation unterstützten elliptischen Kurven.

1.1 Einordnung des Dokuments

Das vorliegende Dokument ist Teil der BSI TR-03109 SMART ENERGY **[1]** und spezifiziert die Architektur sowie die Mindestanforderungen an die Interoperabilität und Sicherheit der SM-PKI. Darüber hinaus sind insbesondere die folgenden Dokumente für die SM-PKI zu berücksichtigen:

- **Weitere Teile der Technische Richtlinie BSI TR-03109 SMART ENERGY [1], insbesondere die Teile 1, 2 und 3 (vgl. [2], [3], [4], [5]):**

  Die Technische Richtlinie BSI TR-03109 spezifiziert die Funktionalitäts-, Interoperabilitäts- und Sicherheitsanforderungen an die Einzelkomponenten in einem Smart Metering System, beschreibt die Betriebsprozesse und die technischen Rollen in der Infrastruktur von Messsystemen und liefert damit die Grundlage für die SM-PKI. Die BSI TR-03109-3 verweist für die kryptographischen Vorgaben auf die BSI TR-03116-3.

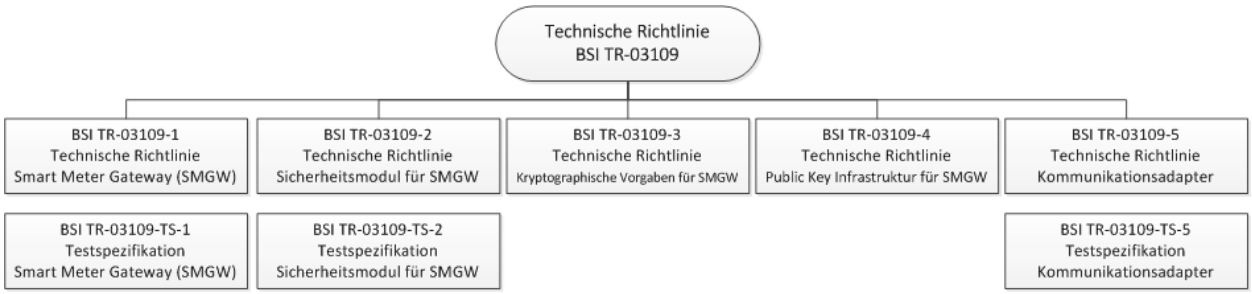


Abbildung 1: Dokumentenstruktur der BSI TR-03109

- **BSI TR-03116-3 eCard-Projekte der Bundesregierung - Kryptographische Vorgaben für die Infrastruktur von intelligenten Messsystemen [6]**:

  In dieser Technischen Richtlinie werden die Sicherheitsanforderungen für den Einsatz kryptographischer Verfahren in der Infrastruktur von Messsystemen im Energiesektor beschrieben. Insbesondere wird dort definiert, welche kryptographischen Algorithmen und Schlüssellängen für die Zertifikate in der SM-PKI eingesetzt werden müssen.

- **Certificate Policy der Root-CA (Root-CP) [7]**:

  In der Certificate Policy werden organisatorische und technische Anforderungen für das Anerkennen, Ausstellen, Verwalten, Benutzen, Zurückziehen und Erneuern von Zertifikaten zur Kommunikation zwischen SMGW und Marktteilnehmern spezifiziert. Die Certificate Policy wird von der Root-CA erstellt und hat die Mindestanforderungen aus dem hier vorliegenden Dokument zu berücksichtigen.

1.2 Terminologie

Diese Technische Richtlinie ist grundsätzlich als normativ anzusehen. Informative Teile werden explizit als solche gekennzeichnet.

Ferner wird in dieser Technischen Richtlinie in verschiedenen Tabellen folgende Terminologie verwendet:

- 'm' (mandatory): Element/Eigenschaft muss vorhanden sein.
- 'x' (do not use): Element/Eigenschaft darf nicht vorhanden sein

- 'r' (recommended): Element/Eigenschaft wird dringend empfohlen. Abweichungen sollten nur in begründeten Ausnahmefällen erfolgen.
- 'c' (conditional): Vorhandensein des/r Elements/Eigenschaft hängt von Bedingungen ab.
- 'o' (optional): Element/Eigenschaft ist optional.

1.3 Abkürzungen

Abkürzung	*Begriff*
BSI	Bundesamt für Sicherheit in der Informationstechnik
C	Certificate
CA	Certificate Authority
CP	Certificate Policy
CPS	Certificate Practices Statement
CRL	Certificate Revocation List (Zertifikatssperrliste)
DN	Distinguished Name (Eindeutiger Name)
EMT	Externer Marktteilnehmer
Enc	Encryption
ENu	Endnutzer (Z.B. Externe Marktteilnehmer, GW-Administrator, SMGW,...)
GW	Gateway
GWA	Gateway-Administrator
HAN	Home Area Network
HSM	Hardware Security Module
HTTP	Hypertext Transfer Protocol
KM	Kryptografiemodul
LDAP	Lightweight Directory Access Protocol
LDAPS	Lightweight Directory Access Protocol over SSL
LMN	Local Metrological Network
M2M	Machine to Machine
MT	Marktteilnehmer
NIST	National Institute of Standards and Technology
OID	Object Identifier
PIN	Personal Identification Number
PKI	Public Key Infrastruktur
PP	Protection Profile (Common Criteria)
RA	Registration Authoritiy

Abkürzung	*Begriff*
RDN	Relative Distinguished Name
SHA	Secure Hash Algorithm
Sign	Signature
SM	Smart Meter
SMGW	Smart Meter Gateway
SM-PKI	Smart Metering - Public Key Infrastruktur (SM-PKI)
SSL	Secure Sockets Layer
TLS	Transport Layer Security
TR	Technische Richtlinie
URI	Uniform Resource Identifier
URL	Uniform Ressource Locator
V-PKI	Verwaltungs – Public Key Infrastruktur
WAN	Wide Area Network
WSDL	Web Services Description Language
Z	Zertifikat

2 Architektur der SM-PKI

Die SM-PKI hat die folgende dreistufige hierarchische Struktur:

- **Hoheitlicher Vertrauensanker (Root-CA)**
- **Endnutzerzertifizierung (Sub-CA)**
- **Endnutzer: Marktteilnehmer, GW-Admin und SMGW**

In Abbildung 2 ist die hierarchische Struktur der SM-PKI exemplarisch dargestellt. Die Rolle der Instanzen wird im Folgenden genauer erläutert.

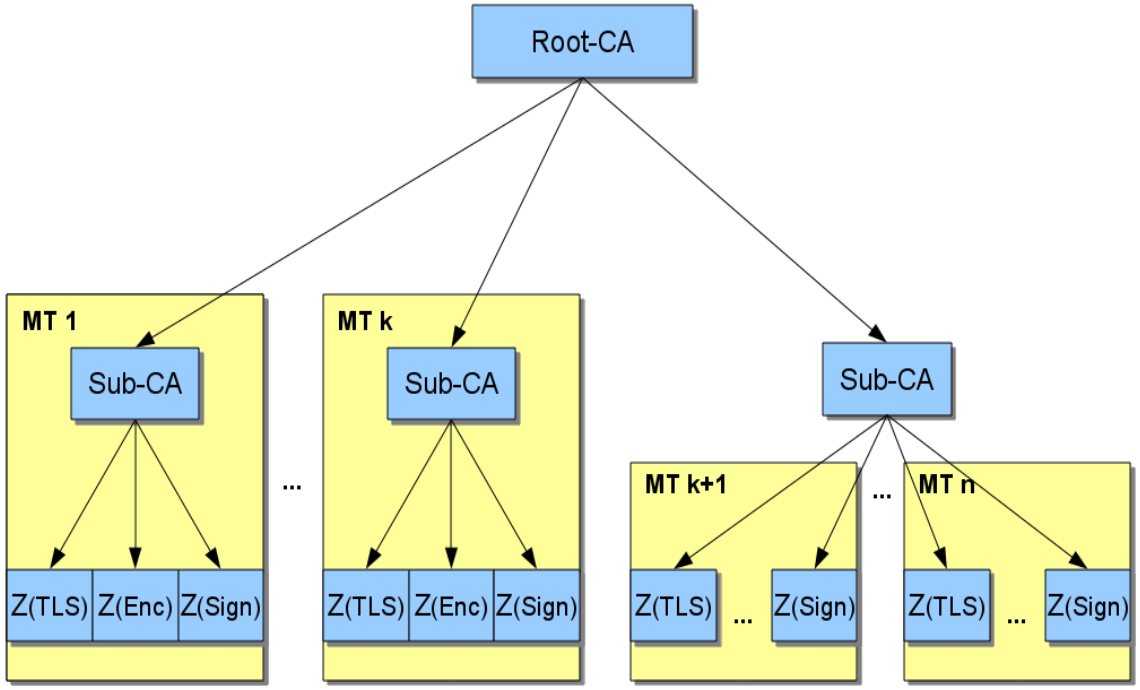


Abbildung 2: Architektur der SM-PKI (Beispiel)

Innerhalb der SM-PKI wird zwischen Zertifikatsausstellern (CAs) und Endnutzern unterschieden. Die Zertifikatsaussteller sind die Root-CA und die Sub-CAs auf den oberen beiden Ebenen der SM-PKI. Die Sub-CAs stellen dabei die Zertifikate für Endnutzer aus. Die Endnutzer bilden die untere Ebene der SM-PKI und nutzen ihre Zertifikate zur Kommunikation miteinander.

2.1 Root-CA

2.1.1 Beschreibung

Die **Root-CA** bildet den Vertrauensanker der SM-PKI. Sie stellt Zertifikate für die Sub-CAs aus.

Die technischen, personellen und organisatorischen Sicherheitsanforderungen für die Ausstellung von Zertifikaten werden von der Root-CA in einer Certificate Policy (Root-CP) unter Berücksichtigung der in diesem Dokument vorgegebenen Mindestanforderungen festgelegt.

2.1.2 Zertifikate der Root-CA

Die Root-CA besitzt ein Root-Zertifikat *C(Root)*. Es bildet den Trust-Point der SM-PKI und kann von der Root-CA bezogen werden.

Das initiale Root-Zertifikat ist mit dem privaten Schlüssel des Zertifikats selbst-signiert. Folgende Root-Zertifikate werden sowohl als selbst-signierte Zertifikate als auch Link-Zertifikate, signiert mit dem privaten Schlüssel des vorhergehenden Root-Zertifikats, veröffentlicht.

Der private Schlüssel zum Root-Zertifikat wird dazu eingesetzt, die Sub-CA-Zertifikate zu signieren.

Außerdem besitzt die Root-CA ein CRL-Signer-Zertifikat $C_{CRL\text{-}S}(Root)$ zur Signatur ihrer Sperrlisten. Das Zertifikat ist mit dem privaten Schlüssel des Root-Zertifikats *C(Root)* signiert.

Zertifikat	***Signatur des Zertifikats***	***Funktion***
C(Root)	Privater Schlüssel zu C(Root) (selbst-signiertes Zertifikat)	Trust-Point der SM-PKI Der zugehörige private Schlüssel wird zur Signatur der Sub-CA-Zertifikate und der Link-Zertifikate verwendet.
Link-C(Root)	Mit dem privaten Schlüssel des vorigen Root-Zertifikats signiert.	Das Link-Zertifikat dient zur Echtheitsprüfung eines neuen Root-Zertifikats. (Neben dem Link-Root-Zertifikat wird für dasselbe Schlüsselpaar immer auch ein selbst-signiertes Root-Zertifikat erstellt.)
$C_{CRL\text{-}S}(Root)$	Privater Schlüssel zu *C(Root)*	Der zugehörige private Schlüssel wird zur Signatur der Root-CRL verwendet.

Tabelle 1: Zertifikate der Root-CA

2.1.3 Verzeichnisse und Sperrlisten der Root-CA

Die Root-CA betreibt einen nicht-öffentlichen Verzeichnisdienst, in dem alle von ihr ausgestellten Zertifikate enthalten sind. Der Zugriff auf den Verzeichnisdienst ist den Teilnehmern der SM-PKI vorbehalten. Darüber hinaus müssen der Vertrauensanker (Root- und Link-Zertifikate) und das CRL-Signer-Zertifikat öffentlich zugänglich gemacht werden.

Ferner ist die Root-CA verantwortlich für die Erstellung, Pflege und öffentliche, vertrauenswürdige Bereitstellung aktueller Sperrlisten.

2.2 Sub-CA

2.2.1 Beschreibung

Eine **Sub-CA** ist eine Organisationseinheit, welche die Zertifikate für die Endnutzer ausstellt. Jede Sub-CA wird dazu von der Root-CA zur Ausstellung von Zertifikaten für die Endnutzer autorisiert.

Eine Sub-CA kann unternehmensintern bei einem Marktteilnehmer oder unternehmensübergreifend betrieben werden (siehe Abbildung 2). Dementsprechend kann eine Sub-CA ausschließlich einen Marktteilnehmer mit Zertifikaten versorgen oder für mehrere Endnutzer zuständig sein.

Jede Sub-CA muss eine Certificate Policy (Sub-CA CP) erstellen. Diese muss konform zu den Anforderungen der Root-CP sein.

2.2.2 Zertifikate einer Sub-CA

Eine Sub-CA besitzt ein Zertifikat *C(Sub-CA)*. Dieses wird der Sub-CA von der Root-CA ausgestellt. Der private Schlüssel zum Sub-CA-Zertifikat dient zur Signatur von Endnutzer-Zertifikaten und Sperrlisten der Sub-CA.

Zertifikat	***Signatur des Zertifikats***	***Funktion des privaten Schlüssels***
C(Sub-CA)	Privater Schlüssel zu *C(Root)*	Signatur von Endnutzerzertifikaten und Sub-CA-CRLs

Tabelle 2: Zertifikate einer Sub-CA

2.2.3 Verzeichnisse und Sperrlisten einer Sub-CA

Jede Sub-CA betreibt einen nicht-öffentlichen Verzeichnisdienst, in dem alle von ihr ausgestellten Zertifikate enthalten sind. Der Zugriff auf den Verzeichnisdienst ist den Teilnehmern der SM-PKI vorbehalten.

Außerdem ist jede Sub-CA verantwortlich für die Erstellung, Pflege und öffentliche, vertrauenswürdige Bereitstellung aktueller Sperrlisten.

2.3 Endnutzer: Externe Marktteilnehmer, GW-Administrator und SMGW

2.3.1 Beschreibung

Den Endnutzern werden Zertifikate von ihren Sub-CAs unter Berücksichtigung der Vorgaben dieser Technischen Richtlinie ausgestellt. Der Besitz dieser Zertifikate ist die Voraussetzung für eine mögliche Kommunikation zwischen den Marktteilnehmern und den SMGWs.

Es gibt verschiedene Typen von Endnutzern:

- **Externe Marktteilnehmer (EMT)**
- **Gateway-Administrator (GWA)**
- **Smart Meter Gateway (SMGW)**

Zu den externen Marktteilnehmern gehören im Kontext der PKI alle Marktteilnehmer, die potentielle Kommunikationspartner eines Smart Meter Gateway im WAN sind, etwa Verteilnetzbetreiber, Messstellenbetreiber oder Lieferanten.

Bemerkung (informativ): Der wichtigste Kommunikationspartner eines SMGW, und daher in diesem Dokument gesondert hervorgehoben, ist der Gateway-Administrator. Der Gateway-Administrator ist dafür verantwortlich seine SMGWs zu konfigurieren und zu überwachen und übernimmt in der SM-PKI die Management-Funktionen des SMGWs, welche dieses nicht selbst ausführen kann (vgl. auch 3.3.2, 4.2.3.2 sowie [2], [4]).

2.3.2 Verwendungszwecke der Zertifikate

Die Zertifikate werden zum sicheren Datenaustausch zwischen autorisierten Marktteilnehmern und dem Smart Meter Gateway eingesetzt. Da die Übermittlung von Daten zwischen SMGW und einem Marktteilnehmer auch über dritte Marktteilnehmer (etwa den Gateway-Administrator) erfolgen kann, muss einerseits die Kommunikationsverbindung abgesichert werden, andererseits müssen die Daten auf Inhaltsebene für den Endempfänger verschlüsselt und signiert werden.

Es gibt also verschiedene Verwendungszwecke für die Zertifikate. Im Folgenden werden die von der Sub-CA ausgestellten Zertifikate und deren Verwendungszwecke beschrieben:

- **TLS-Zertifikate**: Gegenseitige Authentisierung zwischen Smart Meter Gateway und autorisierten Marktteilnehmern sowie Aufbau eines verschlüsselten, integritätsgesicherten TLS-Kanals zwischen beiden.
- **Verschlüsselungszertifikate**: Ende-zu-Ende-Verschlüsselung von Daten für den Endempfänger (Datenebene, unabhängig von TLS-Verbindungen).
- **Signaturzertifikate**: Prüfung von elektronischen Signaturen von Daten (Datenebene, unabhängig von TLS-Verbindungen). Finden bei der TLS-Authentisierung keine Anwendung.

Pro Verwendungszweck müssen jeweils separate Schlüsselpaare verwendet werden. Ein privater Schlüssel darf ausschließlich für den im Zertifikat definierten Verwendungszweck genutzt werden.

2.3.3 Zertifikate eines externen Marktteilnehmers

Ein externer Marktteilnehmer erhält die benötigten Zertifikate von seiner Sub-CA.

Er muss stets ein Verschlüsselungszertifikat und ein Signaturzertifikat besitzen. Ein externer Marktteilnehmer muss ein TLS-Zertifikat besitzen, sofern er direkt mit einem SMGW kommunizieren will, ansonsten kann der EMT auch auf TLS-Zertifikate verzichten.

Zertifikat	*Signatur des Zertifikats*	*Funktion*	*m/c/o*
$C_{TLS}(EMT)$	Privater Schlüssel zu *C(Sub-CA)*	Authentisierung beim Kommunikationspartner und Aufbau eines verschlüsselten, integritätsgesicherten Kanals	c
$C_{Enc}(EMT)$	Privater Schlüssel zu *C(Sub-CA)*	Verschlüsselung für den EMT	m
$C_{Sign}(EMT)$	Privater Schlüssel zu *C(Sub-CA)*	Signatur des EMT	m

Tabelle 3: Zertifikate eines Marktteilnehmers

Autorisierte Marktteilnehmer (informativ): Ist ein Marktteilnehmer autorisiert mit einem SMGW zu kommunizieren (autorisierter Marktteilnehmer), so werden die benötigten, gültigen Zertifikate vom GWA auf dem SMGW installiert (Zur Prozessbeschreibung siehe [4]). Die Installation sowie die Prüfung der Gültigkeit der Zertifikate eines EMT ist Aufgabe des GWA. Erlischt die Autorisierung eines EMT zur Kommunikation mit dem SMGW oder wird das Zertifikat des EMT zurückgerufen, so müssen durch den GWA unverzüglich die Zertifikate des EMT vom SMGW gelöscht werden.

2.3.4 Zertifikate eines Gateway-Administrators

Der GWA besitzt ein TLS-, ein Verschlüsselungs- und ein Signaturzertifikat, welche ihm von seiner Sub-CA ausgestellt werden.

Zertifikat	*Signatur des Zertifikats*	*Funktion*	*m/c/o*
$C_{TLS}(GWA)$	Privater Schlüssel zu *C(Sub-CA)*	TLS-Zertifikat von GW-Administrator	m
$C_{Enc}(GWA)$	Privater Schlüssel zu *C(Sub-CA)*	Verschlüsselung für GW-Administrator	m
$C_{Sign}(GWA)$	Privater Schlüssel zu *C(Sub-CA)*	Signatur von GW-Administrator bzw. externe Authentisierung gegenüber Sicherheitsmodul	m

Tabelle 4: Zertifikate des GWAs

Besitzt der GWA auch gleichzeitig die Rolle eines EMT, so sind für diese verschiedenen Rollen getrennte Zertifikate gemäß Kp. 2.3.3 notwendig.

2.3.5 Zertifikate eines Smart Meter Gateways

Bei den Zertifikaten eines Smart-Meter-Gateways wird abhängig von der Betriebsphase zwischen Wirk- und Gütesiegel-Zertifikaten unterschieden. In den folgenden beiden Abschnitten werden die

Bedeutung und die Funktion der Zertifikate dargestellt. Die Zertifikate des SMGWs werden zusammen mit den zugehörigen privaten Schlüsseln auf dem Sicherheitsmodul des SMGWs gespeichert und bescheinigen die Identität des SMGWs.

2.3.5.1 Gütesiegel-Zertifikate

Die Gütesiegel-Zertifikate werden im Herstellungsprozess des SMGWs zusammen mit den entsprechenden Schlüsselpaaren auf dem Sicherheitsmodul gespeichert (vgl. Lifecycle-Beschreibung in [3]). Mit Hilfe der Gütesiegel-Zertifikate authentisiert sich ein SMGW bei Inbetriebnahme gegenüber dem GWA als echtes SMGW. Gütesiegel sind Endnutzerzertifikate und haben entsprechend die gleiche Laufzeit wie Wirk-Zertifikate. Sollte sich das SMGW bei der Inbetriebnahme mit abgelaufen Zertifikaten authentisieren, muss die Echtheit des SMGW auf separatem Wege geprüft werden. Entsprechende Prozesse sind Teil der Root-CP. In der folgenden Tabelle werden die Gütesiegel-Zertifikate beschrieben.

Zertifikat	***Signatur des Zertifikats***	***Funktion***	***m/c/o***
$C_{TLS}(SMGW)$	Privater Schlüssel zu *C(Sub-CA)*	TLS-Zertifikat für den ersten Verbindungsaufbau mit dem GWA, zur Beantragung von Wirk-Zertifikaten.	m
$C_{Enc}(SMGW)$[1]	Privater Schlüssel zu *C(Sub-CA)*	Verschlüsselung von Inhaltsdaten bei Inbetriebnahme.	m
$C_{Sign}(SMGW)$	Privater Schlüssel zu *C(Sub-CA)*	Signatur der Erstanträge für Wirk-Zertifikate nach Inbetriebnahme	m

Tabelle 5: Gütesiegel-Zertifikate des SMGWs

2.3.5.2 Wirk-Zertifikate

Bei Übergang in den Wirkbetrieb werden die Gütesiegel-Zertifikate im SMGW durch Wirk-Zertifikate ersetzt (vgl. Kp. 3.4 bzw [7]). Nach Erhalt der Wirk-Zertifikate, besitzt das SMGW seine Identität für den Wirkbetrieb. In der Tabelle werden die entsprechenden Zertifikate und deren Funktion beschrieben.

Zertifikat	***Signatur des Zertifikats***	***Funktion***	***m/c/o***
$C_{TLS}(SMGW)$	Privater Schlüssel zu *C(Sub-CA)*	TLS-Zertifikat des SMGW	m
$C_{Enc}(SMGW)$	Privater Schlüssel zu *C(Sub-CA)*	Verschlüsselung für SMGWs	m
$C_{Sign}(SMGW)$	Privater Schlüssel zu *C(Sub-CA)*	Signatur vom SMGW	m

Tabelle 6: Wirk-Zertifikate des SMGWs

1 Mit dem vollständigen Gütesiegel-Zertifikats-Tripel sollen auch Schnittstellentests durchgeführt werden können.

2.4 Zertifikatsablauf

Die Zertifikate der SM-PKI haben eine begrenzte Gültigkeitszeit (siehe Abschnitt 3.2). Die folgende Abbildung 3 zeigt die Zertifikate der SM-PKI und wie die Zertifikatswechsel auf den unterschiedlichen PKI-Ebenen erfolgen müssen.

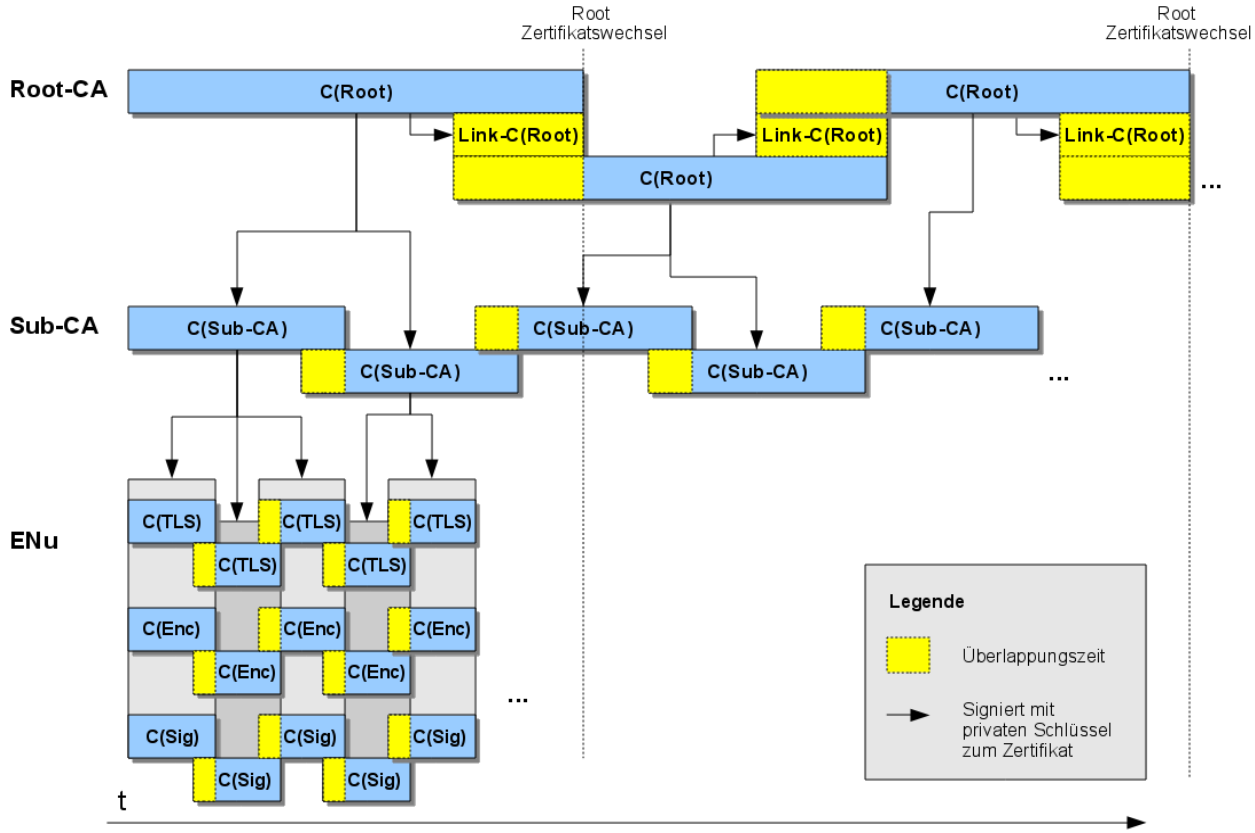


Abbildung 3: Prozess der Zertifikatsausgabe und -erneuerung

Beim Aufsetzen der SM-PKI generiert die Root-CA ein Schlüsselpaar und erzeugt ein selbst-signiertes Zertifikat. Anschließend werden die Zertifikate für die Sub-CAs ausgestellt. Die Sub-CAs stellen wiederum die Zertifikate für die Endnutzer aus.

Auf allen PKI-Ebenen muss rechtzeitig vor Ablauf eines Zertifikats bzw. der Verwendungszeit des zugehörigen privaten Schlüssels ein neues Zertifikat erzeugt werden. Entsprechende Fristen und Vorlaufzeiten werden in der Root-CP [7] festgelegt. Dieses neue Zertifikat ist dann zeitweise parallel zu dem alten Zertifikat gültig. Dieser Zeitraum wird als Überlappungszeit bzw. Übergangszeit bezeichnet. In der Überlappungszeit muss weiterhin das alte Zertifikat verwendet werden können. Mit Ablauf des alten Zertifkats endet die Überlappungszeit und es darf ausschließlich das neue Zertifikat verwendet werden.

Eine Besonderheit beim Zertifikatswechsel bildet die Root-Ebene. Vor Ablauf des Root-Zertifikats wird ein neues Schlüsselpaar erzeugt, für dessen öffentlichen Schlüssel immer zwei neue Zertifikate erzeugt werden, ein selbst-signiertes Zertifikat und ein Link-Zertifikat.

Das Link-Zertifikat dient dazu, den PKI-Teilnehmern den Wechsel vom alten Vertrauensanker zum neuen Vertrauensanker der SM-PKI zu ermöglichen. Es ist mit dem privaten Schlüssel des alten Root-Zertifikats signiert und dient zur Kettenprüfung von Zertifikaten bis zum alten Root-Zertifikat. Entsprechend kann mit diesem Zertifikat in den Übergangsphasen die Echtheit eines beliebigen Zertifikats, aus der SM-PKI, bis zum ersten Root Zertifikat geprüft werden (vgl. auch Kp. 3.3).

Nach erfolgreicher Validierung des Link-Zertifkats mit dem alten Root-Zertifikat kann das neue selbst-signierte Root-Zertifikat als neuer Vertrauensanker der SM-PKI zur Prüfung von Zertifikaten verwendet werden. Ferner ersetzt das neue Root-Zertifikat das alte Root-Zertifikat mit Ablauf von dessen Gültigkeitszeit vollständig.

2.5 Übersicht der PKI-Teilnehmer und Aufgabenstellungen

Folgende Aufgabenstellungen sind in der SM-PKI gegeben.

- Zertifizierungsstelle

  Stellt Zertifikate für sich oder eine untergeordnete Instanz aus.
- Registrierungsstelle

  Führt die Identifizierung und Authentisierung einer untergeordneten Instanz durch.
- Zertifikatsnehmer

  Bezieht Zertifikate von sich (selbstsigniert) oder von einer übergeordneten Stelle.
- Zertifikatsnutzer

  Verwendet Zertifikate für deren Aufgabenstellung. Hierbei wird zwischen der Nutzung zum Ausstellen von Zertifikaten und der Absicherung der Kommunikation unterschieden.

In der folgenden Tabelle ist eine Übersicht der PKI-Teilnehmer und deren Aufgaben dargestellt.

Instanz der PKI	***Zert.-Stelle***	***Reg.-Stelle***	***Zert.-Nehmer***	***Zert.-Nutzer***	
				Ausstellen	***Kommunikation***
Root-CA	■	■	■	■	
Sub-CA	■	■	■	■	
Endnutzer			■		■

Tabelle 7: Aufgaben der Instanzen in der SM-PKI

2.5.1 Zertifizierungsstellen

In der folgenden Tabelle sind die Zertifizierungsstellen (Certification Authority, CA) und die hiervon ausgegebenen Zertifikate aufgeführt.

PKI-Instanz	*Auszustellende Zertifikate*
Root-CA	*C(Root)*, *Link-C(Root)*, $C_{CRL\text{-}S}$*(Root)*, *C(Sub-CA)*
Sub-CA	C_{TLS}*(ENu)*, C_{Enc}*(ENu)*, C_{Sign}*(ENu)*

Tabelle 8: Zertifizierungsstellen

2.5.2 Registrierungsstellen

Registrierungsstellen (Registration Authority, RA) führen vor der Ausstellung eines Zertifikats die Identifizierung und Authentifizierung des Antragstellers durch. Die zur Identifizierung und Authentifizierung erforderlichen Prozesse werden in der Certificate Policy der Root-CA festgelegt.

- Die Root-CA betreibt eine RA zur Identifizierung und Authentifizierung der antragstellenden Sub-CAs.
- Jede Sub-CA muss über eine RA verfügen, um eine Identifizierung und Authentifizierung der Antragssteller von Kommunikationszertifikaten durchzuführen.

2.6 Abgrenzung der SM-PKI

Der Betrieb des SMGW erfordert - insbesondere durch die unterschiedlichen Gateway-Schnittstellen - Zertifikate für verschiedene Einsatzbereiche. Zweck der hier spezifizierten SM-PKI ist im Wesentlichen die Zertifikate für die Kommunikation mit dem SMGW über die WAN-Schnittstelle zur Verfügung zu stellen.

Nachfolgend werden die unterschiedlichen Einsatzbereiche von Zertifikaten im Umfeld des Smart Meter Gateways erläutert. In der folgenden Abbildung 4 ist dargestellt, wo die verschiedenen Zertifikate (farblich markiert) eingesetzt werden.

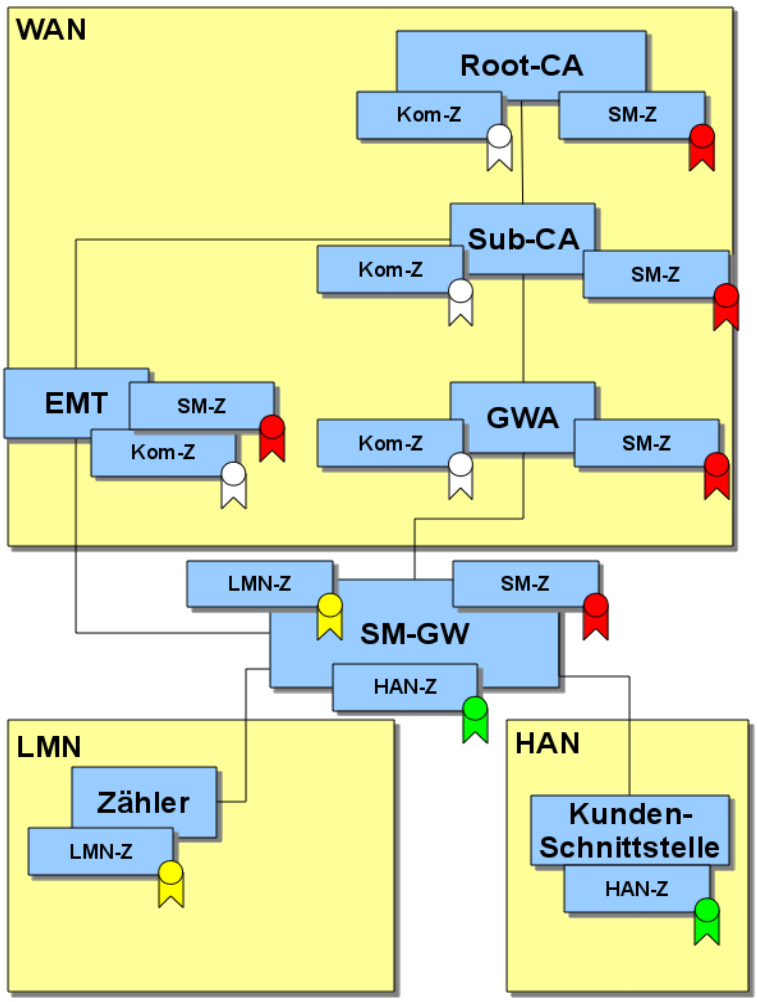


Abbildung 4: Die Zertifikatstypen der SMGW Infrastruktur

2.6.1 Smart Metering-Zertifikate

Die **Smart Metering-Zertifikate (SM-Z, rot markiert)** werden zur Kommunikation mit dem SMGW über die WAN-Schnittstelle verwendet. Diese Zertifikate sind Gegenstand dieser Spezifikation.

2.6.2 Kommunikationszertifikate

Kommunikationszertifikate (Kom-Z, weiß markiert) sind typische Zertifikate zur Client/Server-TLS-Authentisierung. Diese werden innerhalb der SM-PKI Infrastruktur verwendet, um die Kommunikationsverbindung zwischen den PKI-Teilnehmern abzusichern (siehe auch Abbildung 6). Das Sicherheitsniveau dieser Zertifikate muss vergleichbar mit dem Sicherheitsniveau der Verwaltungs-PKI (V-PKI, siehe auch [8]) sein. Alle PKIn, die über ein entsprechendes Sicherheitsniveau verfügen, können zum Ausstellen entsprechender Zertifikate verwendet werden. Generell wird empfohlen, dass die Sub-CAs für sich und ihre Clienten entsprechende Kommunikationszertifikate bereitstellen.

Die weiteren Anforderungen an die Kom-Zertifikate werden in der Root-CP festgelegt.

2.6.3 LMN-Zertifikate (informativ)

LMN-Zertifikate (LMN-Z, gelb markiert) können optional zur gegenseitigen Authentisierung von Zählern und SMGW im Local Metrological Network (LMN) verwendet werden. Die Zertifikate sind selbst-signiert und stammen nicht aus der SM-PKI.

Das Zertifikatsprofil und die weiteren Anforderungen an die LMN-Zertifikate werden in [2] festgelegt. Die zugehörigen kryptografischen Anforderungen werden in Abschnitt 6 in [6] definiert.

2.6.4 HAN-Zertifikate (informativ)

Die **HAN-Zertifikate (HAN-Z, grün markiert)** können optional zur Authentisierung von Geräten (z.B. Kunden-Schnittstelle) an der HAN-Schnittstelle des SMGW verwendet werden.

Die weiteren Anforderungen an die HAN-Zertifikate werden in [2] festgelegt.

3 Zertifikate und ihr Management

In diesem Kapitel wird ein Überblick über die Struktur der Zertifikate, die Infrastruktur und die Prozesse zum Zertifikats-, und Aktualisierungsmanagement gegeben. Hierzu werden insbesondere die zeitlichen Anforderungen definiert.

3.1 Struktur der Zertifikate

Die Zertifikate der SM-PKI sind X.509-Zertifikate gemäß den Vorgaben aus [9].

- Das Zertifikatsprofil wird in Anhang A spezifiziert.
- Der Signaturalgorithmus, die Domain-Parameter und Schlüssellängen, die in den Zertifikaten aktuell zu verwenden sind, werden von [6] vorgegeben.
- Innerhalb einer Zertifikatskette können verschiedene Signaturalgorithmen, Domain-Parameter oder Schlüssellängen verwendet werden. Folgezertifikate (insbesondere Link-Root-Zertifikate) können zu anderen Algorithmen, Parametern und Schlüssellängen wechseln. Hier sind stets die aktuellen Vorgaben aus [6] zu beachten.

3.2 Schlüsselpaare und Zertifikatslaufzeiten

Die Zertifikate der SM-PKI bescheinigen die Identität des jeweiligen Zertifikatsinhabers sowie die Authentizität des im Zertifikat enthaltenen öffentlichen Schlüssels.

Das zum Zertifikat gehörende Schlüsselpaar muss unmittelbar vor der Beantragung des Zertifikats neu erzeugt werden. Einem Schlüsselpaar einer Sub-CA bzw. eines Endnutzers darf nur ein einziges Zertifikat zugeordnet sein. Einem Schlüsselpaar der Root-CA darf nur ein einziges Paar von Zertifikaten, bestehend aus selbst-signiertem Root-Zertifikat und zugehörigem Link-Zertifikat, zugeordnet sein. Anforderungen an die Erzeugung, Speicherung und Verwendung von Schlüsseln sind in. Kp.5 sowie in [6] zu finden.

Jedes in der SM-PKI verwendete Zertifikat besitzt zudem eine Gültigkeitszeit und eine Verwendungszeit für den zugehörigen privaten Schlüssel, welche im Zertifikat angegeben werden.

Die folgende Tabelle 9 gibt die Zertifikatslaufzeiten sowie die maximalen Verwendungszeiten der zugehörigen privaten Schlüssel verbindlich vor.

Zertifikat	*Gültigkeitszeit*	*Private Key Usage Period*
Root-Zertifikat	8 Jahre	4 Jahre
Root-Link-Zertifikat	Bis zum Ablauf des alten[2] Root-Zertifikats	4 Jahre
Root-CRL-Signer-Zertifikat	4Jahre	2 Jahre
Sub-CA-Zertifikat	4 Jahre	2 Jahre
Endnutzerzertifikat	2 Jahre	2 Jahre

Tabelle 9: Zertifikatslaufzeiten

3.3 Zertifikatsvalidierung

In diesem Kapitel werden die Anforderungen an die Zertifikatsvalidierung spezifiziert.

3.3.1 Allgemeine Prozedur

Grundsätzlich muss ein Zertifikat aus der SM-PKI von einem PKI-Teilnehmer vor jeder Verwendung auf Gültigkeit geprüft werden. Die Zertifikatsvalidierung erfolgt nach dem Schalenmodell, d.h. folgende Prüfungen sind jeweils durchzuführen:

- Signaturprüfung bis zum Vertrauensanker (Cert-Path-Validation)
  - Beinhaltet die Prüfung etwaiger Link-Zertifikate
  - Beinhaltet die Sperrlistenprüfung (vgl. Kapitel 4.2.3)
- Prüfung des korrekten Gültigkeitszeitraums
- Prüfung der korrekten Verwendungsart des privaten Schlüssels (Key-Usage Validation)

Insbesondere muss die Validierung eines Link-Zertifikats der Root-CA innerhalb des Gültigkeitszeitraums des zu ersetzenden alten Root-Zertifikats und des entsprechenden Link-Zertifikats erfolgen. Erst nach erfolgreicher Validierung des Link-Zertifkats darf das zugehörige neue selbst-signierte Root-Zertifikat als Vertrauensanker für Zertifikatsvalidierungen verwendet werden.

3.3.2 Spezialfall SMGW

Die Zertifikatsvalidierung des SMGWs basiert auf dem Einbringen des Vertrauensankers bei der Herstellung.

3.3.2.1 Einbringen des Vertrauens

Bei der Herstellung des SMGWs wird auf dem Sicherheitsmodul des SMGWs das Root-Zertifikat der SM-PKI als Vertrauensanker sicher gespeichert [3]. Mit diesem Vertrauensanker kann das SMGW eine Kettenprüfung von Zertifikaten (ohne Sperrlistenprüfung) durchführen. Da grundsätz-

2 Hierunter ist das Root-Zertifikat zu verstehen, mit dem das Link-Zertifikat verfiziert werden kann, d.h. mit dessen zugehörigem privatem Schlüssel das Link-Zertifikat signiert ist.

lich mehrere Root-Zertifikate parallel gültig sein können (siehe Abbildung 3), muss immer das älteste noch gültige Root-Zertifikat gespeichert werden.

Die Konfiguration des SMGWs auf einen GWA (beinhaltet das Einbringen des GWA-Zertifikats in das SMGW) ist in [2] spezifiziert.

3.3.2.2 Wirkbetrieb

Die Konzeption des SMGWs ist dahin gehend ausgerichtet, dass möglichst wenig Kommunikationsaufkommen erzeugt wird und dass die sicherheitskritischen Prozesse im SMGW auf das zwingend notwendige reduziert sind. Daraus resultiert, dass die Zertifikatsvalidierung gemäß 3.3.1 im Wirkbetrieb stellvertretend vom GWA für das SMGW übernommen wird. Allerdings führt das SMGW bei der Installation von neuen Zertifikaten mit diesen eine Zertifikatsvalidierung (ohne Sperrlistenprüfung) durch. Die Prüfung erfolgt über das im Sicherheitsmodul gespeicherte Root-Zertifikat [3]. Entsprechend muss bei der Installation von Zertifikaten die gesamte Zertifikatskette bis zum aktuellen, auf dem Gateway vorhandenen Vertrauensanker, inklusive möglicher Link-Zertifikate, gesendet werden.

3.4 Zertifikatsmanagement

Da die in der SM-PKI eingesetzten Zertifikate die Identität eines Zertifikatsnehmers sowie die Authentizität seines öffentlichen Schlüssels bescheinigen, ist es essentiell, beides schon bei der Antragstellung zu prüfen. Um ein Zertifikat aus der SM-PKI erhalten zu können, muss ein PKI-Endnutzer zudem seine Rolle in der SMGW Kommunikationsinfrastruktur gegenüber der RA des zuständigen Zertifikatsausstellers nachweisen.

Die Beantragung und die anschließende Verteilung von Zertifikaten werden generell in Paketen organisiert. Entsprechend werden die benötigten Zertifikate für die verschiedenen Verwendungszwecke (TLS-, Verschlüsselungs- und Signaturzertifikat) parallel beantragt und ausgestellt (vgl. Anhang C). Ein Zertifikatspaket lässt sich immer eindeutig durch die in den Zertifikatsfeldern gespeicherten Namen in Kombination mit der Seriennummer (Subject-DN) des Antragstellers identifizieren (siehe Abschnitt A.1.3).

Die folgende Abbildung 5 zeigt beispielhaft die paketbasierte Organisation der SMGW Zertifikate beim Ausstellen des Zertifikats und der Speicherung im Verzeichnisdienst.

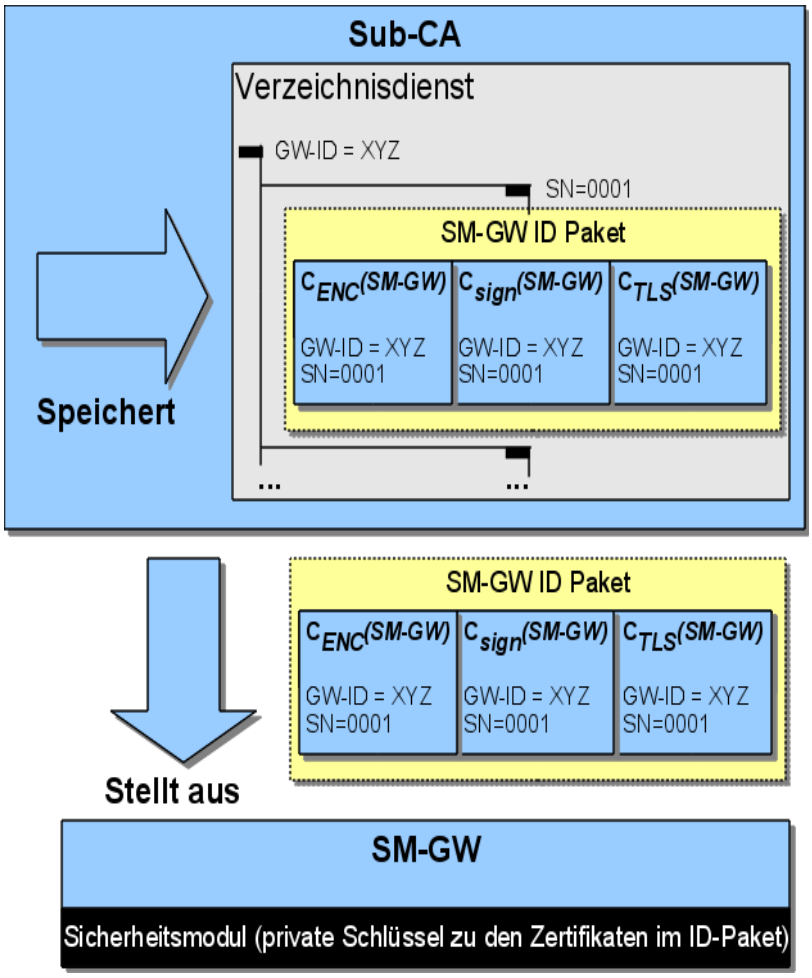


Abbildung 5: Paketbasiertes SMGW Zertifikatsmanagement (Beispiel)

Aus organisatorischen Gründen muss zwischen der initialen Beantragung von Zertifikaten und der Beantragung von Folgezertifikaten unterschieden werden.

3.4.1 Initiale Beantragung von Zertifikaten

Eine initiale Beantragung von Zertifikaten ist immer nötig, wenn der Antragsteller zum Zeitpunkt der Prüfung des Zertifikatsantrags durch die RA kein gültiges Signaturzertifikat besitzt. Die CP [7] definiert in diesem Fall geeignete Mittel zur Identifizierung und Authentifizierung des Antragstellers. Der Zertifikatsantrag muss immer über das in Kapitel C.1 spezifizierte Zertifikatsrequest-Paket erfolgen.

3.4.2 Folgezertifikate

Zur Beantragung von Folgezertifikaten muss der Antragsteller zum Zeitpunkt der Prüfung des Zertifikatsantrags durch die RA ein gültiges Signaturzertifikat besitzen. Da ein von einer CA ausgestelltes Zertifikat in der Regel vor Ablauf der Gültigkeitszeit durch ein Folgezertifikat ersetzt werden muss, ist es notwendig, dass der Prozess des Zertifikatsaustauschs durch die CA sorgfältig

geplant und in der CP [7] beschrieben wird. Dabei sind maximale Zeiten für die Zertifikatsausstellung und -Verteilung zu definieren, welche vom Antragsteller zu beachten sind, und auch Zeiten für die Inbetriebnahme der ausgestellten Zertifikate zu berücksichtigen.

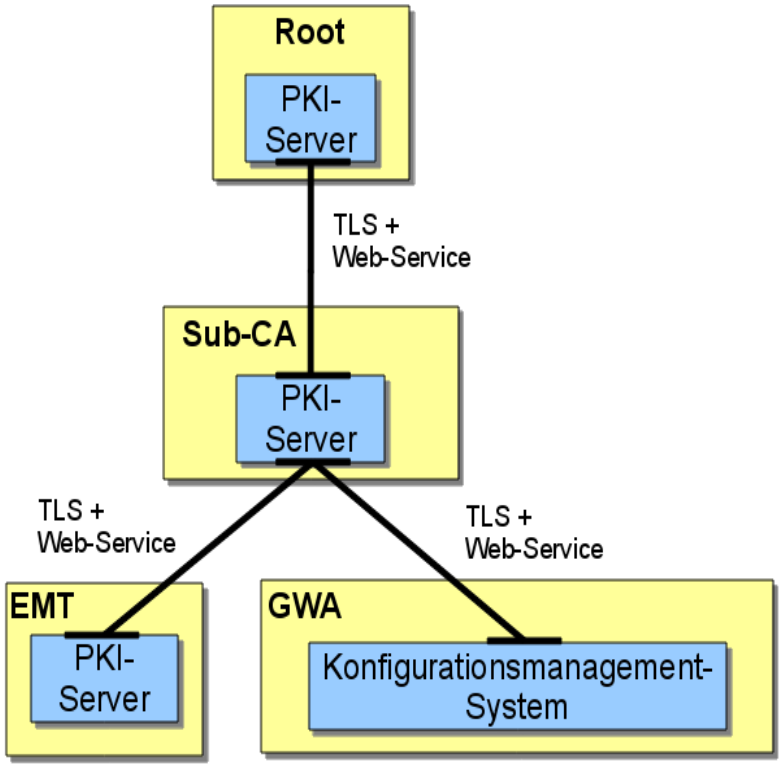
Die Identifizierung und Authentifizierung eines Zertifikatsantrags erfolgt im Falle eines Folgerequests über die Signatur des Zertifikatsrequest-Paktets (äußere Signatur, siehe Abschnitt C.2), die mit dem privaten Schlüssel zum Signaturzertifikat des Antragstellers erstellt werden muss.

Das Zertifikatsrequest-Paket beinhaltet die einzelnen Zertifikatsrequests. Jeder dieser Zertifikatsrequests ist zusätzlich mit dem privaten Schlüssel zum im Zertifikatsrequest enthaltenen öffentlichen Schlüssel signiert (innere Signatur), um den Besitz des neuen privaten Schlüssels nachzuweisen.

3.4.3 Protokolle für das Management von Zertifikaten

Zur Kommunikation der Instanzen der SM-PKI für die Beantragung und Verteilung von Zertifikaten untereinander sollen die auf Web-Services basierenden Protokolle aus Anhang D verwendet werden. Etwaige Ausnahmen hiervon können in der Root-CP [7] festgelegt werden.

In der folgenden Abbildung 6 ist dargestellt, für welche Kommunikationsverbindungen die Protokolle eingesetzt werden sollen.



Die Kommunikationsverbindungen erfordern eine beidseitige TLS-gesicherte Verbindung, um die Authentizität und Vertraulichkeit der Kommunikation sicherzustellen. Hierzu werden Kom-Zertifikate eingesetzt (siehe Abschnitt 2.6.2).

Bemerkung (informativ): Ein SMGW kann selbst keine Zertifikate bei einer Sub-CA beantragen. Dies wird stellvertretend vom GWA übernommen. Für die Erneuerung der Zertifikate eines SMGWs wird das Zertifikatsrequest-Paket vom Gateway selbst erstellt. Der Request wird vom GWA abgerufen und per Web Service durch den GWA an die zuständige Sub-CA weitergeleitet. Die ausgestellten Zertifikate müssen anschließend vom GWA gemäß den Vorgaben von Kp. 3.3 geprüft werden. Ist die Prüfung erfolgreich, so werden die Zertifikate von ihm auf dem SMGW installiert (vgl. [4]). Das SMGW darf die privaten Schlüssel dann zur Kommunikation mit EMTs und dem GWA verwenden.

3.5 Verzeichnisdienste

Die Root-CA und die Sub-CAs müssen einen auf dem LDAP-Protokoll [10] basierenden Verzeichnisdienst betreiben. In dem Verzeichnisdienst muss die jeweilige CA alle Zertifikate veröffentlichen, die von ihr ausgestellt worden sind (siehe Tabelle 10).

Der Zugriff auf den Verzeichnisdienst muss auf Organisationen aus der SMGW Infrastruktur beschränkt werden. Dies wird über eine zertifikatsbasierte TLS-Authentisierung am Verzeichnisdienst sichergestellt (siehe „Kommunikationszertifikate“ in Abschnitt 2.6.2).

Die folgende Abbildung zeigt die Verzeichnisdienste in der SM-PKI. Sperrlisten müssen öffentlich zugänglich sein, daher erfolgt deren Veröffentlichung unabhängig von den zugriffsgeschützten Verzeichnisdiensten (siehe Abschnitt 4.2.1.2).

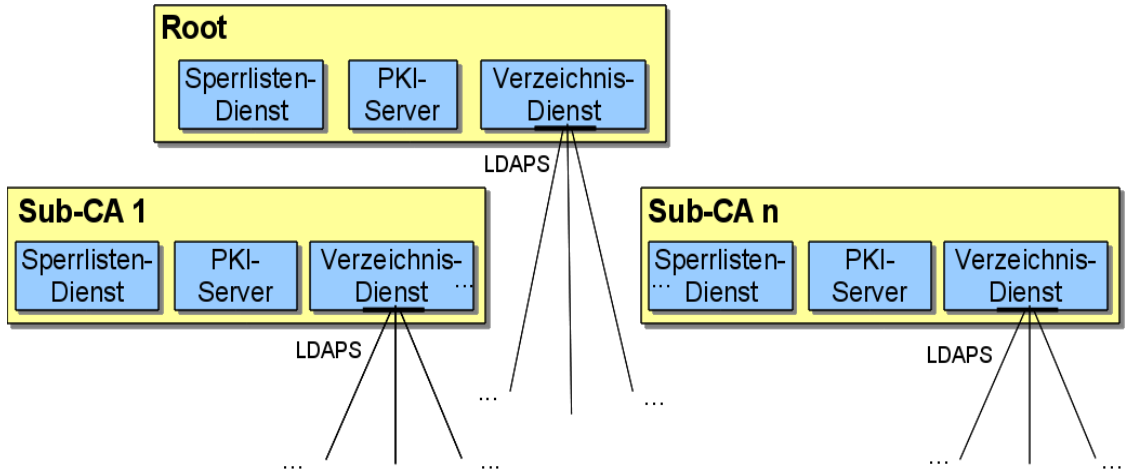


Abbildung 7: Verzeichnisdienste in der SM-PKI

In der folgenden Tabelle ist dargestellt, welche Daten die Verzeichnisdienste der PKI-Teilnehmer beinhalten müssen und wie auf diese zugegriffen werden kann.

Betreiber	*Inhalt*	*Protokoll / Adresse*
Root-CA	*C(Root)*, *Link-C(Root)*, $C_{CRL\text{-}S}$*(Root)*, *C(Sub-CA)*	- LDAPS-Protokoll / Port 636 - TLS-Authentisierung mit Kom-Zertifikaten - URL in Root-CP definiert
Sub-CA	*C(Sub-CA)*, C_{TLS}*(ENu)*, C_{Enc}*(ENu)*, C_{Sign}*(ENu)*	- LDAPS-Protokoll / Port 636 - TLS-Authentisierung mit Kom-Zertifikaten -URL in Sub-CA-CP definiert

Tabelle 10: Verzeichnisdienste in der SM-PKI

Der Verzeichnisdienst der CAs soll die in Anhang E spezifizierte Struktur besitzen.

Die Verzeichnisse werden im Sinne einer Bestandsführung und eines Archivs geführt und sind kontinuierlich zu pflegen.

3.5.1 Spezialfall SMGW

Die Verantwortung für das Management der auf den SMGW installierten Zertifikate von autorisierten Marktteilnehmern und dem aktuellen Root-Zertifikat obliegt dem GWA.

Der GWA muss daher stets auf jedem seiner SMGWs die Liste der installierten Zertifikate aktuell halten. Befinden sich auf einem SMGW in naher Zukunft ablaufende Zertifikate eines autorisierten Marktteilnehmers bzw. des GWA, so muss er auf dem SMGW entsprechende neue Zertifikate installieren. Entsprechende Update-Fristen werden in der Root-CP [7] festgelegt. Die entsprechenden Zertifikate bezieht der GWA über den Verzeichnisdienst der zustädigen CA.

In der Übergangsphase, in der noch beide Zertifikate eines autorisierten Marktteilnehmers für die Verwendungszwecke TLS bzw. Signatur gültig sind, muss das SMGW beide Zertifikate verwenden können. Nach Ablauf eines Zertifikats darf das SMGW diese nicht mehr verwenden.

Bemerkung (informativ): Der GWA verfügt dazu über ein Konfigurationsmanagement, das ein Verzeichnis mit allen ihm verwalteten SMGWs und die darauf installierten Zertifikate enthält. Dieses Verzeichnis muss vom GWA dann regelmäßig auf ablaufende bzw. bald ablaufende Zertifikate geprüft werden

4 Sperrlisten und Sperrdienst

Die in der SM-PKI ausgestellten Zertifikate können zurückgerufen werden. In diesem Kapitel wird ein Überblick über die Sperrlisten und den Sperrdienst gegeben.

4.1 Struktur der Sperrlisten

Die Sperrlisten der SM-PKI sind X.509-Sperrlisten gemäß den Vorgaben aus [9]. Von dieser Spezifikation werden vollständige CRLs und Delta-CRLs unterstützt.

- Das Profil für die Sperrlisten wird in Anhang B spezifiziert.
- Die Algorithmen, Domain-Parameter und Schlüssellängen, die zur Signatur der Sperrlisten zu verwenden sind, werden von [6] vorgegeben.

4.2 Sperrmanagement

In diesem Abschnitt wird die Infrastruktur des Sperrmanagements spezifiziert. Die Gründe für eine Sperrung und die organisatorischen Abläufe werden in der Root-CP [7] definiert.

4.2.1 Veröffentlichung von Sperrlisten

Die in der folgenden Tabelle 11 dargestellten Sperrlisten sind in der SM-PKI vorhanden.

Sperrliste	*Herausgeber*	*Inhalt*	*Signatur-Schlüssel*
Root-CRL	Root-CA	Gesperrte Root- und Sub-CA-Zertifikate.	Privater Schlüssel zum aktuellem $C_{CRL\text{-}S}$*(Root).*
Sub-CA-CRL	Jede Sub-CA erstellt eine eigene CRL.	Gesperrte Endnutzer-Zertifikate (im Verwaltungsbereich der jeweiligen Sub-CA).	Privater Schlüssel zum aktuellen *C(Sub-CA).*

Tabelle 11: Die Sperrlisten in der SM-PKI

Die Root-CA und die Sub-CAs müssen deren Sperrlisten wahlweise über HTTP [11] oder LDAP [10] zum freien Herunterladen bereitstellen (siehe Abbildung 8). Hierbei müssen die in Tabelle 12 definierten Zeiten eingehalten werden. Damit die Authentizität der jeweiligen Sperrliste überprüft werden kann, müssen die hiefür erforderlichen Zertifikate zusätzlich zur Sperrliste frei abrufbar sein.

Auf die Sperrlisten muss frei zugegriffen werden können. Die Bezugsadresse der Sperrlisten muss in den Zertifikaten (siehe Anhang A) und in der CP des jeweiligen PKI-Teilnehmers angegeben werden.

In der folgenden Abbildung 8 ist die Infrastruktur des Sperrmanagements im Zusammenspiel mit dem Verzeichnisdienst dargestellt.

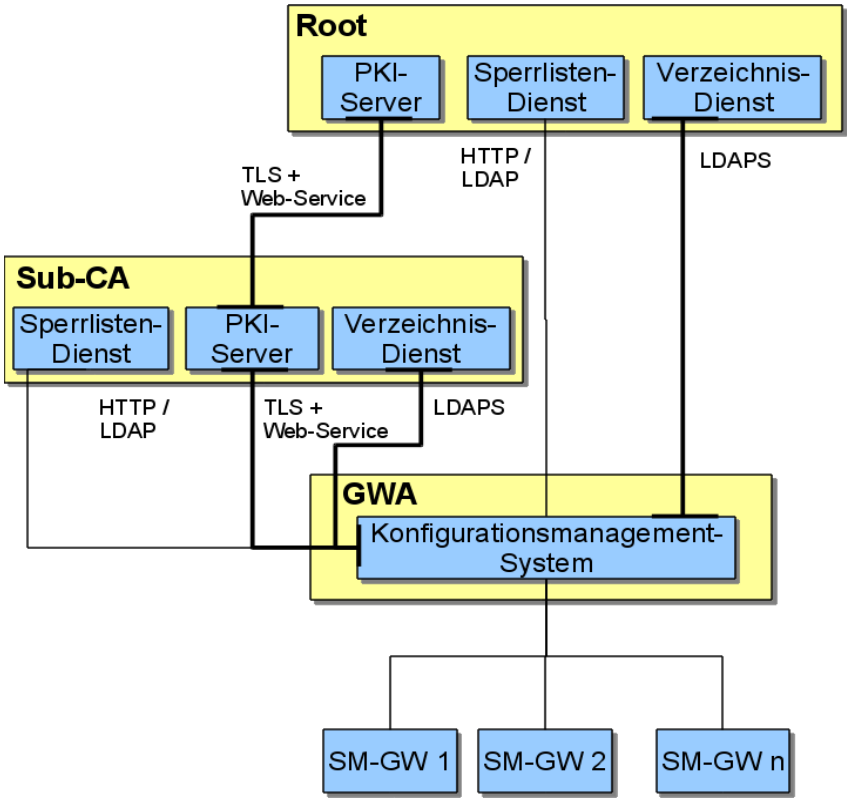


Abbildung 8: Infrastruktur Sperrmanagement (Beispiel)

4.2.1.1 Sperrlisten der Root-CA

Die Root muss einen Sperrlisten-Dienst betreiben. Dabei werden von der Root-CA nur direkte CRLs ausgegeben, d.h. die Sperrliste enthält nur gesperrte Zertifikate, die von der Root-CA selbst ausgegeben wurden. Hierüber kann die Gültigkeit aller von der Root-CA ausgegeben Zertifikate überprüft werden.

Die Root-CA muss in den in Kapitel 4.2.2 angegebenen Fristen stets eine aktuelle vollständige CRL zum Abruf bereitstellen. Optional kann zusätzlich eine aktuelle Delta-CRL veröffentlicht werden. Abgelaufene Zertifikate müssen von der Sperrliste wieder entfernt werden.

Die Root-CA verwendet ein separates Schlüsselpaar (Root-CRL-Signer) zur Signatur der Root-CRLs. Das Zertifikat zum Root-CRL-Signer wird mit dem privaten Schlüssel zum aktuell gültigen Root-Zertifikat signiert. Mit einem Wechsel des Root-Schlüssels wird auch ein neuer Root-CRL-Signer erzeugt.

4.2.1.2 Sperrlisten der Sub-CA

Die Sub-CA muss einen Sperrlisten-Dienst betreiben. Dabei werden von der Sub-CA nur direkte CRLs ausgegeben, d.h. die Sperrliste enthält nur gesperrte Zertifikate, die von der Sub-CA selbst ausgegeben wurden. Hierüber kann die Gültigkeit aller von der jeweiligen Sub-CA ausgegeben Zertifikate überprüft werden.

Eine Sub-CA muss in den in Kapitel 4.2.2 angegebenen Fristen stets eine aktuelle vollständige CRL zum Abruf bereitstellen. Optional kann zusätzlich eine aktuelle Delta-CRL veröffentlicht werden. Abgelaufene Zertifikate müssen von der Sperrliste wieder entfernt werden.

Die Sub-CA verwendet den privaten Schlüssel zum aktuell gültigen Sub-CA-Zertifikat zur Signatur ihrer CRL.

4.2.2 Aktualisierungs- und Prüfungszeiten im Sperrmanagement

In der folgenden Tabelle sind die minimal erforderlichen Aktualisierungs- und Prüfungszeiten der Sperrlisten für die einzelnen PKI-Teilnehmer definiert. Es wird zwischen regelmäßigen Aktualisierungen, verursacht durch den Ablauf der Gültigkeitszeit einer Sperrliste, und anlassbezogenen Aktualisierungen, verursacht durch die Sperrung von Zertifikaten, unterschieden.

Sollte eine Sperrliste nicht verfügbar bzw. abrufbar sein, muss ersatzweise mit der zuletzt bekannten Sperrliste weitergeprüft werden. Die für die Sperrliste verantwortliche CA muss hierüber unverzüglich informiert werden (über Kontaktadresse in der CP der CA). Diese muss dann auf anderem Wege eine aktuelle Sperrliste zur Verfügung stellen. Steht nach 3 Tagen immer noch keine aktualisierte Sperrliste zur Verfügung, muss die Root-CA informiert werden. Den Zertifikaten der entsprechenden CA kann in diesem Fall nicht vertraut werden.

PKI-Teilnehmer	***Regelmäßge Aktualisierung der Sperrliste***	***Anlaßbezogene Aktualisierung der Sperrlist***	***Abruf der Sperrliste***	***Prüfung der Zertifikate auf Sperrung***
Root-CA	Innerhalb von 30 Tagen	Innerhalb von 24 Stunden	Entfällt (Keine übergeordnete Sperrliste)	Entfällt (Keine übergeordnete Sperrliste)
Sub-CA	Innerhalb von 7 Tagen	Innerhalb von 24 Stunden	Täglich	Täglich
Endnutzer (außer SMGW)	Entfällt (Erstellt keine Sperrliste)	Entfällt (Erstellt keine Sperrliste)	Täglich	Bei jeder Verwendung
Endnutzer SMGW	Entfällt (Erstellt keine Sperrliste)	Entfällt (Erstellt keine Sperrliste)	Täglich durch GWA bzw. anlassbezogen (vgl. 4.2.3.2)	Täglich durch GWA bzw. anlassbezogen (vgl. 4.2.3.2)

Tabelle 12: Zeitliche Anforderungen des Sperrmanagements

4.2.3 Sperrlistenvalidierung

4.2.3.1 Generelle Prozedur

Grundsätzlich muss eine Sperrliste aus der SM-PKI von einem PKI-Teilnehmer geprüft werden, bevor diese zur Zertifikatsvalidierung (siehe Abschnitt 3.3) verwendet werden darf. Bei der Sperrlistenvalidierung sind jeweils die folgenden Prüfungen durchzuführen:

- Signaturprüfung bis zum Vertrauensanker (Cert-Path-Validation)
  - Beinhaltet die Prüfung etwaiger Link-Zertifikate
  - Bei Signaturprüfung einer Sub-CA CRL muss C(Sub-CA) auch gegen die Root-CRL geprüft werden.
- Prüfung der Aktualität der Sperrliste.

4.2.3.2 Sonderfall SMGW

Für das SMGW ist die Möglichkeit einer eigenständigen Sperrlistenprüfung nicht gegeben. Diese Aufgabe muss daher vom Gateway-Administrator (GWA) übernommen werden (vgl. [4]). Die Prüfung der relevanten Sperrlisten erfolgt in diesem Falle in regelmäßigen Abständen gemäß den Vorgaben von Tabelle 12 bzw. aus konkretem Anlass[3]. Die Validierung durch den GWA muss dabei erfolgen, wie in Kapitel 4.2.3.1 beschrieben. Befindet sich in der jeweiligen Sperrliste ein Zertifikat eines EMTs, das auf dem SMGW installiert ist, so muss dieses unverzüglich durch den GWA vom SMGW gelöscht werden bzw. durch ein neues gültiges Zertifikat des entsprechenden EMTs ersetzt werden (vgl Kapitel 3.5).

Bemerkung (informativ): Der GWA verfügt über ein Konfigurationsmanagement, in dem ein Verzeichnis alle verwalteten SMGWs und die darin installierten Zertifikate registriert sind. Bei der Prüfung der relevanten Sperrlisten via der oben beschriebenen Prozedur, gleicht der GWA die Zertifikate in seinem Verzeichnis mit den Sub-CA Sperrlisten ab und löscht die gesperrten Zertifikate auf seinen SMGWs bzw. ersetzt diese durch neue gültige Zertifikate des entsprechenden EMTs (vgl. [4]).

3 Die konkreten Anlässe, die eine außerplanmäßige Sperrlistenprüfung durch den GWA notwendig machen, sind in der Root-CP definiert.

5 Sicherheitsanforderungen an die PKI-Teilnehmer

Das Sicherheitsniveau der SM-PKI wird durch die CP der Root-CA [7] definiert. In der Root-CP sind alle organisatorischen, physikalischen und technischen Sicherheitsanforderungen für die verschiedenen PKI-Teilnehmer enthalten.

Diese TR definiert nur die Sicherheitsanforderungen an die Kryptografiemodule zum Schutz der privaten Schlüssel zu den Zertifikaten der SM-PKI. Des Weiteren werden die grundsätzlichen Anforderungen an die IT-Sicherheit der Zertifizierungsstellen(CAs) definiert.

5.1 Anforderungen an kryptographische Module und die Sicherung privater Schlüssel

Folgende Systeme der SM-PKI müssen kryptographische Hardwaremodule, sogenannte Kryptographiemodule (KM), zur Generierung, Speicherung und Nutzung des jeweiligen privaten Schlüssels verwenden. Innerhalb der PKI können verschiedene Typen von Kryptografiemodulen eingesetzt werden z.B. HSMs, Chipkarten und Sicherheitsmodule.

PKI-Instanz	*System*	*Nutzung*
Root-CA	CA	Zertifikatsausstellung
Sub-CAs	CA	Zertifikatsausstellung
Endnutzer	EMT	Kommunikation mit den Teilnehmern der SMGW Infrastruktur.
	GWA	Kommunikation mit den Teilnehmern der SMGW Infrastruktur und Konfiguration von SMGWs.
	SMGW	Kommunikation mit den Teilnehmern der SMGW Infrastruktur.

Tabelle 13: Übersicht Kryptographiemodule in der SM-PKI

Die Kryptographiemodule MÜSSEN je nach Typ nach den folgenden Common Criteria Protection Profiles (PPs) durch das BSI zertifiziert sein.

Root-CA/ Sub-CA / EMT /GWA:

- Hardware Security Module (HSM): „PP-Cryptographic Modules, Security Level "Moderate“ [12] oder „PP-Cryptographic Modules, Security Level "Enhanced“ [13].
- Chipkarte: „Protection Profile for Secure Signature Creation Device - Part 2: Device with Key Generation“ [14].

SMGW:

- In SMGWs MÜSSEN gemäß TR-03109 Sicherheitsmodule eingesetzt werden, die nach dem „Protection Profile for the Security Module of a Smart Metering System“ [15] zertifiziert sind.

Das aktuelle Root-Zertifikat muss bei der Herstellung des SMGWs auf dem Sicherheitsmodul gespeichert werden.

5.2 Sicherheitsanforderungen an die Zertifizierungsstellen

Als Sicherheitsanforderung müssen die Root sowie die Sub-CAs jeweils über ein Zertifikat "ISO/IEC 27001" [16] verfügen. Es wird empfohlen, die Zertifizierung auf der Basis von IT-Grundschutz [17] vorzunehmen. Die Zertifizierung muss alle Geschäftsprozesse und IT-Systeme des Registrierungs- und Zertifizierungsbetriebs des jeweiligen PKI-Teilnehmers umfassen. Hinsichtlich des Schutzbedarfs muss die Stufe „hoch“ umgesetzt sein. Sofern IT-Grundschutz verwendet wird, kann die Anforderung übergangsweise in Abstimmung mit dem BSI auch durch IT-Grundschutz Auditoren-Testate "IT-Grundschutz Einstiegsstufe" & "IT-Grundschutz Aufbaustufe" erfüllt werden.

A Zertifikatsprofile

Die Zertifikate, die in der SM-PKI ausgegeben werden, sind X.509-Zertifikate in der Version 3 und müssen konform zu dem in diesem Kapitel definierten Zertifikatsprofil sein.

A.1 Zertifikatskörper

Der Zertifikatskörper eines X.509-Zertifikats besitzt gemäß [9] die folgende Struktur:

Zertifikatsfeld	*Referenz in [9]*	*m/x/c/o*	*Wert*
`Certificate`	4.1.1		
`TBSCertificate`	4.1.1.1	m	Siehe Tabelle 15.
`SignatureAlgorithm`	4.1.1.2	m	Siehe Abschnitt A.1.1.
`SignatureValue`	4.1.1.3	m	Abhängig vom gewählten Signaturalgorithmus.

Tabelle 14: Zertifikatskörper

Die folgende Tabelle 15 gibt die Struktur des Feldes `TBSCertificate` verbindlich vor.

Zertifikatsfeld	*Referenz in [9]*	*m/x/c/o*	*Wert*
`TBSCertificate`	4.1.2	m	
`Version`	4.1.2.1	m	'`v3`'
`SerialNumber`	4.1.2.2	m	Zufällig gewählte, eindeutige Nummer bestimmt von der CA (nicht länger als 20 Octets).
`Signature`	4.1.2.3	m	Gleicher Wert wie im Feld `signatureAlgorithm`.
`Issuer`	4.1.2.4	m	Eindeutiger Name (Distinguished Name, DN) des Zertifikatsherausgebers, siehe auch Kapitel A.1.3.
`Validity`	4.1.2.5	m	Die Gültigkeitszeiten der Zertifikate sind in Kapitel 3.2 angegeben.
`Subject`	4.1.2.6	m	Eindeutiger Name (Distinguished Name, DN) des Zertifikatsinhabers, vgl. Kapitel A.1.3.
`SubjectPublicKeyInfo`	4.1.2.7	m	Siehe Abschnitt A.1.2.
`IssuerUniqueID`	4.1.2.8	x	Entfällt.

Zertifikatsfeld	*Referenz in [9]*	*m/x/c/o*	*Wert*
`SubjectUniqueID`	4.1.2.8	x	Entfällt.
`Extensions`	4.1.2.9	m	Kapitel A.2 definiert die vorhandenen Extensions.

Tabelle 15: Struktur des Feldes `TBSCertificate`

A.1.1 SignatureAlgorithm

Durch die Datenstruktur `SignatureAlgorithm` wird nach [9] der Signaturalgorithmus des Zertifikats angegeben. Dieser besteht aus der folgenden Datenstruktur:

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

Der Wert von `algorithm` wird von [6], Kapitel 3, verbindlich vorgegeben. Das Feld `parameters` bleibt leer.

A.1.2 SubjectPublicKeyInfo

Das Feld `SubjectPublicKeyInfo` muss folgende Struktur besitzen (siehe [18]) :

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

Die OID in `algorithm` muss den Wert 1.2.840.10045.2.1 (`id-ecPublicKey`) haben. Als EC-Parameter ist gemäß [18] die Variante `namedCurve` zu verwenden. In Abschnitt F sind die von dieser Technischen Richtlinie unterstützten Elliptischen Kurven aufgelistet.

Für die aktuell zu verwendenden Werte siehe [6], Kapitel 3, 4 und 7.

A.1.3 Issuer und Subject

Die nachfolgend definierten Namensschemata sollen für die verschiedenen PKI-Instanzen in der angegebenen Feld-Reihenfolge verwendet werden. Je nachdem ob es bei der PKI-Instanz um einen Zertifikatsaussteller oder -inhaber handelt, müssen die Werte im Feld `Issuer` bzw. `Subject` eingetragen werden.

Auf Endnutzer-Ebene wird zusätzlich zwischen den Rollen SMGW, GWA und EMT unterschieden. Ferner wird dort zwischen Signatur-, TLS- und Verschlüsselungszertifikat getrennt. Diese Unter-

scheidung erfolgt nicht über das Feld `Subject`, sondern über die Extension `KeyUsage` (siehe Abschnitt A.2).

1. Namensschema der Root-Instanz

Attribute Type	*Kürzel*	*m/o/x*	*Value*	*Erläuterung*
`common name`	`CN`	m	“Root-CA“	Name der Root.
`organization`	`O`	m	“SM-PKI“	Name der PKI.
`organizational unit`	`OU`	o	“<Organisationseinheit>“	Name der Organisationseinheit.
`country`	`C`	m	“DE“	Ländercode in Großbuchstaben.
`serial number`	`SERIALNUMBER`	m	“<SN>“	Sequenznummer des Zertifikats. Die Sequenznummer muss im Bereich von 0 und 2^{31}-1 liegen und startet bei 1. Bei jedem Folgezertifikat wird der Wert um 1 hochgezählt.

Tabelle 16: Namensschema Root-Instanz

2. Namensschema der Sub-CA-Instanzen

Attribute Type	*Kürzel*	*m/o/x*	*Value*	*Erläuterung*
`Common Name`	`CN`	m	“<XXX> CA“	Name der Sub-CA. Am Ende des CN muss “CA“ im Feld stehen, um die CA-Funktion in der PKI kenntlich zu machen.
`Organization`	`O`	m	“SM-PKI“	Name der PKI.
`Organizational Unit`	`OU`	o	“<Organisationseinheit>“	Name der Organisationseinheit.
`Country`	`C`	m	“DE“	Ländercode in Großbuchstaben.
`Serial Number`	`SERIALNUMBER`	m	“<SN>“	Sequenznummer des Zertifikats. Die Sequenznummer muss im Bereich von 0 und 2^{31}-1 liegen und startet bei 1. Bei jedem Folgezertifikat wird der Wert um 1 hochgezählt.

Attribute Type	*Kürzel*	*m/o/x*	*Value*	*Erläuterung*
street	street	o	„<Straße>“	Straße der Sub-CA
postal code	postal code	o	„<PLZ>“	PLZ der Sub-CA
Locality	L	o	“<Ortsname>“	Ortsname des Sub-CA-Betreiber- Standorts.
State	ST	o	“<Bundesland>“	Bundesland des Sub-CA-Betreiber- Standorts.

Tabelle 17: Namensschema Sub-CA-Instanzen

3. Namensschema für die Endnutzer GWAs

Attribute Type	*Kürzel*	*m/o/x*	*Value*	*Erläuterung*
Common Name	CN	m	“GWA <XXX>“	Name des GWA. Am Anfang muss “GWA“ im Feld stehen, um die Admin-Funktion in der Infrastruktur kenntlich zu machen.
Organization	O	m	“SM-PKI“	Name der PKI.
Organizational Unit	OU	o	“<Organisationseinheit>“	Name der Organisationseinheit.
Country	C	m	“DE“	Ländercode in Großbuchstaben.
Serial Number	SERIALNUMBER	m	“<SN>“	Sequenznummer des Zertifikats. Die Sequenznummer muss im Bereich von 0 und 2^{31}-1 liegen und startet bei 1. Bei jedem Folgezertifikat wird der Wert um 1 hochgezählt.
street	street	o	„<Straße>“	Straße des GWA
postal code	postal code	o	„<PLZ>“	PLZ des GWA
Locality	L	m	“<Ortsname>“	Ortsname des GWA-Betreiber- Standorts.
State	ST	m	“<Bundesland>“	Bundesland des GWA-Betreiber- Standorts.

Tabelle 18: Namensschema GWA

4. Namensschema für die EMT Endnutzer

Attribute Type	*Kürzel*	*m/o/x*	*Value*	*Erläuterung*
Common Name	CN	m	“<Name>“	Name der Organisation.
Organization	O	m	“SM-PKI“	Name der PKI.
Organizational Unit	OU	o	“<Organisationseinheit>“	Name der Organisationseinheit.
Country	C	m	“DE“	Ländercode in Großbuchstaben
Serial Number	SERIALNUMBER	m	“<SN>“	Sequenznummer des Zertifikats. Die Sequenznummer muss im Bereich von 0 und 2^{31}-1 liegen und startet bei 1. Bei jedem neuen Schlüssel um 1 hochgezählt.
street	street	o	„<Straße>“	Straße des EMTs
postal code	postal code	o	„<PLZ>“	PLZ des EMTs
Locality	L	m	“<Ortsname>“	Ortsname des EMT Betreiber- Standorts.
State	ST	m	“<Bundesland>“	Bundesland des -EMT Betreiber- Standorts.

Tabelle 19: Namensschema Endnutzer EMT

5. Namensschema des Endnutzer SMGWs im Wirkzertifikat

Attribute Type	*Kürzel*	*m/o/x*	*Value*	*Erläuterung*
Common Name	CN	m	“<GW-ID>“	Herstellerübergreifende Identifikationsnummer für Messeinrichtungen des SMGW gemäß [19] (Schema: Ziffern und Großbuchstaben ohne Leerzeichen).
Organization	O	m	“SM-PKI“	Name der PKI.
Organizational Unit	OU	m	“<Organisationseinheit>“	Name des zuständigen GWA = Common Name (CN) aus dem GWA-Zertifikat.
Country	C	m	“DE“	Ländercode in Großbuchstaben.

Attribute Type	*Kürzel*	*m/o/x*	*Value*	*Erläuterung*
Serial Number	SERIALNUMBER	m	"<SN>"	Sequenznummer des Zertifikats. Die Sequenznummer muss im Bereich von 0 und 2^{31}-1 liegen und startet bei 1. Bei jedem Folgezertifikat wird der Wert um 1 hochgezählt.
street	street	x	„<Straße>"	Straße des Standorts
postal code	postal code	x	„<PLZ>"	PLZ des Standorts
Locality	L	x	"<Ortsname>"	Ortsname des Standorts.
State	ST	x	"<Bundesland>"	Bundesland des Standorts.

Tabelle 20: Namensschema SMGW

6. Sonderfall: Namensschema des Endnutzers SMGW in Gütesiegel-Zertifikaten

   Diese Zertifikate sind bei Inbetriebnahme des SMGW auf dem integrierten Sicherheitsmodul gespeichert. Mit diesen Zertifikaten kann sich das SMGW gegenüber dem GW-Admin als authentisches SMGW ausweisen, um Wirk-Zertifikate für den Normalbetrieb zu erhalten.

Attribute Type	*Kürzel*	*m/o/x*	*Value*	*Erläuterung*
Common Name	CN	m	"<GW-ID>"	Herstellerübergreifende Identifikationsnummer für Messeinrichtung des SMGW gemäß [19] (Schema: Ziffern und Großbuchstaben ohne Leerzeichen).
Organization	O	m	"SM-PKI"	Name der PKI.
Organizational Unit	OU	o	"<Organisationseinheit>"	Name des SM-GW Herstellers.
Country	C	m	"DE"	Ländercode in Großbuchstaben
Serial Number	SERIALNUMBER	m	"0"	Sequenznummer der Zertifikate. Belegung 0 = Gütesiegelzertifikat.
street	street	o	„<Straße>"	Straße des GW-Herstellers
postal code	postal code	o	„<PLZ>"	PLZ des GW-Herstellers
Locality	L	o	"<Ortsname>"	Ortsname des SMGW

Attribute Type	*Kürzel*	*m/o/x*	*Value*	*Erläuterung*
				Herstellers.
State	ST	o	“<Bundesland>“	Bundesland des des SMGW Herstellers.

Tabelle 21: Namensschema Gütesiegelzertifikate

A.2 Extensions

Die nun folgenden Tabellen enthalten eine Übersicht der Extensions, die verwendet werden müssen. Weitere Zertifikatsextensions sind nicht erlaubt.

Nr.	*Bezeichnung*	*C(Root)*	*C_{CRL-S}(Root)*	*C(Sub-CA)*
1	AuthorityKeyIdentifier	m	m	m
2	SubjectKeyIdentifier	m	m	m
3	KeyUsage	m	m	m
4	PrivateKeyUsagePeriod	m	m	m
5	CertificatePolicies	m	m	m
6	SubjectAltNames	m	m	m
7	IssuerAltName	m	m	m
8	BasicConstraints	m	m	m
9	ExtendedKeyUsage	x	x	x
10	CRLDistributionPoints	m	m	m

Tabelle 22: Zertifikats-Extensions für CA-Zertifiakte

Nr.	*Bezeichnung*	*C(GWA)*	*C(EMT)*	*C(SMGW)*
1	AuthorityKeyIdentifier	m	m	m
2	SubjectKeyIdentifier	m	m	m
3	KeyUsage	m	m	m
4	PrivateKeyUsagePeriod	o	o	o
5	CertificatePolicies	m	m	m
6	SubjectAltNames	m	m	x
7	IssuerAltName	m	m	m
8	BasicConstraints	m	m	m
9	ExtendedKeyUsage	c[4]	c[4]	c[4]
10	CRLDistributionPoints	m	m	m

Tabelle 23: Zertifikats-Extensions für Endnutzer-Zertifikate (sortiert nach Endnutzer)

Nr.	*Bezeichnung*	C_{TLS}*(ENu)*	C_{Enc}*(ENu)*	C_{Sign}*(ENu)*
1	AuthorityKeyIdentifier	m	m	m
2	SubjectKeyIdentifier	m	m	m
3	KeyUsage	m	m	m
4	PrivateKeyUsagePeriod	o	o	o
5	CertificatePolicies	m	m	m
6	SubjectAltNames	c[5]	c[5]	c[5]
7	IssuerAltName	m	m	m
8	BasicConstraints	m	m	m
9	ExtendedKeyUsage	m	x	x
10	CRLDistributionPoints	m	m	m

Tabelle 24: Zertifikats-Extensions für Endnutzer-Zertifikate (sortiert nach Verwendungszweck)

Nachfolgend werden die verschiedenen Extensions definiert:

1. `AuthorityKeyIdentifier`
   - Extension-ID (OID): 2.5.29.35
   - Kritisch: Nein
   - Beschreibung: Der `authorityKeyIdentifier` wird benutzt, um verschiedene öffentliche Schlüssel desselben Zertifikatsherausgebers unterscheiden zu können.

4 Das Vorhandensein der Extension ist abhängig vom konkreten Verwendungszweck des Zertifikats (`Key Usage` / `ExtendedKeyUsage`), siehe Tabelle 24

5 Das Vorhandensein der Extension ist abhängig vom Zertifikatsinhaber, siehe Tabelle 23.

- Wert: Der `KeyIdentifier` wird mit Methode 1 gemäß [9], 4.2.1.1 berechnet, d.h. er besteht aus dem SHA-1-Wert des Feldes `subjectPublicKey` (ohne tag, length und number of unused bits) aus dem Zertifikat des Zertifikatsherausgebers.

2. `SubjectKeyIdentifier`
   - Extension-ID (OID): 2.5.29.14
   - Kritisch: Nein
   - Beschreibung: Der dient zur Identifikation eines Zertifikats mit einem spezifischen öffentlichen Schlüssel.
   - Wert: Der `KeyIdentifier` wird mit Methode 1 gemäß [9], 4.2.1.1 berechnet, d.h. er besteht aus dem SHA-1-Wert des Felds `subjectPublicKey` (ohne tag, length und number of unused bits) des Zertifikatsinhabers.
3. `KeyUsage`
   - Extension-ID (OID): 2.5.29.15
   - Kritisch: Ja
   - Beschreibung: Die Extension `KeyUsage` ist in [9], 4.2.1.1 spezifiziert. Diese defniert, für welche Zwecke der zertifizierte öffentliche Schlüssel verwendet werden darf.
   - Wert: Folgende Bits müssen in den einzelnen Zertifikaten gesetzt sein:

Zertifikat	***C(Root)***	***$C_{CRL\text{-}S}$(Root)***	***C(Sub-CA)***
Gesetzte Bits	KeyCertSign	cRLSign	KeyCertSign, cRLSign

Tabelle 25: Belegung KeyUsage-Extension für CA-Zertifikate

Zertifikat	***C_{TLS}(ENu)***	***C_{Enc}(ENu)***	***C_{Sign}(ENu)***
Gesetzte Bits	DigitalSignature	KeyEncipherment, KeyAgreement	DigitalSignature

Tabelle 26: Belegung KeyUsage-Extension für Endnutzer-Zertifikate

4. `PrivateKeyUsagePeriod`
   - Extension-ID (OID): 2.5.29.16
   - Kritisch: Nein
   - Beschreibung: Die Extension `PrivateKeyUsagePeriod` wird in [9], 4.2.1.1 spezifiziert. Diese definiert die Gültigkeitszeit des zum Zertifikat gehörenden privaten Schlüssels.
   - Wert: Die einzutragenden Verwendungszeiten des privaten Schlüssels werden von Kapitel 3.2 vorgegeben.
5. `CertificatePolicies`
   - Extension-ID (OID): 2.5.29.32
   - Kritisch: Nein

- Beschreibung: Diese Extension (spezifiziert in [9], 4.2.1.4) gibt Informationen über die zugrundeliegende Certificate Policy, nach der das Zertifikat ausgestellt wurde. Ferner wird angegebenen, wo die CP bezogen werden kann.
- Wert: Einzutragen ist hier jeweils die OID der CP, nach dem das Zertifikat ausgegeben wurde, im Feld `policyIdentifiers` sowie die `CPSuri` zum Bezugspunkt der zugehörigen CP im Feld `policyQualifiers`.

6. `SubjectAltNames`
   - Extension-ID (OID): 2.5.29.17
   - Kritisch: Nein
   - Beschreibung: Diese Extension (Definition siehe [9], 4.2.1.6) enthält weitergehende Informationen zum Inhaber (Subject) - Namen.
   - Wert: Sofern vorhanden, muss der `SubjectAltName` den `rfc822Name` und abhängig vom Zertifkatsinhaber auch den `uniformResourceIdentifier`(URI) des Zertifikatsinhabers enthalten, vgl. Tabellen 27 und 28.

Zertifikat	*C(Root)*	*$C_{CRL\text{-}S}$(Root)*	*C(Sub-CA)*
`rfc822Name`	E-Mail-Adresse	E-Mail-Adresse	E-Mail-Adresse
`uniformResource Identifier`	Webseite	Webseite	Optional: Webseite

Tabelle 27: Belegung SubjectAltNames für CAs

Zertifikat	*C(GWA)*	*C(EMT)*
`rfc822Name`	E-Mail-Adresse	E-Mail-Adresse
`uniformResource Identifier`	Optional: Webseite	Optional: Webseite

Tabelle 28: Belegung SubjectAltNames-Extension für Endnutzer

7. `IssuerAltName`
   - Extension-ID (OID): 2.5.29.18
   - Kritisch: Nein
   - Beschreibung: Diese Extension (spezifiziert in [9], 4.2.1.7) enthält weitergehende Informationen zum Aussteller (Issuer) - Namen.
   - Wert: `IssuerAltName` muss aus dem `rfc822Name` und den `uniformResourceIdentifier`(URI) des aus dem `SubjectAltName`-Feld des Zertifikatsausstellers (`Issuers`) bestehen, vgl. Tabelle 27.

8. `BasicConstraints`
   - Extension-ID (OID): 2.5.29.19
   - Kritisch: Ja
   - Beschreibung: Diese Extension (spezifiziert in [9], 4.2.1.9) gibt an, ob es sich bei dem gegebenen Zertifikat um eine CA handelt und wie viele CAs ihr folgen können.

- Werte: Die Extension hat in den einzelnen einzelnen Zertifikaten folgende Werte:

Zertifikat	*C(Root)*	*C_{CRL-S}(Root)*	*C(Sub-CA)*
`cA`	TRUE	FALSE	TRUE
`pathlengthConstraint`	1	/	0

Tabelle 29: Belegung Basic-Constraints-Extension für CAs

Zertifikat	*C_{TLS}(ENu)*	*C_{Enc}(ENu)*	*C_{Sign}(ENu)*
`cA`	FALSE	FALSE	FALSE
`pathlengthConstraint`	/	/	/

Tabelle 30: Belegung Basic-Constraints-Extension für Endnutzer

9. `ExtendedKeyUsage`
   - Extension-ID (OID): 2.5.29.37
   - Kritisch: Nein
   - Beschreibung: Dieser Extension (spezifiziert in [9], 4.2.1.12) gibt an, ob es sich bei dem gegebenen Zertifikat um ein TLS-Zertifikat handelt.
   - Werte: Die Extension ist in den TLS-Zertifikaten der Endnutzer vorhanden und hat dort folgende Belegung:

Zertifikat	*C_{TLS}(GWA)*	*C_{TLS}(EMT)*	*C_{TLS}(SMGW)*
Wert	TLS-Web-Server-Authentifikation (1.3.6.1.5.5.7.3.1) Optional: TLS-Web-Client-Authentifikation (1.3.6.1.5.5.7.3.2)	TLS-Web-Server-Authentifikation (1.3.6.1.5.5.7.3.1) Optional: TLS-Web-Client-Authentifikation (1.3.6.1.5.5.7.3.2)	TLS-Web-Client-Authentifikation (1.3.6.1.5.5.7.3.2)

Tabelle 31: Belegung ExtendedKeyUsage-Extension

10. `CRLDistributionPoints`
    - Extension-ID (OID): 2.5.29.31
    - Kritisch: Nein
    - Beschreibung: Diese Extension (spezifiziert in [9], 4.2.1.13) definiert eine Sequenz von Bezugspunkten für die aktuelle CRL der CA, die das Zertifikat ausgestellt hat. Es muss mindestens ein Verteilungspunkt für die vollständige CRL der austellenden CA im Zertifikat angegeben werden. Veröffentlicht die austellende CA auch Delta-CRLs, so sollte hierfür auch mindestens ein Verteilungspunkt angegeben werden.
    - Werte: Jeder Eintrag in der Sequenz `CRLDistributionPoints` besteht nur aus dem Feld `distributionPoint`, in welchem ein Bezugspunkt der aktuellen CRL

bzw. Delta-CRL der CA in Form einer HTTP- oder LDAP-URI enthalten ist, vgl. auch Kapitel 4.2.1. Die Felder `reason` und `cRLIssuer` werden nicht verwendet.

B CRL-Profil

Die CRLs, die in der SM-PKI verwendet werden, sind X.509-CRLs gemäß [9] und müssen konform zu dem in diesem Kapitel definierten CRL-Profil sein.

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

TBSCertList::=SEQUENCE {
   version  Version OPTIONAL,   --if present, must be v2
   signature  AlgorithmIndentifier,
   issuer  Name,
   thisUpdate  Time,
   nextUpdate  Time OPTIONAL,
   revokedCertificates  SEQUENCE OF SEQUENCE {
      userCertificate  CertificateSerialNumber,
                               --if present, version must be v2
      revocationDate  Time,
      crlEntryExtensions  Extension OPTIONAL,
   }
   crlExtensions  [0] EXPLICIT Extensions OPTIONAL
                               --if present, version must be v2
}
```

Der Wert von `signatureAlgorithm` gibt dabei den Algorithmus an, mit dem die CRL im Feld `signatureValue` signiert ist und muss identisch mit dem Inhalt von `signature` in `tbsCertList` sein. Die aktuellen kryptographischen Vorgaben über den dabei zu verwendenden Signaturalgorithmus sind in [6] zu finden. Das Feld `nextUpdate` muss vorhanden sein und enthält das Datum, an dem die nächste CRL spätestens veröffentlicht wird. Hiebei sind die zeitlichen Vorgaben aus Kapitel 4.2.2 zu beachten.

B.1 CRL-Extensions

Folgende CRL-Extensions dürfen verwendet werden:

- `authorityKeyIdentifier`
  - Extension-ID (OID): 2.5.29.35
  - Kritisch: Nein
  - Beschreibung: Die Extension muss vorhanden sein. Sie identifiziert den öffentlichen Schlüssel der CA, der zu dem privaten Schlüssel gehört, mit dem die CRL durch die CA signiert wurde (zur Codierung siehe A.2).
- `IssuerAltName`
  - Extension-ID (OID): 2.5.29.18
  - Kritisch: Nein
  - Beschreibung: Die Extension soll vorhanden sein. Sie enthält zusätzliche Informationen über den Herausgeber der CRL (zur Codierung siehe A.2).
- `CRLNumber`
  - Extenion-ID (OID): 2.5.29.20
  - Kritisch: Nein
  - Beschreibung: Die Extension muss vorhanden sein. Sie enthält die Seriennummer der CRL.
- `DeltaCRLIndicator`
  - Extension-ID (OID): 2.5.29.27
  - Kritisch: Ja
  - Beschreibung: Das Vorhandensein dieser Extension gibt an, dass es sich bei der gegebenen CRL um eine Delta-CRL handelt. Das Feld `BaseCRLNumber` identifiziert dabei die Nummer der jeweiligen vollständigen CRL, die als Basis für die Delta-CRL dient. Die Extension muss vorhanden sein, wenn es sich bei der CRL um eine Delta-CRL handelt. Ansonsten darf die Extension nicht vorhanden sein.
- `FreshestCRL`
  - Extension-ID (OID): 2.5.29.46
  - Kritisch: Nein
  - Beschreibung: In Falle einer vollständigen CRL können über diese Extension Bezugspunkte für DeltaCRL-Informationen angegeben werden. Wenn es sich bei der CRL um eine vollständige CRL handelt zu der auch eine Delta-CRL veröffentlicht wird, so sollte die Extension vorhanden sein. Ansonsten darf die Extension nicht vorhanden sein Die zu verwendende Syntax der `CRLDistributionPoints` ist die gleiche wie in A.2.

Weitere CRL-Extensions sollen nicht verwendet werden.

B.2 CRL-Entry-Extensions

Folgende CRL-Extensions dürfen verwendet werden:

- `ReasonCode`
  - Extension-ID (OID): 2.5.29.21
  - Kritsch: Nein
  - Beschreibung: Diese Extension soll in DeltaCRLs verwendet werden, um abgelaufene widerrufene Zertifikate aus der CRL zu entfernen. Hierfür wird der Code `removefromCRL` verwendet. Ansonsten darf diese Extension nicht verwendet werden. Die Sperre eines noch gültigen zurückgerufenen Zertifikats darf in keinem Fall wieder aufgehoben werden.
- `InvalidityDate`
  - Extension-ID (OID): 2.5.29.24
  - Kritisch: Nein
  - Beschreibung: Die Extension muss vorhanden sein und gibt den Zeitpunkt an, an dem das entsprechende Zertifikat gesperrt wurde.

Weitere CRL-Entry-Extensions sollen nicht verwendet werden.

C Zertifkatsrequests

Zur Beantragung von Zertifikaten werden die Kapitel C.1 spezifizierten Zertifikatsrequest-Pakete verwendet.

Die Zertifikatsrequest-Pakete können (je nach Hierarchieebene in der SM-PKI) aus einem oder mehreren Zertifikatsrequests (etwa für die verschiedenen Zertifikatstypen) bestehen. Bei einem Folgeantrag beinhaltet das Zertifikatsrequest-Paket eine zusätzliche äußere Signatur, wie in C.2 definiert. Die äußere Signatur muss mit dem privaten Schlüssel zu dem Signaturzertifikat des Antragstellers erstellt werden.

C.1 Datenstruktur eines Zertifikatsrequests

Für Zertifkatrequest-Pakete wird die Datenstruktur `CertReqMessages` aus [20] verwendet. Zur Identifikation dieses Datentyps muss die folgende OID als Content Type verwendet werden:

```
id-rfc4211CertReqMesgs OBJECT IDENTIFIER ::= {
   bsi-de pki(4) x509(1) cert-Requests(1) id-rfc4211-CertReqMsgs(1)
}
```

Der Datentyp `CertReqMessages` besitzt gemäß [20] folgende Struktur

```
CertReqMessages::=SEQUENCE SIZE (1..MAX) OF CertReqMsg

CertReqMsg::=SEQUENCE {
   certReq CertRequest,
   popo ProofOfPossesion OPTIONAL,
   regInfo SEQUENCE SIZE(1..MAX) of AttributeTypeAndValue OPTIONAL
}

CertRequest::=SEQUENCE {
   certReqId INTEGER,
   certTemplate  CertTemplate
   controls Controls OPTIONAL
}

CertTemplate::=SEQUENCE {
```

```
        version [0] Version OPTIONAL,
        serialNumber [1] INTEGER OPTIONAL,
        signingAlg [2] Algorithm Identifier OPTIONAL,
        issuer [3] Name OPTIONAL,
        validity [4] OptionalValidity OPTIONAL,
        subject [5] Name OPTIONAL,
        publicKey [6] SubjectPublicKeyInfo OPTIONAL,
        issuerUID [7] UniqueIdentifier OPTIONAL,
        subjectUID [8] UniqueIdentifier OPTIONAL,
        extensions [9] Extensions OPTIONAL,
}
```

Die Datenfelder `regInfo` in `CertReqMSg` und `controls` in `CertRequest` sollen nicht verwendet werden. Die Felder `version, serialNumber, signingAlg, issuer, validity, issuerUID, subjectUID` in `CertTemplate` sollen ebenso nicht verwendet werden.

Die Bedeutungen der Felder in dem Zertifikatsrequest werden im Folgenden spezifiziert.

- `CertReqId` enthält einen ganzzahligen Wert, welcher den Zertifikatsrequests mit einer Zertifikatsresponse dient.
- `certTemplate` enthält das Zertifikatstemplate mit den Datenfeldern, die für die Zertifikatsausstellung durch den Antragsteller an die CA zu liefern sind.

In den vorhandenen Feldern von `CertTemplate` sollen folgende Werte stehen:

- `subject` enthält den Namen des Antragstellers.
- `SubjectPublicKeyInfo` enthält den Public Key, für den das Zertifikat beantragt wird.
- `Extensions` enthält die durch der Antragsteller anzugebenden Extensions.

Folgende Extensions können im Zertifikatsrequest enthalten sein:

- `AuthorityKeyIdentifier` sollte vorhanden sein und den privaten Schlüssel der CA, den der Antragsteller für die Signatur des ausgestellten Zertifikats erwartet benennen. Enthält das von der CA ausgestellte Zertifikat einen anderen `AuthorityKeyIdentifier` als im zugehörigen Request angegeben, sollte das entsprechende Zertifikat in der Response-Nachricht mitgeliefert werden.
- `SubjectkeyIdentifier` ist optional und enthält den `SubjectkeyIdentifier` zu dem öffentlichen Schlüssel, für den das Zertifikat beantragt wird.
- `KeyUsage` muss vorhanden sein und gibt an für welche Verwendungszwecke das Zertifikat beantragt wird.
- `SubjectAltNames` muss vorhanden sein und enthält die `subjectaltNames`, die im Zertifkat enthalten sein sollen.

- `ExtendedKeyUsage` muss vorhanden sein, wenn ein TLS-Zertifikat beantragt wird. Ansonsten darf die Extension nicht vorhanden sein.

Das Feld `popo` in `CertReqMsg` muss vorhanden und vom Typ `POPOSigningKey` sein.

```
POPOSigningKey::= SEQUENCE {
    poposkInput [0] POPOSigningKeyInput OPTIONAL,
    algorithmIdentifier AlgorithmIdentifier,
    signature BIT STRING
}
```

Das Feld `poposkInput` wird nicht verwendet, d.h. die Signatur wird über den DER-codierten Wert von `certReq` gebildet. Die anderen Felder haben folgende Werte:

- `algorithmIdentifier` enthält den Signaturalgorithmus, mit dem die Signatur im Feld `signature` erzeugt wurde. Für die Auswahl des Signaturalgorithmus sind die aktuellen Vorgaben aus [6] zu beachten.
- `Signature` enthält den Wert der Signatur.

Handelt es sich bei dem Zertifikatsrequest-Paket um einen Folgeantrag, so muss dieses Paket zusätzlich mit einer äußeren Signatur versehen werden. Die äußere Signatur wird dann mit dem privaten Signaturschlüssel erstellt. Besitzt der Antragsteller kein gültiges Signaturzertifikat, ist die äußere Signatur des Request-Pakets optional. Das Request-Paket kann in diesem Fall zusätzlich durch eine andere vertrauenswürdige Instanz signiert sein. Die Authentizität eines Zertifikatsrequests durch den Antragsteller muss in jedem Fall sichergestellt werden.

Für die äußere Signatur eines Zertifikatsrequest-Pakets wird die Datenstruktur `SignedData` gemäß Abschnitt C.2 verwendet.

C.2 Signatur eines Zertifikatsrequests

Für die äußere Signatur eines Zertifikatsrequest-Pakets muss der Datentyp `ContentInfo` gemäß [21] mit Content Type `SignedData` verwendet werden.

```
ContentInfo::=SEQUENCE {
    contentType  ContentType,
    content  [0] EXPLICIT ANY DEFINED BY contentType
}
```

Zur Identifikation dieses Content Types muss demnach die folgende OID verwendet werden:

```
id-signedData OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) rsadsi(113549) pkcs(1) pkcs7(7) 2
```

```
}
```

Das Feld `content` ist dann vom Typ `SignedData` und hat folgendes Format:

```
SignedData :: SEQUENCE
   version  CMSVersion,
   digestAlgorithms  DigestAlgorithmIdentifiers,
   encapContentInfo  EncapsulatedContentInfo,
   certificates  [0] IMPLICIT CertificateSet OPTIONAL,
   crls [1] IMPLICIT RevocationInfoChoices OPTIONAL,
   signerInfos SignerInfos
}

DigestAlgorithmIdentifiers ::= SET OF DIGESTAlgorithmIdentifier

EncapsulatedContentInfo ::= SEQUENCE {
   eContentType ContentType,
   eContent [0] EXPLICIT OCTET STRING OPTIONAL
}

ContentType ::= OBJECT IDENTIFIERs

SignerInfos ::= SET OF SignerInfo

SignerInfo ::= SEQUENCE {
   version CMSVersion,
   sid SignerIdentifier,
   digestAlgorithm SignatureAlgorithmIdentifier,
   signature SignatureValue
}

SignerIdentifier ::= CHOICE {
   issuerandSerialNumber IssuerAndSerialNumber,
   subjectKeyIdentifier [0] SubjectKeyIdentifier
}
```

```
SignatureValue ::= OCTET STRING
```

Das Paket (`CertReqMessages`) der Zertifikatsrequests für die zu beantragenden Zertifikate (etwa TLS, Verschlüsselungs- bzw. Signaturzertifikat) ist dabei, wie in Abschnitt C.1 definiert, in `EncapsulatedContentInfo` enthalten. Im Feld `eContentType` muss hierbei die OID `id-rfc4211-CertReqMsgs` enthalten sein. Das Feld `Certificates` enthält das Zertifikat, mit dem das Zertifikatsrequest-Paket signiert ist, und das Feld `SignerInfos` enthalt die äußere Signatur.

D Protokolle für das Zertifikatsmanagement

In diesem Kapitel werden die Kommunikationsprotokolle für die Beantragung und Verteilung von Zertifikaten aus der SM-PKI spezifiziert.

Für jede Message, die in den Kommunikationsprotokollen verwendet wird, werden folgende Bestandteile spezifiziert:

- Einsatzzweck
- Eingabeparameter
- Ausgabeparameter
- Mögliche Return Codes

Die Messages können durch den Empfänger entweder synchron oder asynchron verarbeitet werden. Bei der synchronen Verarbeitung wird das Ergebnis der Verarbeitung in den Ausgabeparametern der Message zurück an der Sender gesendet.

Bei der asynchronen Verarbeitung wird das Ausgabeparameter-Feld nicht verwendet. Es wird nur ein Return Code an den Sender zurückgegeben und eine eventuelle Ausgabe wird anschließend (enthalten in den Eingangsparametern) einer Callback-Message an den Sender der ursprünglichen Message zurück gesendet. Sowohl Sender als auch Empfänger müssen dazu als Client (Senden einer Message) als auch als Server (Empfangen einer Message) arbeiten.

D.1 Parameter und Return Codes

D.1.1 Parameter

Die folgenden Parameter werden in den Kommunikationsprotokollen für das Zertfikatsmanagement verwendet:

- `callbackIndicator`:

  Mit diesem Parameter informiert der Absender der Nachricht den Empfänger, ob er mit Callback-Messages umgehen kann.

  Unterstützt der Absender Callback-Messages, so muss der Parameter auf den Wert `callback_possible` gesetzt werden. Der Empfänger kann die Message dann synchron oder asynchron (durch Senden der Response mit der entsprechenden Callback-Message) verarbeiten. Kann der Absender keine Callback-Messages verarbeiten, so muss der Parameter auf den Wert `callback_not_possible` gesetzt werden.

- `MessageID`:

  Dieser Parameter enthält die ID der Message. Der Wert muss die Message eindeutig unter aller Messages des Absenders identifizieren und wird vom Absender der Message bestimmt.

  Unterstützt der Absender Callback-Messages (`CallbackIndicator=callback_possible`), so muss die `messageID` vorhanden sein. Eine Response Message soll dann stets dieselbe

messageID wie die Ursprungs-Message enthalten, um eine eingehende Response Message der korrekten Ursprungsnachricht zuordnen zu können. Gibt der Absender an keine Callback-Messages zu unterstützen (CallbackIndicator=callback_not_possible), so muss die messageID weggelassen werden.

- responseURL:

  Dieser Parameter enthält die URL, an der der Absender der Message bei asynchroner Verarbeitung die Response-Message erwartet.

  Gibt der Absender der Message an, Callback Messages zu unterstützen (CallbackIndicator= callback_possible), so muss die responseURL in der Message angegeben werden. Unterstützt der Absender keine asynchrone Verarbeitung (CallbackIndicator= callback_not_possible), so muss der Parameter responseURL weggelassen werden.

- statusInfo:

  Dieser Parameter enthält einen Status Code über die Verarbeitung der zugehörigen Message.

- certReq:

  Der Parameter certReq enthält das Zertifikatsrequest-Paket. Die Codierung ist in C spezifiziert.

- certificateSeq:

  Dieser Parameter enthält ein oder mehrere Zertifikate, sofern die (Ursprungs-)Message korrekt verarbeitet worden ist.

  Werden mit einer Message Zertifikate versendet, so müssen diese in certificateSeq enthalten sein, Ansonsten wird der Parameter weggelassen.

D.1.2 Return Codes

Folgende Return Codes werden in diesem Kapitel häufig verwendet:

- ok_syntax:

  Mit diesem Return Code wird der Erhalt einer Message bestätigt. Die Syntax der erhaltenen Message wurde erfolgreich verifiziert, die Message wird asynchron verarbeitet und mit der entsprechenden Response-Message an die URL aus dem Parameter responseURL gesendet.

- ok_reception_ack:

  Mit diesem Return Code wird der Erhalt einer Message bestätigt. Die Syntax der erhaltenen Message wurde noch nicht verifiziert, die Message wird asynchron verarbeitet und mit der entsprechenden Response-Message an die URL aus dem Parameter responseURL gesendet.

- ok_received_correctly:

  Die Message wurde erhalten und synchron verarbeitet. Es wurde kein Output erzeugt.

- failure_syntax_request:

  Das empfangene Request-Paket ist syntaktisch nicht korrekt.

- failure_syntax_soap:

  Die empfangene SOAP-Message ist syntaktisch nicht korrekt.

- `failure_synchronous_processing_not_possible`:

  Der Absender hat angegeben keine Callback-Messages zu unterstützen, weshalb die Message synchron verarbeitet werden müsste. Der Empfänger kann die Message jedoch nicht synchron verarbeiten. In diesem Fall muss die weitere Vorgehensweise durch organisatorische Maßnahmen festgelegt werden.

- `failure_messageID_unknown`:

  Die enthaltene `messageID` passt zu keiner vorherigen `messageID`.

- `failure_internal_error`:

  Jeder andere Fehler.

- `ok_cert_available`:

  Die Message wurde erfolgreich verarbeitet. Der Parameter `certificateSeq` enthält ein oder mehrere Zertifikate.

- `failure_inner_signature`:

  Die Verifikation der inneren Signatur des gegenwärtigen Zertifikatsrequests ist fehlgeschlagen.

- `failure_outer_signature`:

  Die Verifikation der äußeren Signatur des gegenwärtigen Zertifikatsrequests ist fehlgeschlagen.`failure_expired`:

  Das Zertifikat zur Verifikation der äußeren Signatur ist abgelaufen.

- `failure_request_not_accepted`:

  Die Message wurde zwar korrekt verarbeitet, allerdings wurde der Request nicht akzeptiert. In diesem Fall muss die weitere Vorgehensweise durch organisatorische Maßnahmen festgelegt werden.

D.2 Nachrichten zur Verteilung von Zertifikaten

Im Folgenden werden die zu verwendenden Messages für das Management der X.509-Zertifikate aus der SM-PKI spezifiziert.

D.2.1 RequestCertificate

Mit dieser Nachricht beantragen Endnutzer bzw. Sub-CAs die Erzeugung von neuen Zertifikaten bei ihrer Sub-CA bzw. der Root-CA.

Input parameters:

- `callbackIndicator` **(REQUIRED)**
- `messageID` **(CONDITIONAL)**
- `responseURL` **(CONDITIONAL)**
- `certReq` **(REQUIRED)**

Output parameters:

- `certificateSeq` **(CONDITIONAL)**

Return codes:

- `ok_cert_available`
- `ok_syntax`
- `ok_reception_ack`
- `failure_inner_signature`
- `failure_outer_signature`
- `failure_domain_parameters`
- `failure_expired`
- `failure_request_not_accepted`
- `failure_syntax_request`
- `failure_syntax_soap`
- `failure_synchronous_processing_not_possible`
- `failure_internal_error`

D.2.2 GetCACertificates

Diese Nachricht wird von Endnutzern bzw. Sub-CAs verwendet, um alle relevanten CA-Zertifikate bei ihrer Sub-CA bzw. der Root-CA abzurufen.

Input parameters:

- `callbackIndicator` **(REQUIRED)**
- `messageID` **(CONDITIONAL)**
- `responseURL` **(CONDITIONAL)**

Output parameters:

- `certificateSeq` **(CONDITIONAL)**

Return codes:

- `ok_cert_available`
- `ok_syntax`
- `ok_reception_ack`
- `failure_syntax_request`
- `failure_syntax_soap`
- `failure_synchronous_processing_not_possible`

- failure_internal_error

Bemerkung: Wird die Nachricht vom Empfänger erfolgreich verarbeitet und akzeptiert, so muss der Empfänger dem Absender in der Antwort alle relevanten CA-Zertifikate zusenden. Bei synchroner Verarbeitung muss dies via dem Output-Parameter certificateSeq erfolgen. Bei asynchroner Verarbeitung wird zur Versendung der angeforderten Zertifikate die Response-Message SendCertificates verwendet.

D.2.3 SendCertificates

Verarbeitet eine CA eine der Messages RequestCertificate oder GetCACertificates asynchron, so wird die Response Message SendCertificates für die Übersendung des Verarbeitungsergbnisses verwendet. Die Message wird stets an die URL gesendet, welche im Parameter responseURL der zuvor empfangenen Message enthalten war.

Die Message kann auch dazu verwendet werden, registrierten Instanzen die Verfügbarbeit neuer X.509-Zertifikate bekannt zu machen. In diesem Fall muss dann die messageID weggelassen werden.

Diese Message muss durch den Empfänger immer synchron verarbeitet werden.

Input parameters:

- messageID **(CONDITIONAL)**
- statusInfo **(REQUIRED)**

  The following status codes are possible:

  - ok_cert_available
  - failure_inner_signature
  - failure_outer_signature
  - failure_expired
  - failure_domain_parameters
  - failure_request_not_accepted
  - failure_not_forwarded
  - failure_syntax_request
  - failure_syntax_soap
  - failure_internal_error
- certificateSeq **(CONDITIONAL)**

Output parameters:

none

Return codes:

- ok_received_correctly

- `failure_syntax_request`
- `failure_syntax_soap`
- `failure_messageID_unknown`
- `failure_internal_error`

D.2.4 GeneralMessage

Diese Message kann von einer PKI-Instanz verwendet werden, einer anderen Instanz beliebige Mitteilungen (etwa Klartextnachrichten) zu senden.

Input parameters:

- `callerID` **(REQUIRED)**

  Dieser Parameter enthält den Identifier der Instanz, die die Kommunikation veranlasst hat. Der Wert muss in der SM-PKI eindeutig sein.
- `messageID` **(REQUIRED)**
- `subject` **(REQUIRED)**

  Dieser Parameter enthält den Betreff der Message und sollte den Inhalt des Message-Body kurz beschreiben.
- `body` **(REQUIRED)**

  Dieser Parameter enthält den Message-Body und enthält Klartext, der nicht zur automatischen Verarbeitung gedacht ist.

Output parameters:

Keine

Return codes:

- `ok_received_correctly`
- `failure_internal_error`

D.3 Implementierung als Web Service

Für die Implementierung der hier angegebenen Protokolle als Web Service wird es empfohlen, SOAP-Nachrichten zu verwenden, die konform zu den dieser Technischen Richtlinie angehängten WSDL-Dateien sind.

D.3.1 Web Services der Root-CA

Eine Root-CA muss einen Web Service implementieren, der folgende Nachrichten unterstützt:

- `RequestCertificate`
- `GetCACertificates`

- `GeneralMessage`

D.3.2 Web Services der Sub-CAs

Für die Kommunikation mit einem Endnutzer muss eine Sub-CA folgende Nachrichten unterstützten:

- `RequestCertificate`
- `GetCACertificates`

Für die Kommunikation mit einer Root-CA und einem Endnutzer muss eine Sub-CA einen Web-Service implementieren, der folgende Nachrichten unterstützt:

- `GeneralMessage`

Unterstützt eine Sub-CA auch Callback-Nachrichten durch die Root-CA, so sind außerdem folgende (Callback-)Nachrichten zu implementieren:

- `SendCertificates`

D.3.3 Web Services der Endnutzer

Ein Endnutzer, der Callback-Nachrichten durch die Sub-CA unterstützt, muss folgende Nachrichten per Web Service unterstützen:

- `SendCertificates`
- `GeneralMessage`

E LDAP-Schema und Entry-Profil für Zertifikate im Verzeichnisdienst

Das verwendete LDAP-Schema für die Veröffentlichung der Zertifikate im Verzeichnisdienst basiert auf den in [22] und [23] definierten Schemata. Es werden zwei zusätzliche Objektklassen definiert.

```
Objectclass (0.4.0.127.0.7.2.1.1.1.1 NAME 'countryExt'
    SUP top AUXILIARY
    MUST ( c ) )
Objectclass (0.4.0.127.0.7.2.1.1.1.2 NAME 'serialNoExt'
    SUP top AUXILIARY
    MUST ( c ) )
```

In einem Verzeichniseintrag können folgende Attribute veröffentlicht werden. Es dürfen nur die genannten Attribute veröffentlicht werden.

Attribut	*LDAP-Attribut*	*Objektklasse*	*m/o/c*
Nachname	`sn`	`inetOrgPerson`	m
Common Name	`cn`	`inetOrgPerson`	m
Organisation	`o`	`inetOrgPerson`	m
Organisationseinheit	`ou`	`inetOrgPerson`	c[6]
Ort	`l`	`inetOrgPerson`	c[6]
Staat	`c`	`CountryExt`	m
Sequenznummer der Zertifikate	`serialNumber`	`serialNoExt`	m
Straße	`street`	`inetOrgPerson`	c[6]
Postleitzahl	`postal code`	`inetOrgPerson`	c[6]
Ort	`l`	`inetOrgPerson`	c[6]
State	`st`	`inetOrgPerson`	c[6]
Zertifikat	`userCertificate`	`inetOrgPerson`	m

Die Werte der Attribute entsprechen dabei jeweils denen der Attribute in Subject-Feld des Zertifikats, sofern diese dort vorhanden sind. Das Attribut `userCertificate` kann in einem Eintrag mehrmals vorkommen (etwa für die Veröffentlichung von Endnutzer-Zertifikats-Paketen, d.h. Endnutzer-Zertifikaten mit der gleichen Sequenznummer für verschiedene Verwendungszwecke). Alle anderen angegebenen Attribute dürfen in einem Eintrag nur jeweils einmal verwendet werden, selbst wenn das jeweilige LDAP-Schema eine mehrfache Verwendung zuließe.

6 Dieses Attribut ist genau dann in dem entsprechenden LDAP-Eintrag vorhanden, wenn es im Subject-Feld der zugehörigen Zertifikate (des `userCertificate`-Attributs) angegeben ist.

Das Attribut `sn` erhält in jedem Eintrag den Wert `'1'`. Das Attributpaar `'cn'` und '`serialNumber`' soll als `RDN` zur Identifikation eines Eintrags verwendet werden.

Jede CA muss die von ihr ausgestellten Zertifikate in einem Verzeichnis mit dem DN der Form 'dc=Certificates, dc=SM-PKI' veröffentlichen.

F Elliptische Kurven

Von dieser Technischen Richtlinie werden Brainpool- und NIST-Kurven über Primkörpern unterstützt.

- Die OBJECT IDENTIFIER der Brainpool-Kurven sind in [24] und [25] spezifiziert.
- Die OBJECT IDENTIFIER der NIST-Kurven über Primkörpern sind in [26] spezifiziert.

Die Vorgaben über die aktuell zu verwendenden elliptischen Kurven aus [6] sind zu beachten.

Literaturverzeichnis

[1] BSI TR-03109, SMART ENERGY (Dachdokument), 2013
[2] BSI TR-03109-1, Smart Meter Gateway, 2013
[3] BSI TR-03109-2, Das Sicherheitsmodul eines Smart Metering Systems, 2013
[4] BSI TR-03109-1 Anlage VI, Smart Meter Gateway - Anhang Betriebsprozesse, 2013
[5] BSI TR-03109-3, Kryptographische Vorgaben, 2013
[6] BSI TR-03116-3, eCard-Projekte der Bundesregierung - Kryptographische Vorgaben für die Infrastruktur von intelligenten Messsystemen,
[7] N.N. Root-CP, tbd (Certificate Policy der Wurzelinstanz für die SM-PKI),
[8] BSI, Sicherheitsleitlinien der Wurzelzertifizierungsinstanz der Verwaltung, Version 3.2,
[9] IETF RFC 5280, D. Cooper, S. Santesson, S. Farrell, S. Boyen, R. Housley, W. Polk, Internet X.509 Public Key Infrastructures - Certificates and Certificate Revocation List (CRL) Profiles, 2008
[10] IETF RFC 4511, J. Sermersheim, Lightweight Directory Access Protocol (LDAP): The Protocol, 2006
[11] IETF RFC 2616, R. Fielding, UC Irvine, J. Gettys, Compaq/W3C, J. Mogul, Compaq, H. Frystyk, W3C/MIT, L. Masinter, Xerox, P. Leach, Microsoft, T. Berners-Lee, W3C/MIT, Hypertext Transfer Protocol -- HTTP/1.1, 1999
[12] BSI CC-PP-0042 , Protection Profile Cryptografic Modules, Security Level "Moderate", Version 1.01 (derzeit in Überarbeitung), 2008
[13] BSI CC-PP-0045, Protection Profile Cryptografic Modules, Security Level "Enhanced", Version 1.01, 2008
[14] CEN/TC 224 EN 14169-1, Protection profiles for Secure signature creation device — Part 2: Device with key generation, 2009
[15] BSI BSI-CC-PP-0077 , Protection Profile for the Security Module of a Smart Metering System, 2011
[16] ISO/IEC 27001, Information technology - Security techniques - Information security management systems - Requirements, 2005
[17] BSI-Standard 100-Serie, IT-Grundschutz-Standards, 2008
[18] IETF RFC 5480, S. Turner, R. Housley, T. Polk, Elliptic Curve Cryptography Subject Public Key Information,
[19] DIN, 43863-5 Herstellerübergreifende Identifikationsnummer für Messeinrichtungen, 2012
[20] IETF RFC 4211, J. Schaad, Internet X.509 Public Key Infrastructure Certificate Request Message Syntax, 2005
[21] IETF RFC 5652, R. Housley, Cryptographic Message Syntax (CMS), 2009
[22] IETF RFC 4519, A. Sciberras, Lightweight Directory Access Protocol (LDAP): Schema for User Applications, 2006
[23] IETF RFC 2798, M. Smith, Definition of the inetOrgPerson LDAP Object Class, 2000
[24] BSI TR-03111, Elliptic Curve Cryptography, Version 2.0, 2012
[25] IETF RFC 5639, M. Lochter, J. Merkle, Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation, 2010
[26] ANSI X9.62, Public Key Cryptography for the Financial Services Industry . The Elliptic Curve Digital Signature Algorithm (ECDSA), 2005