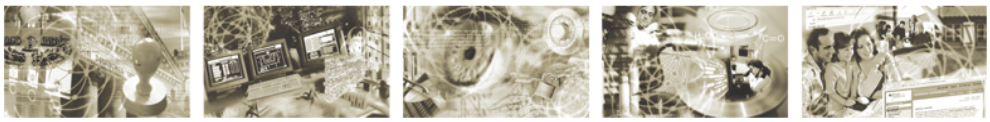
Bundesamt
für Sicherheit in der
Informationstechnik



Technische Richtlinie BSI TR-03109-4

Smart Metering PKI - Public Key Infrastruktur für Smart Meter Gateways

Version 1.2.1

Datum: 09.08.2017

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.

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 festgelegt. Abschließend wird das Sperrlistenmanagement der PKI-Teilnehmer erläutert.

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 **[14]** 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 [14], insbesondere die Teile 1, 2, 3 und 6 (vgl. [16], [17], [15], [18], [19]):**

  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, die BSI TR-03109-6 definiert die Anforderungen an die SMGW-Administration (GWA).

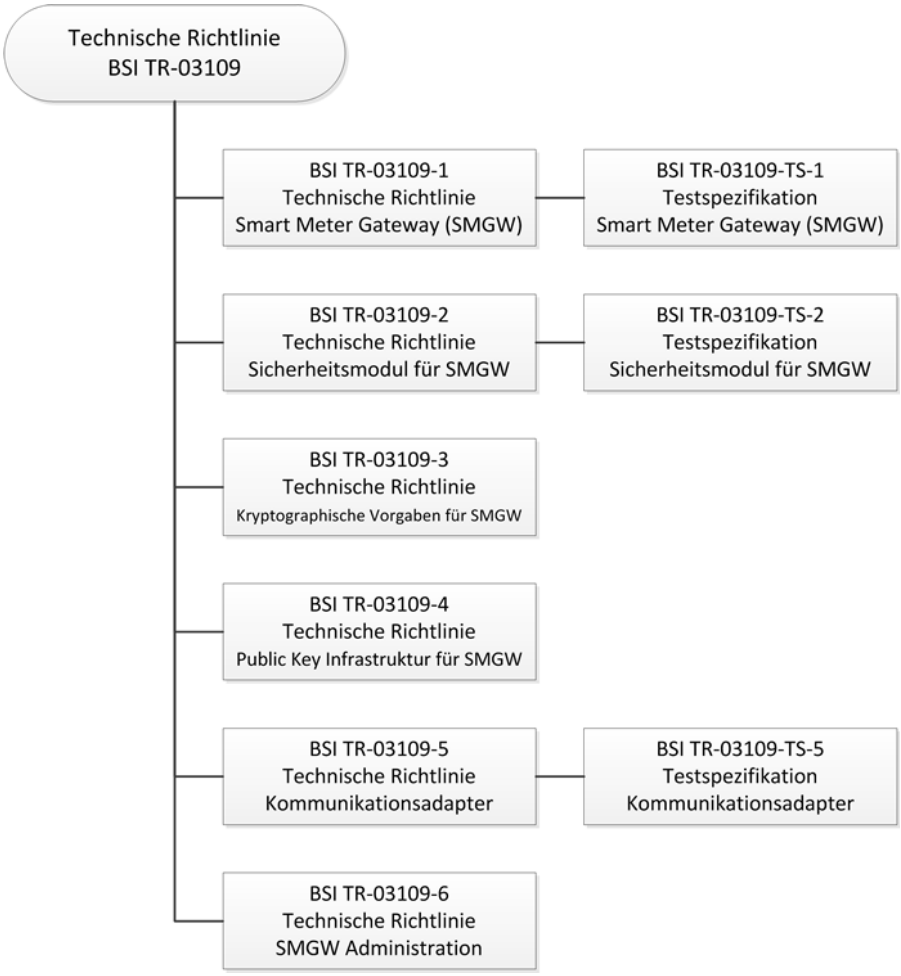


Abbildung 1: Dokumentenstruktur der BSI TR-03109

- **BSI TR-03116 Kryptographische Vorgaben für Projekte der Bundesregierung – Teil 3: Intelligente Messsysteme [21]**:

  In dieser Technischen Richtlinie werden die Sicherheitsanforderungen für den Einsatz kryptographischer Verfahren in der Infrastruktur von Messsystemen im Energiesektor beschrie-

ben. Insbesondere wird dort definiert, welche kryptographischen Algorithmen und Schlüssellängen für die Zertifikate in der SM-PKI eingesetzt werden müssen.

- **Certificate Policy der Smart Metering-PKI (SM-PKI Policy) [13]**:

  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. Des Weiteren werden auch die Sicherheitsanforderungen an Teilnehmer der PKI definiert. 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
GW	Gateway
GWA	Gateway-Administrator
GWH	Gateway-Hersteller

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

2 Architektur der SM-PKI

Die SM-PKI hat die folgende dreistufige hierarchische Struktur:

- **Hoheitlicher Vertrauensanker (Root-CA)**
- **Endnutzerzertifizierung (Sub-CA)**
- **Endnutzer: EMT, GWA, GWH und SMGW (vgl. [13])**

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

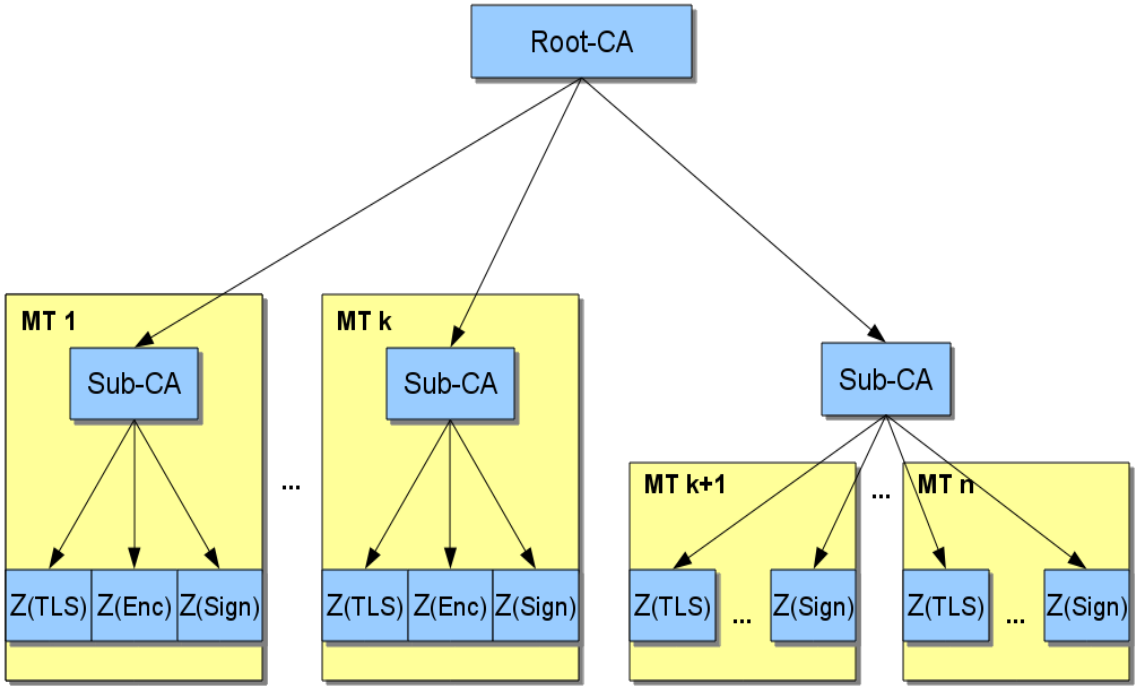


Abbildung 2: Architektur der SM-PKI (Beispiel)

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 (SM-PKI Policy) [13] 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 als 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 Ausstellung ihrer Sperrlisten. Das Zertifikat ist mit dem privaten Schlüssel des Root-Zertifikats *C(Root)* signiert.

Des Weiteren verfügt die Root-CA über ein TLS-Signer-Zertifikat $C_{TLS\text{-}S}(Root)$ zur Ausstellung der eigenen TLS-Zertifikate $C_{TLS}(Root)$. Hinsichtlich der gesicherten Kommunikation zwischen Root-CA und einer Sub-CA über die Web-Service-Schnittstelle wird dieser zusätzlich zur Ausstellung von TLS-Zertifikaten für die Sub-CAs $C_{TLS,Root}(Sub\text{-}CA)$ verwendet (siehe Abschnitt 2.2.2).

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, Link-Zertifikate, CRL-Signer-Zertifikate und TLS-Signer-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-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-CA-CRL verwendet.
$C_{TLS\text{-}S}(Root)$	Privater Schlüssel zu *C(Root)*	Der zugehörige private Schlüssel wird zur Signatur der TLS-Zertifikate der Root und der Root-TLS-CRL verwendet. Ferner werden mit diesem auch die Sub-CA TLS-Zertifikate signiert, die für die Kommunikation mit der Root über Web-Service-Schnittstelle verwendet werden.
$C_{TLS}(Root)$	Privater Schlüssel zu $C_{TLS\text{-}S}(Root)$	Nutzung bei dem Verzeichnisdienst der Root zur Authentisierung gegenüber dem Kommunikationspartner und zum Aufbau eines verschlüsselten, integritätsgesicherten Kanals.

Tabelle 1: Zertifikate der Root-CA

2.1.3 Verzeichnisse und Sperrlisten der Root-CA

Die Root-CA muss einen Verzeichnisdienst betreiben, in dem alle von ihr ausgestellten Zertifikate veröffentlicht werden (siehe [13]). Auf den Verzeichnisdienst dürfen ausschließlich die Teilnehmer der SM-PKI zugreifen können (siehe [13]). Die Sperrlisten der Root müssen öffentlich zugänglich sein (siehe Tabelle 13). Darüber hinaus müssen der Vertrauensanker (Root- und Link-Zertifikate), CRL-Signer-Zertifikate und TLS-Signer-Zertifikate 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 SM-PKI Policy sein [13].

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, der Sperrliste der Sub-CA und den TLS-Zertifikaten der Sub-CA.

Die TLS-Zertifikate $C_{TLS}(Sub\text{-}CA)$ der Sub-CA müssen zur Authentisierung und Aufbau eines verschlüsselten, integritätsgesicherten Kanals bei dem Verzeichnisdienst der Sub-CA verwendet werden. Des Weiteren können die TLS-Zertifikate genutzt werden, um sich gegenüber den Kommunikationspartnern der Sub-CA zu authentisieren und die Verbindung zu verschlüsseln. Die TLS-Zertifikate $C_{TLS,Root}(Sub\text{-}CA)$ zur gesicherten Kommunikation mit der Root-CA über die Web-Service-Schnittstelle werden von der Root ausgestellt (siehe 2.1.2).

Zertifikat	*Signatur des Zertifikats*	*Funktion des privaten Schlüssels*
C(Sub-CA)	Privater Schlüssel zu *C(Root)*	Signatur von Endnutzerzertifikaten und der Sub-CA-CRL
C_{TLS}*(Sub-CA)*	Privater Schlüssel zu *C(Sub-CA)*	Nutzung bei dem Verzeichnisdienst der Sub-CA zur Authentisierung gegenüber dem Kommunikationspartner und zum Aufbau eines verschlüsselten, integritätsgesicherten Kanals Nutzung bei dem Verzeichnisdienst der Sub-CA zur Authentisierung gegenüber dem Kommunikationspartner und zum Aufbau eines verschlüsselten, integritätsgesicherten Kanals
$C_{TLS,Root}$*(Sub-CA)*	Privater Schlüssel zu $C_{TLS\text{-}S}$*(Root)*	Nutzung bei der Kommunikation mit der Root über die Web-Service-Schnittelle zur Authentisierung und zum Aufbau eines verschlüsselten, integritätsgesicherten Kanals

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 (siehe [13]). 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 ihrer aktuellen Sperrlisten.

2.3 Endnutzer

2.3.1 Beschreibung

Den Endnutzern werden Zertifikate von den von ihnen gewählten Sub-CAs unter Einhaltung der Vorgaben der SM-PKI Policy und dieser Technischen Richtlinie ausgestellt. Der Besitz dieser Zertifikate ist die Voraussetzung für eine mögliche Kommunikation zwischen den Endnutzern.

Es gibt verschiedene Typen von Endnutzern (vgl. [13]):

- **Externe Marktteilnehmer (EMT)**
- **Gateway-Administrator (GWA)**
- **Gateway-Hersteller (GWH)**
- **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, 4.2.2.2 sowie [16], [15]).

2.3.2 Verwendungszwecke der Zertifikate

Die Zertifikate werden zum sicheren Datenaustausch zwischen autorisierten Marktteilnehmern selbst und bei Kommunikation dieser mit 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 (vgl. [16], [15]).

Entsprechend sind verschiedene Verwendungszwecke für die Zertifikate gegeben. Im Folgenden werden die von der Sub-CA für die Endnutzer ausgestellten Zertifikate und deren Verwendungszwecke beschrieben:

- **TLS-Zertifikate**: Aufbau eines verschlüsselten/integritätsgesicherten und gegenseitig authentisierten Kanals zwischen den Teilnehmern der SM-PKI.
- **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.

Jeder Endnutzer besitzt jeweils ein Zertifikatstripel besteht aus TLS-, Verschlüsselungs-, und Signaturzertifikat. Diese drei Zertifikate haben immer die gleiche Zertifikatslaufzeit (siehe Abschnitt 2.4).

2.3.3 Zertifikate eines externen Marktteilnehmers

Ein externer Marktteilnehmer erhält die benötigten Zertifikate von einer Sub-CA (Die Spezifikation in diesem Dokument sind für einen aktiven und einen passiven EMT identisch (vgl. [13]), entsprechend werden diese hier nicht unterschieden).

Mit seinem TLS-Zertifikat kann ein EMT direkt mit einem SMGW kommunizieren. Des Weiteren können die Zertifikate zur Kommunikation mit der Sub-CA verwendet werden.

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	m
$C_{Enc}(EMT)$	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

2.3.4 Zertifikate eines Gateway-Administrators

Der GWA besitzt TLS-, Verschlüsselungs- und Signaturzertifikate, welche ihm von einer Sub-CA ausgestellt werden (GWA können mehrere Zertifikatstripel besitzen, weitere Erläuterungen siehe [16] und [13]).

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

Tabelle 4: Zertifikate des GWAs

2.3.5 Zertifikate eines Gateway-Herstellers

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

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

Tabelle 5: Zertifikate des GWHs

2.3.6 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.

2.3.6.1 Gütesiegel-Zertifikate

Die Gütesiegel-Zertifikate werden im Herstellungsprozess des SMGWs zusammen mit den entsprechenden Schlüsselpaaren auf dem Sicherheitsmodul gespeichert (vgl. Lifecycle-Beschreibung in [17]). 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.

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

Tabelle 6: Gütesiegel-Zertifikate des SMGWs

2.3.6.2 Wirk-Zertifikate

Bei Übergang in den Wirkbetrieb werden die Gütesiegel-Zertifikate im SMGW durch Wirk-Zertifikate ersetzt (vgl. Kp. 3.4 bzw [13]). 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 7: Wirk-Zertifikate des SMGWs

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.

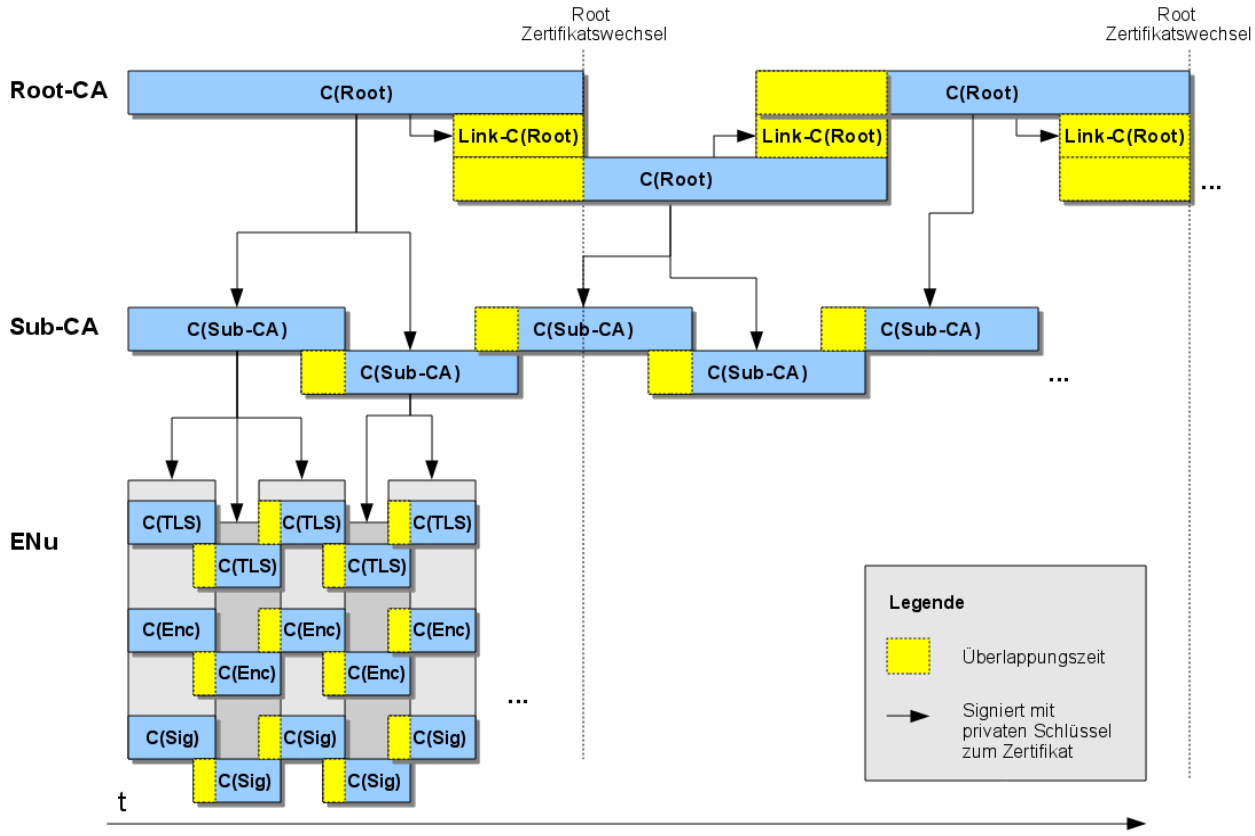


Abbildung 3: Prozess der Zertifikatsausgabe und -erneuerung

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. 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 Zertifikats 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-Zertifikats 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	■	■	■	■	■
EMT			■		■
GWA			■		■
GWH			■		■
SMGW			■		■

Tabelle 8: 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-S}(Root), C(Sub-CA), C_{TLS-S}(Root), C_{TLS}(Root), $C_{TLS,Root}$(Sub-CA)*
Sub-CA	*C_{TLS}(ENu), C_{Enc}(ENu), C_{sign}(ENu), C_{TLS}(Sub-CA)*

Tabelle 9: 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 Endnutzerzertifikaten durchzuführen.

2.6 Abgrenzung der SM-PKI

Aufgrund der unterschiedlichen Schnittstellen erfordert der Betrieb des SMGW mehrere bzw. unterschiedliche Zertifikate für die verschiedenen Einsatzbereiche.

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.

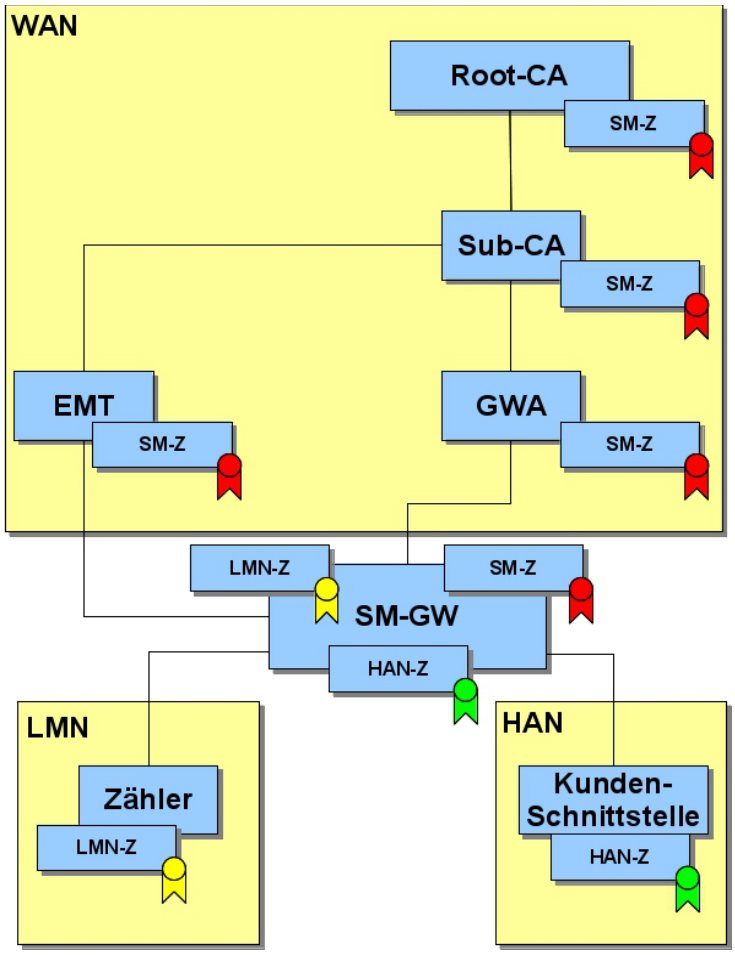


Abbildung 4: Die Zertifikatstypen der SMGW Infrastruktur (ohne CRL-Verbindungen)

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. Des Weiteren werden die Zertifikate zur Absicherung der Kommunikation über die Web-Service-Schnittstelle und mit den Verzeichnisdiensten verwendet. Die Zertifikate müssen zur Absicherung der Kommunikation zwischen den Endnutzern verwendet werden.

Diese Zertifikate sind Gegenstand dieser Spezifikation.

2.6.2 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 [16] festgelegt. Die zugehörigen kryptografischen Anforderungen werden in Abschnitt 6 in [21] definiert.

2.6.3 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 [16] 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 [5].

- Das Zertifikatsprofil wird in Anhang A spezifiziert.
- Der Signaturalgorithmus, die Domain-Parameter und Schlüssellängen, die in den Zertifikaten aktuell zu verwenden sind, werden von [21] 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 [21] 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 der SM-PKI Policy [13] sowie in [21] 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 10 gibt die Zertifikatslaufzeiten sowie die maximalen Verwendungszeiten der zugehörigen privaten Schlüssel der Root-Zertifikate verbindlich vor.

Zertifikat	***Gültigkeitszeit***	***Private Key Usage Period***
Root-Zertifikat	8 Jahre	3 Jahre
Root-Link-Zertifikat	Bis zum Ablauf des alten[1] Root-Zertifikats	Bis zum Ablauf der Private Key Usage Period des neuen[2] Root-Zertifikats

Tabelle 10: Zertifikatslaufzeiten der Root-CA-Zertifikate

Die folgende Tabelle 11 gibt die Zertifikatslaufzeiten der übrigen Zertifikate verbindlich vor.

1 Hierunter ist das Root-Zertifikat zu verstehen, mit dem das Link-Zertifikat verifiziert werden kann, d.h. mit dessen zugehörigem privatem Schlüssel das Link-Zertifikat signiert ist.
2 Hierunter ist das Root-Zertifikat zu verstehen, das denselben Public Key enthält.

Zertifikat	*Gültigkeitszeit*
Root-CRL-Signer-Zertifikat	4 Jahre
Root-TLS-Signer-Zertifikat	4 Jahre
Sub-CA-Zertifikat	5 Jahre
TLS-Zertifikate der Root-CA und der Sub-CA	2 Jahre
GWA-Zertifikate (TLS/Sign/Enc)	3 Jahre
Andere Endnutzerzertifikate (TLS/Sign/Enc)	2 Jahre

Tabelle 11: Zertifikatslaufzeiten der übrigen Zertifikate

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. Entsprechend dürfen ausschließlich gültige Zertifikate in der SMGW-Infrastruktur (siehe [16]) genutzt werden.

Die Zertifikatsvalidierung muss nach dem Schalenmodell gemäß [5] (Cert Path Validation) erfolgen, d. h. insbesondere folgende Prüfungen sind jeweils bei allen im Zertifizierungspfad enthaltenen Zertifikaten (bis zum Vertrauensanker) durchzuführen:

- Signaturprüfung
  - Die Signatur des jeweiligen Zertifikats muss korrekt sein.
- Prüfung des korrekten Gültigkeitszeitraums
  - Der Prüfungszeitpunkt muss im Gültigkeitszeitraum des jeweiligen Zertifikats liegen.
- Sperrlistenprüfung (vgl. Abschnitt 4.2.2)
  - Das jeweilige Zertifikat darf nicht in der aktuellen Sperrlisten der ausstellenden CA enthalten sein.
- Prüfung des korrekten Ausstellers
- Prüfung der korrekten Verwendungsart (Key-Usage Validation und Extended Key Usage Validation)
  - Die Verwendungsart des privaten Schlüssels des jeweiligen Zertifikats muss korrekt sein.

Die Signaturprüfung beinhaltet die Prüfung etwaiger Link-Zertifikate.

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-Zertifikats darf das zugehörige neue selbst-signierte Root-Zertifikat als Vertrauensanker für Zertifikatsvalidierungen verwendet werden.

3.3.2 Spezialfall Zertifikatsvalidierung beim 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 [17]. Mit diesem Vertrauensanker kann das SMGW eine Kettenprüfung von Zertifikaten (ohne Sperrlistenprüfung, siehe Abschnitt 4.2.2.2) durchführen. Grundsätzlich können mehrere Root-Zertifikate parallel gültig sein (siehe Abbildung 3). Das Aufbringen neuer Root-Zertifikate wird in [17] und [16] spezifiziert.

Die Konfiguration des SMGWs auf einen GWA (beinhaltet das Einbringen von GWA-Zertifikaten in das SMGW) ist in [16] spezifiziert.

3.3.2.2 Installation von Endnutzer-Zertifikaten auf dem SMGW

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.

Nachfolgend wird beschrieben, welche Aufgaben das SMGW im Kontext der Zertifikatsvalidierung im Zusammenarbeit mit dem GWA bei der Installation eines Zertifikats auf dem SMGW durchführen muss:

- Vor der Installation eines Zertifikats auf einem SMGW muss dieses vom GWA gemäß Abschnitt 3.3.1 geprüft werden.
- Der GWA sendet das zu installierende Zertifikat mit der gesamten Zertifikatskette bis zum aktuellen, auf dem SMGW vorhandenen Vertrauensanker, inklusive möglicher Link-Zertifikate an das SMGW.
- Das SMGW prüft die gesamte Zertifikatskette gemäß Abschnitt 3.3.1 mit dem Vertrauensanker auf seinem Sicherheitsmodul ohne Sperrlistenprüfung. Zusätzlich muss verifiziert werden, dass das Ablaufdatum des jeweiligen Zertifikats innerhalb des Gültigkeitszeitraums des übergeordneten Zertifikats liegt.

3.3.2.3 Zertifikatsvalidierung im Betrieb

Nach der Installation eines Endnutzer-Zertifikats erfolgt die Zertifikatsvalidierung von Endnutzer-Zertifikaten durch das SMGW gemäß den Anforderungen aus [16].

Die regelmäßige Prüfung des Sperrstatus muss durch den GWA gemäß der Vorgaben aus Kapitel 4.2.2.2 erfolgen.

3.4 Zertifikatsmanagement

In diesem Kapitel wird das Zertifikatsmanagement spezifiziert. Dieses umfasst die Beantragung, Ausstellung und Verteilung von Zertifikaten und die hierbei zu verwendenden Datenformate und Übertragungsprotokolle.

3.4.1 Beantragung von Zertifikaten

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.

Bevor eine Beantragung von Zertifikaten erfolgen kann, muss sich der PKI-Teilnehmer zunächst bei seiner CA registrieren. Die Registrierungsprozesse sind in der SM-PKI Policy [13] definiert.

Die technische Beantragung von Zertifikaten wird in Paketen (Zertifikatsrequest-Pakete, siehe Anhang C), organisiert. Die Zertifikatsrequest-Pakete können (je nach Hierarchieebene in der SM-PKI) aus einem oder mehreren Zertifikatsrequests (etwa für die verschiedenen Zertifikatstypen) bestehen. Auf Endnutzer-Ebene wird immer ein Zertifikatstripel beantragt. Entsprechend werden die benötigten Zertifikate für die verschiedenen benötigten Verwendungszwecke (TLS-, Verschlüsselungs- und Signaturzertifikat) parallel beantragt und ausgestellt.

3.4.1.1 Authentizitätsnachweis bei Zertifikatsrequests

Ein Zertifikatsrequest-Paket beinhaltet immer einen oder mehrere Zertifikatsrequests. Um die Authentizität eines Zertifikatsrequest-Pakets technisch nachzuweisen, sind je nach Art der Beantragung verschiedene Signaturen vorgesehen (siehe nachfolgende Unterkapitel):

- Innere Signatur

  Jeder Zertifikatsrequest in einem Zertifikatsrequest-Paket ist mit dem privaten Schlüssel zum im Zertifikatsrequest enthaltenen öffentlichen Schlüssel signiert (innere Signatur), um den Besitz des jeweiligen neuen privaten Schlüssels nachzuweisen.

- Äußere Signatur

  Mit dieser Signatur wird nachgewiesen, dass der beantragende PKI-Teilnehmer selbst im Besitz eines noch gültigen Schlüssels ist. Dies ist für die Durchführung eines automatisierten Folgeantrags (vgl. [13]) erforderlich.

- Autorisierungssignatur

  Wenn eine dritte Partei, die bei einer Sub-CA registriert ist, stellvertretend Zertifikate beantragt, muss diese das Zertifikatsrequest-Paket autorisieren. Dies bei ist der Beantragung von Zertifikaten für das SMGW durch den GWH und den GWA der Fall.

3.4.1.2 Initiale Beantragung und Folgeantrag (außer SMGW)

Bei der Beantragung von Zertifikaten werden zwei Arten unterschieden, die von CA unterschiedlich authentisiert werden müssen:

- Initiale Beantragung: Der PKI-Teilnehmer besitzt noch kein Signaturzertifikat oder kein gültiges Signaturzertifikat aus der SM-PKI.

  Die CP [13] definiert, wie die Identifizierung und Authentifizierung bei der initialen Beantragung durchgeführt werden muss. Technisch muss ein Zertifikatsrequest-Paket zur initialen Beantragung grundsätzlich mit dem Object Identifier `id-CertReqMsgs` gekennzeichnet werden und enthält nur eine innere Signatur je Zertifikatsrequest (siehe Abschnitt C.2; Ausnahme SMGW, siehe Abschnitt 3.4.1.3).

- Folgeantrag: Der PKI-Teilnehmer besitzt bereits ein Signatur-Zertifikat aus der SM-PKI, welches zum Zeitpunkt der Verifikation des Requests durch die CA noch gültig ist.

Die Identifizierung und Authentifizierung eines Zertifikatsantrags erfolgt im Falle eines Folgerequests über die äußere Signatur des Zertifikatsrequest-Pakets (äußere Signatur, siehe Abschnitt C.1), die mit dem privaten Schlüssel zum Signaturzertifikat des Antragstellers erstellt werden muss. Technisch muss ein Zertifikatsrequest-Paket bei einem Folgeantrag mit dem Object Identifier `id-CertReq-Msgs-with-outer-Signature` gekennzeichnet werden (siehe Abschnitt C.1; Ausnahme SMGW, siehe Abschnitt 3.4.1.3). Des Weiteren enthält jeder Zertifikatsrequest im Zertifikatsrequest-Paktet eine innere Signatur. Sollte der PKI-Teilnehmer sein Signaturzertifikat nicht mehr nutzen können, kann das Request-Paket alternativ mit dem privaten Schlüssel zum TLS-Zertifikat des Antragsstellers signiert werden. Details zu diesen "nicht routinemäßigen Folgeanträgen" sind in [13] definiert.

3.4.1.3 Sonderfall SMGW

Das SMGW kann selbst keine Zertifikate beantragen. Entsprechend beantragt eine dritte Partei stellvertretend für das SMGW die Zertifikate.

Hierbei wird die Autorisierungssignatur (siehe Abschnitt 3.4.1.1) verwendet. Technisch muss ein Zertifikatsrequest-Paket für ein SMGW entsprechend mit dem Object Identifier `id-authorized-CertReqMsgs` gekennzeichnet werden (siehe Abschnitt C.1).

Generell wird wie nachfolgend beschrieben, zwischen der initialen Beantragung von Gütesiegel-Zertifikaten und der Beantragung von Wirk-Zertifikaten unterschieden.

Beantragung von Gütesiegel-Zertifikaten

Bei der Produktion beantragt der GWH initial die Gütesiegel-Zertifikate für das SMGW. Die Autorisierungssignatur des Zertifikatsrequest-Pakets wird mit dem Signatur-Schlüssel des GWH erstellt. In diesem Fall muss im `EncapsulatedContent` der `contentType id-CertReqMsgs` enthalten sein. Des Weiteren enthalten die Zertifikatsrequests im Zertifikatsrequest-Pakets jeweils eine innere Signatur (siehe Abschnitt 3.4.1.1).

Übertragung der technischen Verantwortlichkeit

Bevor die Gütesiegel-Zertifikate eines SMGW durch Wirk-Zertifikate ersetzt werden, kann die technische Verantwortlichkeit für diese Gütesiegelzertifikate formal an den zuständigen GWA übertragen werden. Der Prozess ist in der SM-PKI Policy [13] beschrieben.

Zur Übertragung der technischen Verantwortlichkeit nennt der GWH die betreffenden SMGWs (identifiziert durch ihre eindeutige herstellerübergreifende Identifikationsnummer) und den verantwortlichen GWA (identifiziert durch die Common Names von Subject-DN und Issuer-DN aus C_{SIG}(GWA) an die Sub-CA, die die Gütesiegelzertifikate ausgestellt hat. Der GWH muss die Datenstruktur für die Übertragung der technischen Verantwortlichkeit signieren. Technisch muss eine solche Übertragung mit dem Object Identifier `id-signedRevReqs` gekennzeichnet werden (iehe Abschnitt C.1).

Beantragung von Wirk-Zertifikaten

Die Beantragung der Wirk-Zertifikate für ein SMGW erfolgt im Wirk-Betrieb durch den GWA. In diesem Fall muss im EncapsulatedContent der content type `id-CertReq-Msgs-with-outer-Signature` enthalten sein.

Entsprechend enthält das Zertifikatsrequest-Paket folgende Signaturen:

- Äußere Signatur

  Erstellt mit dem Signaturschlüssel des SMGW. Hierüber wird nachgewiesen, dass es sich um ein authentisches SMGW handelt.

- Autorisierungssignatur

  Erstellt mit dem Signatur-Schlüssel des GWA. Hierüber authentisiert sich der GWA gegenüber der Sub-CA und autorisiert die Zertifikatsrequests des SMGW.

Das Signatur-Zertifikat vom SMGW und vom GWA müssen zum Zeitpunkt der Beantragung noch gültig sein. Des Weiteren enthalten die Zertifikatsrequests im Zertifikatsrequest-Paket jeweils eine innere Signatur (siehe Abschnitt 3.4.1.1).

3.4.2 Ausstellung und Verteilung von Zertifikaten

Eine CA muss nach der Entgegennahme des Zertifikatsrequest-Pakets die Authentizität, wie in der SM-PKI Policy definiert, prüfen (vergl. [13]). Die technische Prüfung der Authentizität erfolgt über Verifikation der Gültigkeit der in Abschnitt 3.4.1.1 beschriebenen Signaturen. Es dürfen ausschließlich Zertifikate für korrekt signierte Zertifikatsrequests bzw. Zertifikatsrequest-Pakete ausgestellt werden.

Ob es sich um einen initialen Antrag oder einen Folgeantrag handelt, erkennt die CA durch die Auswertung der Datenstruktur des Zertifikatsrequest-Pakets.

Nach erfolgreicher Prüfung der Authentizität eines Zertifikatsrequest-Pakets stellt eine CA für die darin enthaltenen Zertifikatsequests die entsprechenden Zertifikate aus. Hierzu muss ein noch gültiger privater Signaturschlüssel der CA verwendet werden. Ein von einer CA ausgestelltes Zertifikat darf nicht länger gültig sein als das Zertifikat zum Signaturschlüssel der CA. Eine CA muss bei der Ausstellung eines Zertifikats nicht zwingend die Werte aus dem Zertifikatsrequest übernehmen. Eine CA kann Änderungen vornehmen, die aber konform zu den Anforderungen aus diesem Dokument und der SM-PKI Policy [13] sein müssen.

Auf Endnutzer-Ebene wird immer ein Zertifikatstripel ausgestellt (siehe Abschnitt 2.3.2). Die Inhalte im Subject-DN der Zertifikate eines Zertifikatstripels sind identisch und eindeutig. Wesentlich sind hier der Name (Feld Common Name, CN) und die Sequenznummer (Feld SERIALNUMBER) unterhalb einer Sub-CA.

Die Zertifikate müssen dann dem Antragssteller zugestellt werden (siehe Protokolle in Abschnitt 3.4.3). Überdies müssen alle ausgestellten Zertifikat einer CA im zugehörigen Verzeichnisdienst veröffentlicht werden (siehe Kapitel 3.5). Bei einem SMGW ist dann der GWH bzw. GWA für die Installation der auf Gültigkeit geprüften Zertifikate verantwortlich (vgl. SM-PKI Policy [13]).

Die folgende Abbildung 5 zeigt beispielhaft die Organisation der Zertifikate eines SMGW (Zertifikatstripel) über den Subject-DN beim Ausstellen des Zertifikats und der Speicherung im Verzeichnisdienst.

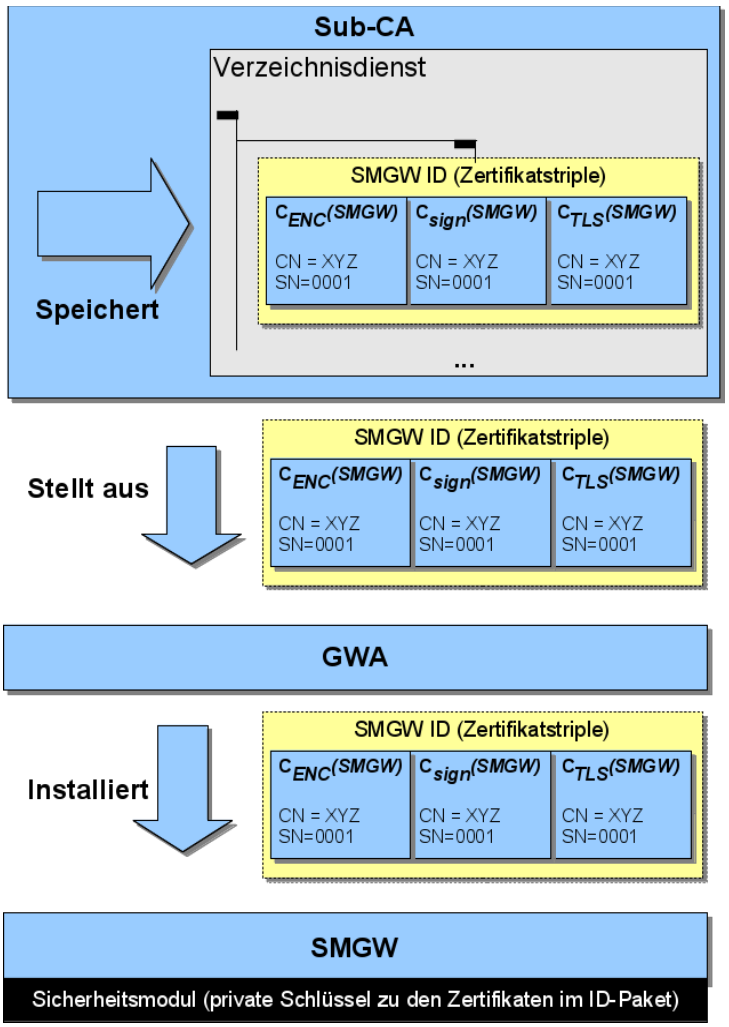


Abbildung 5: SMGW Zertifikatsmanagement (Beispiel)

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 SM-PKI Policy [13] festgelegt werden.

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

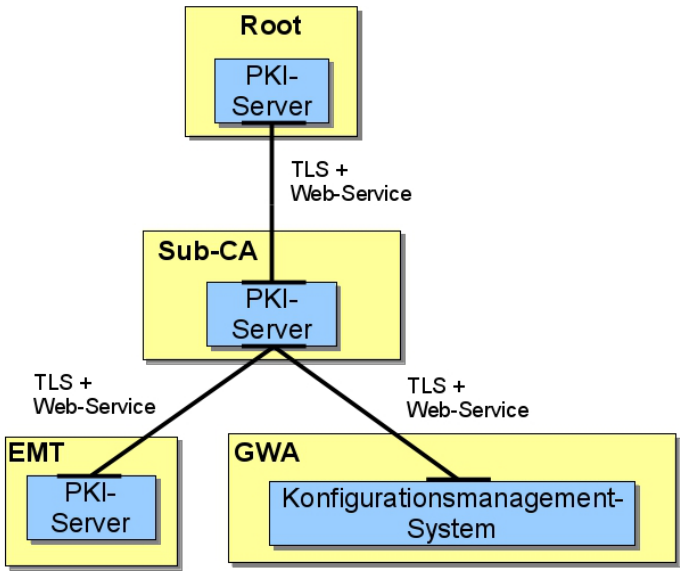


Abbildung 6: Web-Service Kommunikation in der SM-PKI

Die Kommunikationsverbindungen erfordern eine beidseitige TLS-gesicherte Verbindung, um die Authentizität und Vertraulichkeit der Kommunikation sicherzustellen. Hierzu werden die TLS-Zertifikate aus der SM-PKI eingesetzt.

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 SMGW selbst erstellt. Der Request wird vom GWA abgerufen, geprüft und signiert (Autorisierungssignatur) und per Web Service durch den GWA an die ausgewählte 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. [15]). 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 [4] basierenden Verzeichnisdienst betreiben. In dem Verzeichnisdienst muss die jeweilige CA alle Zertifikate veröffentlichen, die von ihr ausgestellt worden sind (siehe Tabelle 12).

Der Zugriff auf den Verzeichnisdienst muss auf Teilnehmer der SM-PKI beschränkt werden. Dies wird über eine zertifikatsbasierte TLS-Authentisierung am Verzeichnisdienst mit SM-PKI-Zertifikaten sichergestellt.

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).

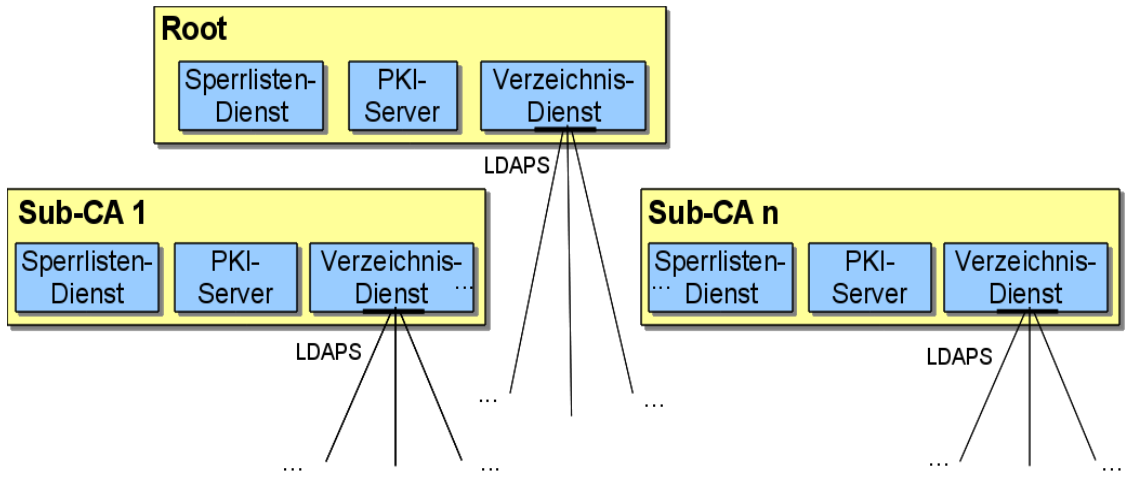


Abbildung 7: Verzeichnisdienste in der SM-PKI

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

Betreiber	***Inhalt***	***Protokoll / Adresse***
Root-CA	*C(Root)*, *Link-C(Root)*, $C_{CRL\text{-}S}$*(Root)*, $C_{TLS\text{-}S}$*(Root)*, *C(Sub-CA)*, $C_{TLS,Root}$*(Sub-CA)*	- LDAPS-Protokoll / Port 636 - TLS-Authentisierung mit SM-PKI-Zertifikaten - Informationen zur Adresse sind in [13] definiert
Sub-CA	*C(Sub-CA)*, C_{TLS}*(ENu)*, C_{Enc}*(ENu)*, C_{Sign}*(ENu)*	- LDAPS-Protokoll / Port 636 - TLS-Authentisierung mit SM-PKI-Zertifikaten - URL in Sub-CA-CP definiert

Tabelle 12: Verzeichnisdienste in der SM-PKI

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

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

Zusätzlich sollte eine CA über einen öffentlichen Verzeichnisdienst (ldap/389) bzw. über einen öffentlichen Bereich auch ihre CA-Zertifikate und ihre Sperrliste zur Verfügung stellen.

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 Zertifikate eines autorisierten Marktteilnehmers bzw. des GWA, die in naher Zukunft ablaufen, so muss er auf dem SMGW rechtzeitig entsprechende neue Zertifikate installieren. Die entsprechenden Zertifikate bezieht der GWA über den Verzeichnisdienst der zuständigen CA.

Die Anforderungen an die Konfiguration und Verwendung der Zertifikate für die Kommunikation des SMGW werden in [16] und [19] definiert. Abgelaufene Zertifikate dürfen vom SMGW nicht mehr verwendet werden.

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 [5]. 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 [21] 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 SM-PKI Policy [13] definiert.

4.2.1 Veröffentlichung von Sperrlisten

Die in der folgenden Tabelle 13 dargestellten Sperrlisten sind in der SM-PKI vorhanden.

Sperrliste	*Herausgeber*	*Inhalt*	*CRL-Typ*	*Signatur-Schlüssel*
Root-CA-CRL	Root-CA	Gesperrte Root- und Sub-CA-Zertifikate.	Indirekt	Privater Schlüssel zum aktuellem $C_{CRL\text{-}S}(Root)$.
Root-TLS-CRL	Root-CA	Gesperrte TLS-Zertifikate, die von der Root ausgestellt wurden.	Direkt	Privater Schlüssel zu $C_{TLS\text{-}S}(Root)$.
Sub-CA-CRL	Jede Sub-CA erstellt eine eigene CRL.	Gesperrte Endnutzer-Zertifikate (im Verwaltungsbereich der jeweiligen Sub-CA).	Direkt	Privater Schlüssel zu *C(Sub-CA)*.

Tabelle 13: Die Sperrlisten in der SM-PKI

Die Root-CA und die Sub-CAs müssen deren Sperrlisten wahlweise über HTTP [1] oder LDAP [4] zum freien Herunterladen bereitstellen (siehe Abbildung 8). Hierbei müssen die in [13] definierten Zeiten eingehalten werden. Damit die Authentizität der jeweiligen Sperrliste überprüft werden kann, müssen die hierfür 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.

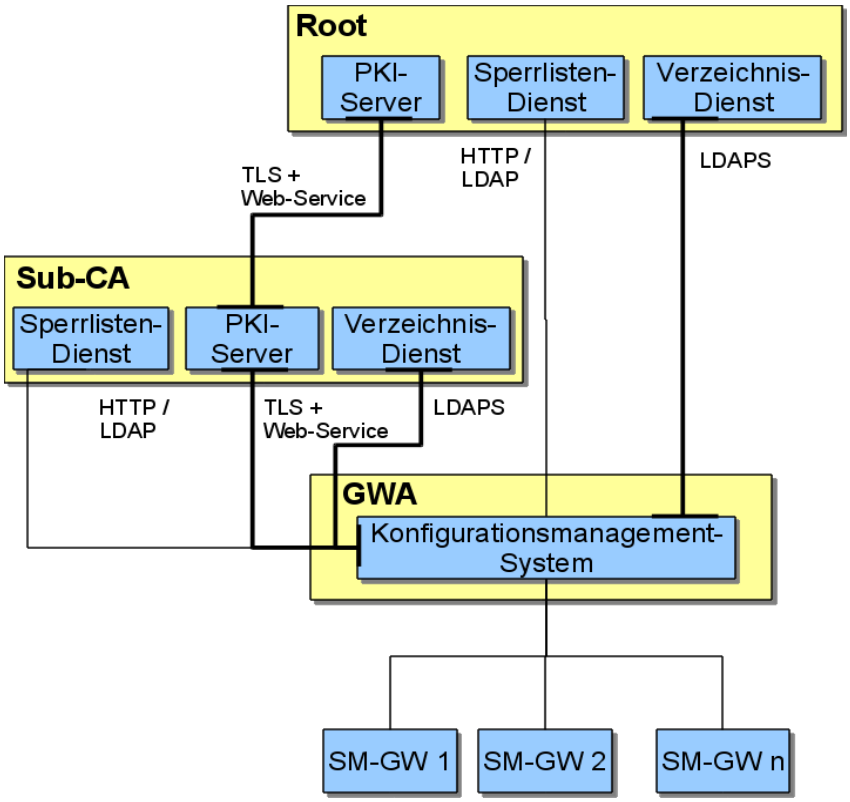


Abbildung 8: Infrastruktur Sperrmanagement (Beispiel)

4.2.1.1 Sperrlisten der Root-CA

Die Root muss einen Sperrlisten-Dienst betreiben. Von der Root werden zwei verschiedene Sperrlisten ausgegeben.

Die Root-TLS-CRL enthält ausschließlich die gesperrten TLS-Zertifikate, die von der Root für die Root selbst und für die Sub-CAs ausgestellt werden. Die Sperrliste wird mit dem Root-TLS-Signer signiert, mit dem auch diese TLS-Zertifikate ausgestellt werden. Entsprechend handelt es sich um eine direkte Sperrliste.

Die Root-CA-CRL enthält die gesperrten Sub-CA- und Root-Zertifikate (außer den TLS-Zertifikaten, die mit dem Root-TLS-Signer ausgesellt wurden). Die Sperrliste wird nicht mit dem Root-CA Schlüssel, sondern mit einem separaten Schlüsselpaar (Root-CRL-Signer), signiert. Entsprechend handelt es sich um eine indirekte Sperrliste. Mit Erstellung eines neuen Root-CA Zertifikates wird auch ein neues CRL-S Zertifikat erstellt. Mit dem zu diesem Zertifikat gehörenden privaten Schlüssel werden ab diesem Zeitpunkt die CRLs der/aller Root-CAs signiert.

Die Root-CA muss in den in [13] 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.

4.2.1.2 Sperrlisten der Sub-CA

Eine 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 ausgegebenen Zertifikate überprüft werden.

Eine Sperrliste muss mit dem privaten Schlüssel zum aktuell gültigen Sub-CA Zertifikat signiert werden, mit dem auch die Zertifikate ausgestellt werden. Sind bei einen Zertifikatswechsel vorübergehend mehrere Sub-CA Zertifikate parallel gültig (Überlappungszeit, siehe Abschnitt 2.4), ist jedem der gültigen Sub-CA-Zertifikate jeweils eine Sperrliste zugeordnet. Dabei kann die Sperrliste immer nur die Zertifikate enthalten, die mit gleichem Schlüssel ausgestellt wurden, mit dem auch die Sperrliste signiert ist.

Eine Sub-CA muss in den in [13] 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.

4.2.2 Sperrlistenvalidierung

4.2.2.1 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 gemäß [5]) Dabei müssen alle im Zertifizierungspfad enthaltenen Zertifikate zum Prüfungzeitpunkt gültig sein (vgl. Abschnitt 3.3.1).
  - Beinhaltet die Prüfung etwaiger Link-Zertifikate
  - Bei Signaturprüfung einer Sub-CA-CRL muss *C(Sub-CA)* auch gegen die Root-CA-CRL geprüft werden.
- Prüfung der Aktualität der Sperrliste.

4.2.2.2 Sonderfall SMGW

Für das SMGW ist die Möglichkeit einer eigenständigen Sperrlistenprüfung nicht gegeben. Diese Aufgabe muss daher vom Gateway-Administrator (GWA) übernommen werden (vgl. [15]). Die Prüfung der relevanten Sperrlisten erfolgt in diesem Falle in regelmäßigen Abständen gemäß den Vorgaben aus[13] bzw. aus konkretem Anlass[3]. Die Validierung durch den GWA muss dabei erfolgen, wie in Kapitel 4.2.2.1 beschrieben. Befindet sich in der jeweiligen Sperrliste ein Zertifikat eines EMTs, welches 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 aller verwalteten SMGWs und die darin installierten Zertifikate registriert sind. Bei der Prüfung der relevanten Sperrlisten mit 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. [15]).

3 Die konkreten Anlässe, die eine außerplanmäßige Sperrlistenprüfung durch den GWA notwendig machen, sind in [13] definiert.

4.3 Sperrung von Zertifikaten

Details zum Ablauf von Sperrungen und den dazu notwendigen Berechtigungen sind [13] zu entnehmen.

Sperrungen werden wirksam durch Aufnahme in die CRL der entsprechenden CA. Hierbei wird zwischen der im Folgenden beschriebenen Suspendierung (spezielle Art einer Sperrung, nur für das SMGW) und normalen Sperrungen unterschieden. Mit Ausnahme der Suspendierung dürfen Sperrungen nicht aufgehoben werden.

Die Sperrung von Endnutzer-Zertifikaten erfolgt analog zur Beantragung in Paketen. Bei der Beantragung von Zertifikatsperrungen müssen durch den Beantragenden der Sperrung daher stets alle Zertifikate des zu sperrenden Pakets im Revocationrequest angegeben werden (vgl. Datenformat der Revocationrequests in Abschnitt C.3.1).

Spezialfall bei der Sperrung von SMGW-Zertifikaten:

Eine Sperrung von SMGW-Zertifikaten kann durch den dazu berechtigten Teilnehmer der SM-PKI (GWA/GWH) über automatisierte Schnittstellen (siehe Anhang D) veranlasst werden. Hierbei muss für den Revocationrequest das Datenformat aus Anhang C verwendet werden. Der Revocationrequest ist hierbei vom Sperrberechtigten zu signieren.

Es ist neben der Sperrung möglich, eine Suspendierung (Spezialfall der Sperrung, siehe [13]) der SMGW-Zertifikate über diesen Weg vornehmen zu lassen. Gütesiegelzertifikate können nicht suspendiert werden.

Nach einer automatisiert durchgeführten Sperrung oder Suspendierung von SMGW-Zertifikaten muss die entsprechende Sub-CA eine anlassbezogene Erneuerung der Sperrliste entsprechend [13] vornehmen.

Bis zum Ablauf des in der SM-PKI Policy [13] definierten Zeitraums kann eine Suspendierung wieder zurückgenommen werden, damit neue Zertifikate für das SMGW beantragt und installiert werden können.

Nach dem Ablauf dieses Zeitraums endet eine Suspendierung automatisch. Dabei entfällt die in Anhang B.2 geforderte Kennzeichnung mit reasonCode = „certificateHold“ gemäß [5]. Entsprechend ist eine Rücknahme der Sperrung nicht mehr möglich.

Die automatische endgültige Sperrung einmal suspendierter Zertifikate nach Fristablauf erfolgt ohne weitere Veranlassung selbst dann, wenn die Suspendierung zwischenzeitlich wieder zurückgenommen wurde. Um die Suspendierung eines SMGW dauerhaft aufzuheben, muss dieses vor Fristablauf neue Zertifikate erhalten. Der entsprechende Ablauf ist in der SM-PKI Policy [13] definiert.

Die Sperrung einzelner Zertifikate eines Tripels ist gemäß [12] nicht zulässig. Ein Sperrauftrag für ein beliebiges Zertifikat eines Tripels muss stets zur Sperrung aller Zertifikate des Tripels führen. Auf der Sperrliste muss für die Zertifikate eines Tripels derselbe Sperrgrund und ggf. dasselbe InvalidityDate erscheinen. Wurden im Sperrauftrag unterschiedliche Sperrgründe genannt, so muss der Sperrauftrag dennoch angenommen werden, in diesem Fall werden die Zertifikate des Tripels mit dem reasonCode `unspecified` gemäß [5] gesperrt. Wurden im Sperrauftrag unterschiedliche InvalidityDates genannt, so muss der Sperrauftrag ebenfalls angenommen werden und es wird für alle Zertifikate des Tripels das früheste genannte InvalidityDate in die Sperrliste übernommen.

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 [5] die folgende Struktur:

Zertifikatsfeld	*Referenz in [5]*	*m/x/c/o*	*Wert*
`Certificate`	4.1.1	m	
`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 [5]*	*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.
`SubjectUniqueID`	4.1.2.8	x	Entfällt.

Zertifikatsfeld	*Referenz in [5]*	*m/x/c/o*	*Wert*
`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 [5] 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 [21], Kapitel 3, verbindlich vorgegeben. Das Feld `parameters` bleibt leer.

A.1.2 SubjectPublicKeyInfo

Das Feld `SubjectPublicKeyInfo` muss folgende Struktur besitzen (siehe [6]) :

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

Die OID in `algorithm` muss den Wert 1.2.840.10045.2.1 (`id-ecPublicKey`) haben. Im Feld `EC-Parameters` ist gemäß [6] 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 [21], Kapitel 3, 4 und 7.

A.1.3 Issuer und Subject

Das Namensschemata für die verschiedenen PKI-Teilnehmer ist in der SM-PKI Policy [13] definiert. Dieses muss in der angegebenen Feld-Reihenfolge verwendet werden. Je nach dem, ob es sich bei der PKI-Instanz um einen Zertifikatsaussteller oder -inhaber handelt, müssen die Werte im Feld `Issuer` bzw. `Subject` eingetragen werden.

Link-Zertifikate unterscheiden sich von Root-Zertifikaten bzgl. `Issuer` und `Subject` dadurch, dass der `Issuer-DN` identisch mit `Subject-DN` des vorherigen Root-Zertifikats ist (Signatur erfolgt mit privatem Schlüssel zum vorherigen Root-CA Zertifikat).

Auf Endnutzer-Ebene wird zusätzlich zwischen den Rollen SMGW, GWA, GWH und EMT unterschieden. Ferner wird dort zwischen Signatur-, TLS- und Verschlüsselungszertifikat getrennt. Diese Unterscheidung erfolgt nicht über das Feld `Subject`, sondern über die Extension `KeyUsage` (siehe Abschnitt A.2).

A.2 Extensions

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

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

Tabelle 16: Zertifikats-Extensions für CA-Zertifikate

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

Tabelle 17: Zertifikats-Extensions für die TLS-Zertifikate der CAs

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

Tabelle 18: 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	x	x	x
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 19: 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 19

5 Das Vorhandensein der Extension ist abhängig vom Zertifikatsinhaber, siehe Tabelle 18.

- Wert: Der `KeyIdentifier` wird mit Methode 1 gemäß [5], 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 [5], 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` (spezifiziert in [5], 4.2.1.1) definiert, 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-S}(Root)***	***C_{TLS-S}(Root)***	***C(Sub-CA)***
Gesetzte Bits	KeyCertSign	cRLSign	KeyCertSign cRLSign	KeyCertSign, cRLSign

Tabelle 20: Belegung KeyUsage-Extension für CA-Zertifikate

Zertifikat	***C_{TLS} (Root)***	***C_{TLS}(Sub-CA)***	***$C_{TLS,Root}$(Sub-CA)***
Gesetzte Bits	DigitalSignature	DigitalSignature	DigitalSignature

Tabelle 21: Belegung KeyUsage-Extension für die TLS-Zertifikate der CAs

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

Tabelle 22: Belegung KeyUsage-Extension für Endnutzer-Zertifikate

4. `PrivateKeyUsagePeriod`

- Extension-ID (OID): 2.5.29.16
- Kritisch: Nein
- Beschreibung: Die Extension `PrivateKeyUsagePeriod` (spezifiziert in [5], 4.2.1.1) definiert die Gültigkeitszeit des zum Zertifikat gehörenden privaten Schlüssels. Das Vorhandensein dieser Extension wird in den Tabellen 16 bis 19 definiert.
- 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 [5], 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 [5], 4.2.1.6) enthält weitergehende Informationen zum Inhaber (Subject) – Namen.
   - Wert: Das Feld muss den Alternativnamen des Zertifikatsinhabers gemäß [13] enthalten, sofern dieser vorhanden ist.
7. `IssuerAltName`
   - Extension-ID (OID): 2.5.29.18
   - Kritisch: Nein
   - Beschreibung: Diese Extension (spezifiziert in [5], 4.2.1.7) enthält weitergehende Informationen zum Aussteller (Issuer) - Namen.
   - Wert: `IssuerAltName` muss den Alternativnamen des Zertifikatsausstellers gemäß [13] enthalten.
8. `BasicConstraints`
   - Extension-ID (OID): 2.5.29.19
   - Kritisch: Ja
   - Beschreibung: Diese Extension (spezifiziert in [5], 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)*	*Link-C(Root)*	*$C_{CRL\text{-}S}$(Root)*	*$C_{TLS\text{-}S}$(Root)*	*C(Sub-CA)*
`cA`	TRUE	FALSE	FALSE	TRUE	TRUE
`path-lengthCon-straint`	1	/	/	0	0

Tabelle 23: Belegung Basic-Constraints-Extension für CAs

Zertifikat	*C_{TLS} (Root)*	*C_{TLS}(Sub-CA)*	*$C_{TLS,Root}$(Sub-CA)*
`cA`	FALSE	FALSE	FALSE
`pathlength-Constraint`	/	/	/

Tabelle 24: Belegung Basic-Constraints-Extension für die TLS-Zertifikate der CAs

Zertifikat	*C_{TLS}(ENu)*	*C_{Enc}(ENu)*	*C_{Sign}(ENu)*
`cA`	FALSE	FALSE	FALSE
`pathlength-Constraint`	/	/	/

Tabelle 25: Belegung Basic-Constraints-Extension für Endnutzer

9. `ExtendedKeyUsage`

- Extension-ID (OID): 2.5.29.37
- Kritisch: Ja
- Beschreibung: Dieser Extension (spezifiziert in [5], 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} (Root)*	*C_{TLS} (Sub-CA)*	*$C_{TLS,Root}$ (Sub-CA)*	*C_{TLS} (GWA)*	*C_{TLS} (GWH)*	*C_{TLS} (EMT)*	*C_{TLS} (SMGW)*
TLS-Web-Server-Authentifikation (1.3.6.1.5.5.7.3.1)	m	m	m	m	m	m	-
TLS-Web-Client-Authentifikation (1.3.6.1.5.5.7.3.2)	m	m	m	m	m	m	m

Tabelle 26: Belegung ExtendedKeyUsage-Extension

10. `CRLDistributionPoints`

- Extension-ID (OID): 2.5.29.31
- Kritisch: Nein
- Beschreibung: Diese Extension (spezifiziert in [5], 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 ausstellenden CA im Zertifikat angegeben werden. Veröffentlicht die ausstellende CA auch Delta-CRLs, so sollte hierfür auch mindestens ein Verteilungspunkt angegeben werden.
- Werte: Jeder Eintrag in der Sequenz `CRLDistributionPoints` besteht 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. Wenn dieselbe CRL über verschiedene Zugriffsmechanismen bereitgestellt wird (z.B. via HTTP und via LDAP), so müssen sämtliche Zugriffsmechanismen als GeneralNames in einem einzigen DistributionPoint kodiert werden. Wird im

Zertifikat auf eine indirekte Sperrliste referenziert (vgl. Tabelle 13), so muss der Subject-DN des Zertifikats im Feld `CRLIssuer` angeben werden, mit dem die Sperrliste aussgestellt wird. Das Feld `reason` wird nicht verwendet.

Zertifikat	*C(Sub-CA)*	*$C_{TLS\text{-}S}$(Root)*	*C(Root)*
`CRLIssuer`	Subject-DN von $C_{CRL\text{-}S}$ (Root)	Subject-DN von $C_{CRL\text{-}S}$ (Root)	Subject-DN von $C_{CRL\text{-}S}$ (Root)

Tabelle 27: Belegung CRLIssuer

B CRL-Profil

Die CRLs, die in der SM-PKI verwendet werden (siehe Tabelle 13), sind X.509-CRLs gemäß [5] 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 [21] zu finden. Das Feld `nextUpdate` muss vorhanden sein und enthält das Datum, an dem die nächste CRL spätestens veröffentlicht wird. Hierbei sind die zeitlichen Vorgaben aus [13] 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 Delta-CRL-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.
- `IssuingDistributionPoint`
  - Extension-ID (OID): 2.5.29.28
  - Kritisch: Ja

- Beschreibung: Diese Extension muss bei einer Root-CA-CRL (siehe Tabelle 13) vorhanden sein. Hierbei muss das Flag `indirectCRL` in der Extension entsprechend auf TRUE gesetzt werden. Ansonsten darf die Extension nicht vorhanden sein.

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 kann verwendet werden, um einen bestimmten Grund für die Sperrung anzugeben. Sie muss verwendet werden, wenn ein SMGW-Zertifikat suspendiert wird (reasonCode „certificateHold“, siehe Kapitel 4.3).
- `InvalidityDate`
  - Extension-ID (OID): 2.5.29.24
  - Kritisch: Nein
  - Beschreibung: Die Extension kann vorhanden sein. Wenn zum Sperrzeitpunkt ein exakter - und vom Zeitpunkt der Sperrung abweichender - Zeitpunkt bekannt ist, ab dem ein Zertifikat als gesperrt gelten muss, dann ist die Extension zwingend zu nutzen.
- `CertificateIssuer`
  - Extension-ID (OID): 2.5.29.29
  - Kritisch: Nein

    Beschreibung: Diese Entry-Extension muss bei indirekten Sperrlisten verwendet werden (siehe Tabelle 13). Ansonsten darf diese nicht vorhanden sein. In dieser Entry-Extension muss der Wert des Issuer-DN des Zertifikats eingetragen werden, zum dem der Sperreintrag gehört. Dabei muss dieser Eintrag identisch codiert sein mit dem Issuer-DN in zu sperrenden Zertifikat.

Weitere CRL-Entry-Extensions sollen nicht verwendet werden.

C Datenstrukturen für das Zertifikatsmanagement

In diesem Kapitel gelten folgende Begriffsdefinitionen:

- **„Einzelantrag“**: Ein Request für ein einzelnes Zertifikat. Ein Einzelantrag enthält eine „innere Signatur“ gemäß Kapitel 3.4.1.1 und wird stets als Teil eines Zertifikatsrequest-Pakets übertragen. Entspricht der in Kapitel C.2.1 definierten Struktur `CertReqMsg`.
- **„Zertifikatsrequest-Paket“**: Eine Sequenz aus einem oder mehreren Einzelanträgen, womit gemäß Kapitel 3.4.1 mehrere Zertifikate in einem Schritt beantragt werden können. Entspricht der in Kapitel C.2.1 definierten Struktur `CertReqMessages`. Durch die Einbettung in eine signierte Datenstruktur ergibt sich je nach Anwendungsfall ein „signiertes Zertifikatsrequest-Paket“ oder ein „autorisiertes Zertifikatsrequest-Paket“ (siehe Kapitel C.2). Zur Abgrenzung von einem signierten oder autorisierten Zertifikatsrequest-Paket wird diese Struktur auch als „Zertifikatsrequest-Paket ohne äußere Signatur“ bezeichnet. In dieser Form wird es für Initialanträge (außer für SMGW) verwendet.
- **„Signiertes Zertifikatsrequest-Paket“**: Eine signierte Datenstruktur, die ein Zertifikatsrequest-Paket für Folgezertifikate eines bestimmten Zertifikatsnehmers enthält und deren Signatur die in Kapitel 3.4.1.1 definierte „äußere Signatur“ desselben Zertifikatsnehmers ist. Für die Beantragung von SMGW-Folgezertifikaten wird das signierte Zertifikatsrequest-Paket wiederum in eine signierte Datenstruktur eingebettet, um ein autorisiertes Zertifikatsrequest-Paket zu bilden.
- **„Autorisiertes Zertifikatsrequest-Paket“**: Eine signierte Datenstruktur, die entweder ein Zertifikatsrequest-Paket (bei SMGW-Initialanträgen) oder ein signiertes Zertifikatsrequest-Paket (bei SMGW-Folgeanträgen) enthält und deren Signatur die in Kapitel 3.4.1.1 definierte Autorisierungssignatur ist. Autorisierte Zertifikatsrequest-Pakete werden ausschließlich bei Anträgen für SMGW-Zertifikate verwendet. Die Autorisierungssignatur wird im Folgenden auch kurz als „Autorisierung“ bezeichnet.
- **„Revocationrequest”**: Ein Sperrauftrag für sämtliche Zertifikate, die mit einem bestimmten Zertifikatsrequest-Paket beantragt wurden. Durch Einbettung in eine signierte Datenstruktur ergibt sich ein signierter Revocationrequest, siehe Kapitel C.3.1.
- **“Signierter Revocationrequest”**: Eine signierte Datenstruktur, die einen Revocationrequest enthält. Die Signatur ist vom jeweiligen Sperrberechtigten zu leisten.
- **“SMGW-Übertragung”**: Eine Datenstruktur, die zur Übertragung der technischen Verantwortlichkeit für bestimmte SMGWs auf einen bestimmten GWA dient. Durch die Einbettung in eine signierte Datenstruktur ergibt sich eine signierte SMGW-Übertragung, siehe Kapitel C.4.1.
- **“Signierte SMGW-Übertragung”**: Eine signierte Datenstruktur, die eine SMGW-Übertragung enthält. Die Signatur ist vom bisherigen Inhaber der technischen Verantwortlichkeit für die genannten SMGWs zu leisten.

Zur Beantragung von Zertifikaten werden (ggf. signierte und/oder autorisierte) Zertifikatsrequest-Pakete gemäß den Vorgaben von Kapitel 3.4 verwendet. Revocationrequests werden gemäß den Vorgaben von Kapitel 4.3 verwendet. Datensätze zur Übertragung der technischen Verantwortlichkeit für SMGW werden gemäß den Vorgaben von Kapitel 3.4.1.3 verwendet.

Hierfür muss jeweils die folgende Datenstruktur verwendet werden:

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

Als `contentType` wird ein Object Identifier (OID) angegeben. Von diesem ist der Datentyp von `content` abhängig. Die zulässigen Werte für `contentType` und `content` sind der folgenden Tabelle zu entnehmen:

contentType OID	*content-Datentyp*	*Beschreibung*
`id-CertReqMsgs`	`CertReqMessages`	Zertifikatsrequest-Paket (ohne äußere Signatur)
`id-CertReqMsgs-with-outerSignature`	`SignedData`	signiertes Zertifikatsrequest-Paket
`id-authorized-CertReqMsgs`	`SignedData`	autorisiertes Zertifikatsrequest-Paket
`id-signedRevReqs`	`SignedData`	signierter Revocationrequest
`id-signedUpdateDeviceAdmin`	`SignedData`	signierte SMGW-Übertragung

Tabelle 28: ContentInfo und zulässige Werte für den Datentyp

Die in Tabelle 28 genannten OIDs sind wie folgt definiert:

ContentType	*Object Identifier*
id-CertReqMsgs	bsi-de pki(4) x509(1) id-CertRequests(1) 1
id-CertReqMsgs-with-outerSignature	bsi-de pki(4) x509(1) id-CertRequests(1) 2
id-authorized-CertReqMsgs	bsi-de pki(4) x509(1) id-CertRequests(1) 3
id-signedRevReqs	bsi-de pki(4) x509(1) revocation(2) 2
id-signedUpdateDeviceAdmin	bsi-de application(3) Smart-Metering(4) sm-processes(2) 2

Tabelle 29: Für ContentType zu verwendende OIDs

Zur Definition des Datentyps `CertReqMessages` siehe Kapitel C.2.1.

C.1 Datentyp SignedData

Der Datentyp `SignedData` besitzt gemäß [8] folgende Struktur:

```
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 DigestAlgorithmIdentifier,
   signedAttrs [0] IMPLICIT SignedAttributes OPTIONAL,
   signatureAlgorithm SignatureAlgorithmIdentifier,
   signature SignatureValue,
   unsignedAttrs [1] IMPLICIT UnsignedAttributes OPTIONAL
}

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

SignatureValue ::= OCTET STRING
```

Die zu signierenden Daten befinden sich im Feld `eContent` in der Struktur `EncapsulatedContentInfo`. Um welchen Datentyp es sich dabei handeln muss, ist implizit durch den `contentType` in der übergeordneten `ContentInfo`-Struktur festgelegt. Eine explizite Angabe dieses Datentyps muss durch den Object Identifier im Feld `eContentType` gegeben sein. Die OIDs zur Verwendung als `eContentType` werden bei den konkreten Anwendungen einer `SignedData`-Struktur in Kapitel C.2 (signierte und/oder autorisierte Zertifikatsrequest-Pakete), Kapitel C.3 (signierte Revocationrequests) und Kapitel C.4 (signierte SMGW-Übertragungen) definiert.

Das Feld `certificates` muss die notwendigen Zertifikate (die vollständige Zertifikatskette) enthalten, mit denen die jeweilige Signatur verifiziert werden kann.

Das Feld `signerInfos` muss gemäß [8] verwendet werden und die jeweilige Signatur enthalten.

Für den Hashalgorithmus ist hierbei jeweils die OID zu verwenden, die mit dem Suffix der Hash-Funktion des verwendeten Signaturalgorithmus gemäß [21] übereinstimmt (siehe auch [12]).

C.2 Zertifikatsrequests

Im Falle von Zertifikatsrequests definiert das Feld `contentType` in der Struktur `ContentInfo`, ob es sich um ein Zertifikatsrequest-Paket mit oder ohne äußere Signatur bzw. Autorisierungssignatur handelt.

- Für ein Zertifikatsrequest-Paket (ohne äußere Signatur oder Autorisierung) muss in `ContentInfo` als `contentType` der Object Identifier `id-CertReqMsgs` und als `content` der Datentyp `CertReqMessages` gemäß Kapitel C.2.1 verwendet werden.
- Für ein signiertes Zertifikatsrequest-Paket muss in `ContentInfo` als `contentType` der Object Identifier `id-CertReqMsgs-with-outerSignature` und als `content` der Datentyp `SignedData` gemäß Kapitel C.1 verwendet werden. In der Unterstruktur `EncapsulatedContentInfo` muss als `eContentType` der Object Identifier `id-CertReqMsgs` und als `eContent` der Datentyp `CertReqMessages` gemäß Kapitel C.2.1 verwendet werden.
- Für ein autorisiertes Zertifikatsrequest-Paket muss in `ContentInfo` als `contentType` der Object Identifier `id-authorized-CertReqMsgs` und als `content` der Datentyp `SignedData` gemäß Kapitel C.1 verwendet werden. Ein autorisiertes Zertifikatsrequest-Paket muss in `EncapsulatedContentInfo` entweder ein signiertes Zertifikatsrequest-Paket oder ein Zertifikatsrequest-Paket ohne äußere Signatur beinhalten:
  - Für ein enthaltenes signiertes Zertifikatsrequest-Paket muss als `eContentType` der Object Identifier `id-CertReqMsgs-with-outerSignature` mit dem Datentyp `SignedData` gemäß Kapitel C.1 verwendet werden.
  - Für ein enthaltenes Zertifikatsrequest-Paket ohne äußere Signatur muss als `eContentType` der Object Identifier `id-CertReqMsgs` mit dem Datentyp `CertReqMessages` gemäß Kapitel C.2.1 verwendet werden.

Die hier genannten Object Identifier sind in Tabelle 29 definiert.

C.2.1 Datentyp CertReqMessages

Ein Zertifikatsrequest-Paket entspricht der Struktur `CertReqMessages`. Dieser ist gemäß [3] wie folgt definiert:

```
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 zur Identifizierung des Zertifikatsrequests innerhalb einer `CertReqMsg`-Struktur 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` muss den Namen des Antragstellers enthalten
  - Enthält insbesondere die von Antragsteller vorgeschlagene Sequenznummer.
- `SubjectPublicKeyInfo` enthält den Public Key, für den das Zertifikat beantragt wird.
- `Extensions` enthält die durch der Antragsteller anzugebenden Extensions.

Folgende Extensions dürfen 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 gemäß Anhang D (bzw. [22]) 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 [21] zu beachten.
- `Signature` enthält den Wert der Signatur.

C.3 Revocationrequests

Für zu übertragende Revocationrequests darf in `ContentInfo` als `contentType` nur der Object Identifier `id-signedRevReqs` gemäß Tabelle 29 und als `content` nur der Datentyp `Signed-Data` gemäß Kapitel C.1 verwendet werden. Revocationrequests müssen somit zwingend mit einer (äußeren) Signatur versehen werden.

Ein signierter Revocationrequest muss in `EncapsulatedContentInfo` als `eContentType` den ObjectIdentifier `id-RevReqs` und als `eContent` den Datentyp `RevReqContent` gemäß Kapitel C.3.1 verwenden.

Der Object Identifier `id-RevReqs` ist wie folgt definiert:

ContentType	*Object Identifier*
`id-RevReqs`	`bsi-de pki(4) x509(1) revocation(2) 1`

Tabelle 30: OID für Revocationrequest

C.3.1 Datentyp RevReqContent

Der Datentyp `RevReqContent` besitzt gemäß [9] folgende Struktur:

```
RevReqContent ::= SEQUENCE OF RevDetails

RevDetails ::= SEQUENCE {
         certDetails         CertTemplate,
         -- allows requester to specify as much as they can about
         -- the cert. for which revocation is requested
         -- (e.g., for cases in which serialNumber is not available)
         crlEntryDetails     Extensions       OPTIONAL
         -- requested crlEntryExtensions
     }
```

Zur Angabe eines Sperrgrunds ist eine CRL Entry Extension des Typs „ReasonCode“ gemäß [5] im Feld `crlEntryDetails` zu nutzen. Für eine Suspendierung ist der Sperrgrund `certificate-Hold` zu verwenden. Zur Rücknahme einer Suspendierung muss der Sperrgrund `removeFromCRL` angegeben werden. Sonstige mögliche Sperrgründe sind in [5] im Datentyp `CRLReason` definiert.

Zur Angabe eines zurückliegenden Datums, ab dem das Zertifikat als gesperrt zu betrachten ist, ist eine CRL Entry Extension des Typs „InvalidityDate“ gemäß [5] im Feld `crlEntryDetails` zu nutzen.

Weitere CRL Entry Extensions im Feld `crlEntryDetails` dürfen nicht angegeben werden.

Das Feld `certDetails` dient zur Identifizierung eines zu sperrenden Zertifikats. Dessen Datentyp `CertTemplate` besitzt gemäß [3] folgende Struktur:

```
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,
}
```

Im Feld `certDetails` müssen die folgenden Angaben über das zu sperrende SMGW-Zertifikat enthalten sein:

- `serialNumber` (Seriennummer des zu sperrenden Zertifikats)
- `issuer` (IssuerDN des zu sperrenden Zertifikats)

Weitere Felder von `certDetails` dürfen nicht verwendet werden.

Je Sperrauftrag dürfen nicht Zertifikate aus unterschiedlichen Zertifikatstripeln genannt werden. Sollen Zertifikate aus unterschiedlichen Tripeln gesperrt werden, so muss für jedes Tripel ein separater Sperrauftrag gesendet werden. Hinweise zur CA-seitigen Verarbeitung eines Sperrauftrags sind in Kapitel 4.3 gegeben.

C.4 Übertragung der technischen Verantwortlichkeit

Zur Übertragung der technischen Verantwortlichkeit für ein SMGW an einen GWA darf in `ContentInfo` als `contentType` nur der Object Identifier `id-signedUpdateDeviceAdmin` („signierte SMGW-Übertragung“) gemäß Tabelle 29 und als `content` der Datentyp `SignedData` gemäß Kapitel C.1 verwendet werden. Eine SMGW-Übertragung muss somit zwingend mit einer (äußeren) Signatur versehen werden.

In `EncapsulatedContentInfo` innerhalb der Datenstruktur `SignedData` muss als `eContentType` der Object Identifier `id-updateDeviceAdmin` und als `eContent` der Datentyp `UpdateDeviceAdminContent` („SMGW-Übertragung“) verwendet werden.

Der Object Identifier `id-updateDeviceAdmin` ist wie folgt definiert:

ContentType	*Object Identifier*
`id-updateDeviceAdmin`	`bsi-de application (3) Smart-Metering(4) sm-processes(2) 1`

Tabelle 31: OID für eine SMGW-Übertragung

C.4.1 Datentyp UpdateDeviceAdminContent

Der Datentyp `UpdateDeviceAdminContent` besitzt folgende Struktur:

```
UpdateDeviceAdminContent ::= SEQUENCE {
   adminCn      [0] PrintableString,
   subCaCn      [1] PrintableString,
   deviceCnSeq  [2] SEQUENCE OF PrintableString
}
```

Im Feld `adminCn` wird der CommonName des SubjectDN von C_{SIG}(GWA) desjenigen Gateway-Administrators übertragen, der die technische Verantwortlichkeit für die genannten SMGWs erhalten soll. Ein eventuell vorhandenes `<extension>`-Fragment in diesem CommonName (siehe Namensschema in [13]) ist bei der Verarbeitung zu ignorieren, so dass der GWA mit beliebigen CommonName-Extensions arbeiten kann.

Im Feld `subCaCn` wird der CommonName des IssuerDN von C_{SIG}(GWA) desjenigen Gateway-Administrators übertragen, der die technische Verantwortlichkeit für die genannten SMGWs erhalten soll.

In der Sequence `deviceCnSeq` werden die CommonNames von C_{SIG}(SMGW) der Gateways genannt, für die der genannte GWA die technische Verantwortlichkeit erhalten soll. Die Vertragspartner sollten eine maximale Anzahl von Gateways vereinbaren, die je Nachricht genannt werden dürfen.

Um die technische Verantwortlichkeit für verschiedene SMGWs an unterschiedliche GWAs zu übergeben, muss für jeden GWA ein separater Datensatz des Datentyps `UpdateDeviceAdmin-Content` übertragen werden.

D Protokolle für das Zertifikatsmanagement

Die Kommunikationsprotokolle für die Beantragung und Verteilung von Zertifikaten aus der SM-PKI, sowie für die Suspendierung und Sperrung von SMGW-Zertifikaten sind in [22] spezifiziert.

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 [11] und [2] 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 ( serialNumber ) )
```

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	c[6]
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

Tabelle 32: Attribute eines Verzeichniseintrags

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 Mit Ausnahme von SMGW-Zertifikaten ist dieses Attribut genau dann in dem entsprechenden LDAP-Eintrag vorhanden, wenn es im Subject-Feld der zugehörigen Zertifikate (des userCertificate-Attributs) angegeben ist. Bei SMGW-Zertifkaten darf dieses Feld nicht vorhanden sein.

Das Attribut `sn` erhält in jedem Eintrag den Wert `'1'`. Das Attributpaar `'cn'` und `'serialNumber'` soll als `RDN` zur Identifikation eines Eintrags verwendet werden. Abweichend davon werden Einträge von SMGW nur über das Attribut `'cn'` identifiziert. Es erfolgt die Ausgabe aller im jeweiligen Verzeichnis enthaltenen Zertifikate des SMGW.

Jede CA muss die von ihr ausgestellten Zertifikate in einem Verzeichnis unterhalb einer RDN-Struktur 'dc=Certificates, dc=SM-PKI-DE' veröffentlichen. Die jeweilige LDAP-BaseDN ist als Teil der LDAP-Server URL der CA gemäß [10] zu veröffentlichen.

Es wird empfohlen, den Verzeichnisdienst so zu konfigurieren, dass die Anzahl der zurückgegebenen Suchergebnisse geeignet begrenzt ist, siehe [13].

F Elliptische Kurven

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

- Die OBJECT IDENTIFIER der Brainpool-Kurven sind in [20] und [7] spezifiziert.
- Die OBJECT IDENTIFIER der NIST-Kurven über Primkörpern sind in [23] spezifiziert.

Die Vorgaben über die aktuell zu verwendenden elliptischen Kurven aus [21] sind zu beachten.

Literaturverzeichnis

[1] 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

[2] IETF RFC 2798, M. Smith, Definition of the inetOrgPerson LDAP Object Class, 2000

[3] IETF RFC 4211, J. Schaad, Internet X.509 Public Key Infrastructure Certificate Request Message Syntax, 2005

[4] IETF RFC 4511, J. Sermersheim, Lightweight Directory Access Protocol (LDAP): The Protocol, 2006

[5] 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

[6] IETF RFC 5480, S. Turner, R. Housley, T. Polk, Elliptic Curve Cryptography Subject Public Key Information,

[7] IETF RFC 5639, M. Lochter, J. Merkle, Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation, 2010

[8] IETF RFC 5652, R. Housley, Cryptographic Message Syntax (CMS), 2009

[9] IETF RFC 4210, C. Adams, S. Farrell, T. Kause, T. Mononen, Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP), 2005

[10] Network Working Group RFC 4516, M. Smith, Ed., Pearl Crescent, LLC, T. Howes, Opsware, Inc., Lightweight Directory Access Protocol (LDAP): Uniform Resource Locator, 2006

[11] IETF RFC 4519, A. Sciberras, Lightweight Directory Access Protocol (LDAP): Schema for User Applications, 2006

[12] IETF RFC 5754, S. Turner, RFC 5754: Using SHA2 Algorithms with Cryptographic Message Syntax, 2010

[13] BSI , Certificate Policy der Smart Metering-PKI,

[14] BSI TR-03109, Technische Richtlinie BSI-TR-03109 (Dachdokument), 2013

[15] BSI TR-03109-1 Anlage VI, Anforderungen an die Interoperabilität der Kommunikationseinheit eines 2 intelligenten Messsystems - Anlage VI: Betriebsprozesse, 2013

[16] BSI TR-03109-1, Anforderungen an die Interoperabilität der Kommunikationseinheit eines intelligenten Messsystems, 2013

[17] BSI TR-03109-2, Smart Meter Gateway – Anforderungen an die Funktionalität und Interoperabilität des Sicherheitsmoduls, 2013

[18] BSI TR-03109-3, Kryptographische Vorgaben für die Infrastruktur von intelligenten Messsystemen, 2013

[19] BSI TR-03109-6, Smart Meter Gateway Administration, 2015

[20] BSI TR-03111, Elliptic Curve Cryptography, Version 2.0, 2012

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

[22] BSI TR-03129-X, Protocols for the Management of Certificates and CRLs in Public-Key-Infrastructures (PKIs), V1.30, 2016

[23] ANSI X9.62, Public Key Cryptography for the Financial Services Industry . The Elliptic Curve Digital Signature Algorithm (ECDSA), 2005