Bundesamt
für Sicherheit in der
Informationstechnik

Technische Richtlinie BSI TR-03109-2

Smart Meter Gateway – Anforderungen an die Funktionalität und Interoperabilität des Sicherheitsmoduls

Version 1.1 – 15.12.2014

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

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

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

Zur Erfüllung der dazu benötigten kryptographischen Funktionen bedient sich das SMGW eines gemäß [PP 0077] nach Common Criteria zertifizierten Sicherheitsmoduls. Als zentrale Sicherheitskomponente

- stellt das Sicherheitsmodul die kryptographische Identität des SMGW sicher und
- dient dem SMGW als Service Provider für kryptographische Operationen.

Das Sicherheitsmodul stellt insbesondere Funktionen

- zur Schlüsselgenerierung,
- zur Erzeugung und Verifikation von Digitalen Signaturen und
- zur Schlüsselaushandlung

auf Basis von Kryptographie mit Elliptischen Kurven bereit.

Weiterhin dient das Sicherheitsmodul

- als zuverlässige Quelle für Zufallszahlen und
- als sicherer Speicher von Schlüsseln und Zertifikaten.

Ferner unterstützt das Sicherheitsmodul einen authentisierten und gesicherten Kanal zwischen dem Gateway und dem Sicherheitsmodul.

In der vorliegenden Technischen Richtlinie BSI TR-03109-2 werden die Anforderungen an die Funktionalität und Interoperabilität des Sicherheitsmoduls für das SMGW spezifiziert. Hierzu liefert diese Technische Richtlinie neben einer detaillierten Beschreibung des Lebenszyklus-Modells von Sicherheitsmodul bzw. SMGW insbesondere die Spezifikation des File- und Objektsystems, der Zugriffsregeln und des Kommandosets des Sicherheitsmoduls. In einem Anhang zu dieser Technischen Richtlinie [TR-03109-2A] werden weiterhin Anwendungsfälle (Use Cases) mit einer Beschreibung der Nutzung des Sicherheitsmoduls durch das SMGW aufgeführt.

Sicherheitstechnische Anforderungen an das Sicherheitsmodul selbst werden darüber hinaus durch das Common Criteria-Schutzprofil [PP 0077] festgelegt.

Diese Technische Richtlinie BSI TR-03109-2 gliedert sich wie folgt:

In Kapitel 2 wird das Lebenszyklus-Modell für das SMGW mit seinem Sicherheitsmodul detailliert beschrieben. Betrachtet werden dabei die verschiedenen Phasen des Lebenszyklus inklusive der jeweils wesentlichen Aufgaben, der beteiligten Rollen und des relevanten Schlüssel- und Zertifikatsmaterials.

Kapitel 3 beschreibt das File- bzw. Objektsystem des Sicherheitsmoduls, die vom Sicherheitsmodul umgesetzten Zugriffsregeln für Files, Objekte und Kommandos des Sicherheitsmoduls sowie das vom Sicherheitsmodul angebotene Set von Kommandos. Die Spezifikation des File- und Objektsystems und des Kommandosets ist auf das Lebenszyklus-Modell des Sicherheitsmoduls bzw. SMGW abgestimmt und wurde mit dem Ziel größtmöglicher Flexibilität aufgesetzt.

Kapitel 4 bricht die Spezifikationen aus Kapitel 3 auf die technische Ebene herunter und liefert insbesondere die Detail-Spezifikation der Kommandos des Sicherheitsmoduls inklusive Kommando-spezifischer Informationen und Anforderungen zu Implementierungsdetails.

In einem informativen Anhang zu dieser Technischen Richtlinie [TR-03109-2A] schließlich werden Anwendungsfälle aufgeführt, die die Nutzung des Sicherheitsmoduls durch das SMGW beschreiben und das Zusammenspiel zwischen GW und Sicherheitsmodul in den verschiedenen Phasen des Lebenszyklus-Modells illustrieren.

1.1 Einordnung des Dokuments

Das vorliegende Dokument ist Teil der BSI TR-03109 ([TR-03109]) und spezifiziert die Anforderungen an die Funktionalität und Interoperabilität des Sicherheitsmoduls für das Smart Meter Gateway (SMGW). Neben den in den folgenden Kapiteln referenzierten Spezifikationen und Standards sind darüber hinaus insbesondere die folgenden Dokumente zu berücksichtigen:

- **Technische Richtlinie BSI TR-03109** ([TR-03109])

  Die Technische Richtlinie [TR-03109] spezifiziert die Funktionalitäts-, Interoperabilitäts- und Sicherheitsanforderungen an die Einzelkomponenten in einem Smart Metering System, beschreibt die Betriebsprozesse und die Rollen der Marktteilnehmer in der Infrastruktur von Messsystemen und liefert damit die zentrale Grundlage für die Spezifikation des Sicherheitsmoduls in der vorliegenden Technischen Richtlinie BSI TR-03109-2.

  Die Technische Richtlinie [TR-03109] gliedert sich insgesamt wie folgt:

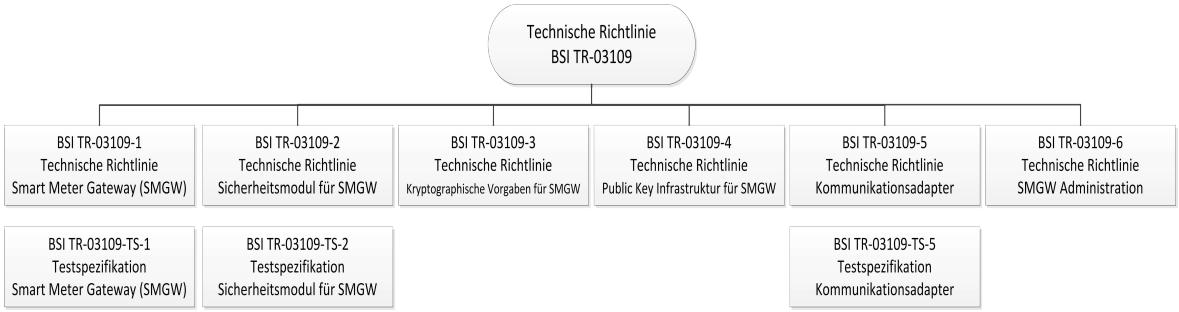

Abbildung 1: Dokumentenstruktur der BSI TR-03109

- **Technische Richtlinie BSI TR-03109-1** ([TR-03109-1])

  Die Technische Richtlinie [TR-03109-1] mit ihren zugehörigen Anhängen spezifiziert die funktionalen Anforderungen, Sicherheitsanforderungen und Interoperabilitätseigenschaften der Kommunikationseinheit eines intelligenten Messsystems, kurz Smart Meter Gateway. Das Sicherheitsmodul bildet dabei im Smart Meter Gateway eine zentrale Sicherheits-komponente. Die [TR-03109-1] mit ihren Anhängen stellt somit die Grundlage für die Spezifikation des Sicherheitsmoduls für das Smart Meter Gateway dar.

  Unter den Anhängen zur Technischen Richtlinie [TR-03109-1] ist insbesondere der Anhang [TR-03109-1A], der der Beschreibung von Betriebsprozessen dient und sich insbesondere mit dem Lebenszyklus-Modell des Smart Meter Gateways beschäftigt und insofern für die vorliegende Technische Richtlinie BSI TR-03109-2 mit ihrem Anhang von Relevanz ist, hervorzuheben.

- **Technische Richtlinie BSI TR-03109-3** ([TR-03109-3])

  In der Technischen Richtlinie [TR-03109-3] werden die Sicherheitsanforderungen für den Einsatz kryptographischer Verfahren in der Infrastruktur von Messsystemen im Energiesektor beschrieben.

- **Technische Richtlinie BSI TR-03109-4** ([TR-03109-4])

  Die Technische Richtlinie [TR-03109-4] spezifiziert die Architektur sowie die Mindestanforderungen an die Interoperabilität und Sicherheit der SM-PKI.

- **Technische Richtlinie BSI TR-03109-5**

  Die Technische Richtlinie BSI TR-03109-5 spezifiziert die Anforderungen an den sog. Kommunikationsadapter, der für die Kommunikation zwischen MID-konformen Zählern und dem Smart Meter Gateway erforderlich ist.

- **Technische Richtlinie BSI TR-03109-6**

  Die Technische Richtlinie BSI TR-03109-6 beschreibt die Aufgaben des GW-Administrators und stellt die Anforderungen an den GW-Administrator zusammen.

1.2 Terminologie

Diese Technische Richtlinie BSI TR-03109-2 ist grundsätzlich als normativ anzusehen. Informative Teile werden explizit als solche gekennzeichnet (mit dem Vermerk „informativ“ oder „Hinweis“).

1.3 Abkürzungen

In dieser Technischen Richtlinie BSI TR-03109-2 werden folgende Abkürzungen verwendet:

Abkürzung	Begriff
AID	Application Identifier
APDU	Application Protocol Data Unit
AT	Authentication Template
ATR	Answer To Reset
ATS	Answer To Select
BSI	Bundesamt für Sicherheit in der Informationstechnik
CA	Certificate Authority
CC	Common Criteria
CP	Control Parameter
CRT	Control / Cryptographic Reference Template
DF	Dedicated File
DH	Diffie-Hellman
DST	Digital Signature Template
ECC	Elliptic Curve Cryptography
EF	Elementary File
EMT	Externer Marktteilnehmer
Enc	Encryption
ENu	Endnutzer (z.B. Externer Marktteilnehmer, GW-Administrator, SMGW, ...)

Abkürzung	Begriff
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDH	Elliptic Curve Diffie-Hellman
ECKA	Elliptic Curve Key Agreement
ECKA-DH	Elliptic Curve Key Agreement-Diffie-Hellman
ECKA-EG	Elliptic Curve Key Agreement-ElGamal
EG	ElGamal
EVG	Evaluierungsgegenstand
FCP	File Control Parameter
FID	File Identifier
GW	Gateway
GWA	Gateway-Administrator
HAN	Home Area Network
ID	Identifier
ISO	International Organization for Standardization
KDF	Key Derivation Function
KID	Key Identifier / Key-ID
KM	Kryptografiemodul
LCSI	Life Cycle Status Information
LMN	Local Metrological Network
MF	Master File
MSBit	Most Significant Bit
MT	Marktteilnehmer
NIST	National Institute of Standards and Technology
OID	Object Identifier
OS	Operating System
PIN	Personal Identification Number
PKI	Public Key Infrastruktur
PP	Protection Profile (Common Criteria)
RFU	Reserved for Future Use
RNG	Random Number Generator
SE	Security Environment
SEID	Security Environment ID
SecMod	Security Module / Sicherheitsmodul
SFI	Short File Identifier
SHA	Secure Hash Algorithm
Sign	Signature
SM	Smart Meter
SMGW	Smart Meter Gateway
SM-PKI	Smart Metering - Public Key Infrastruktur (SM-PKI)
SP	Security Parameter
TLS	Transport Layer Security
TOE	Target Of Evaluation (Common Criteria)
TR	Technische Richtlinie

Abkürzung	Begriff
WAN	Wide Area Network

Tabelle 1: Übersicht der Abkürzungen

1.4 Änderungshistorie

Version	Datum	Änderung
V 1.0	18.03.2013	Erstausgabe
V 1.01	08.05.2014	Inhaltliche und editorische Anpassungen in allen Kapiteln, Aktualisierung des Literaturverzeichnisses
V 1.0.2	15.08.2014	Ergänzung von Klarstellungen, kleinere Fehlerkorrekturen
V 1.1	15.12.2014	Ergänzung von Klarstellungen, Aktualisierung des Literaturverzeichnisses

Tabelle 2: Änderungshistorie

2 Lebenszyklus-Modell

Das vorliegende Kapitel geht auf das Lebenszyklus-Modell für Sicherheitsmodul und Gateway (GW) ein. Dieses wird bereits auch in [TR-03109-1] und [TR-03109-1A] betrachtet. Jedoch werden für die vorliegende Spezifikation des Sicherheitsmoduls und die in [TR-03109-2A] beschriebenen Anwendungsfälle (Use Cases) weitere Detailbetrachtungen erforderlich, die in den folgenden Kapiteln genauer ausgeführt werden.

2.1 Übersicht über das Lebenszyklus-Modell

In den folgenden Abschnitten wird zunächst eine grobe Übersicht über das Lebenszyklus-Modell für Sicherheitsmodul und Gateway (GW) gegeben. Insbesondere werden die einzelnen Phasen des Lebenszyklus-Modells und die beteiligten Rollen benannt.

2.1.1 Phasen des Lebenszyklus-Modells

Das Lebenszyklus-Modell für Sicherheitsmodul und GW gliedert sich in folgende aufeinander aufbauende Phasen:

1. Herstellungs- und Produktionsprozesse von GW und Sicherheitsmodul (siehe Kap. 2.2.1)
2. Vor-Personalisierung + Integration von Sicherheitsmodul und GW (siehe Kap. 2.2.2)
3. Installation + Vor-Ort-Inbetriebnahme des SMGW (siehe Kap. 2.2.3)
4. Personalisierung des SMGW (siehe Kap. 2.2.4)
5. Normalbetrieb (End-Usage mit Administration und Smart Meter Wirkbetrieb) des SMGW (siehe Kap. 2.2.5)
6. Außerbetriebnahme des SMGW (siehe Kap. 2.2.6)

Das Lebenszyklus-Modell stützt sich dabei ausschließlich auf die für das Smart Meter-System vorgesehene SM-PKI.

In Kap. 2.2 erfolgt eine detaillierte Beschreibung der zuvor genannten Phasen des Lebenszyklus-Modells. In den einzelnen Phasenbeschreibungen werden zum einen die durchzuführenden Aufgaben und Randbedingungen benannt. Zum anderen werden jeweils auch die beteiligten Rollen, die einen Zugriff auf das GW bzw. Sicherheitsmodul haben, zugeordnet.

2.1.2 Rollen im Lebenszyklus-Modell

Folgende Rollen sind im Lebenszyklus-Modell für Sicherheitsmodul und GW involviert:

- Integrator
- Erstkonfigurator (für die Parametrierung der Kommunikationsschnittstellen des GW im Rahmen der Inbetriebnahme)
- GW-Administrator (für die Administration des SMGW)
- Service Techniker (für die Parametrierung der Kommunikationsschnittstellen des GW im Normalbetrieb, Diagnose usw.)

- Kommunikationspartner im WAN, LMN bzw. HAN (für den Smart Meter Wirkbetrieb)

In Kap. 2.2 erfolgt im Rahmen der Detailbeschreibung der einzelnen Phasen des Lebenszyklus-Modells auch eine Benennung der jeweils beteiligten Rollen und eine detaillierte Beschreibung ihrer Aufgaben.

2.1.3 Weitere Aspekte

2.1.3.1 Lokale Schnittstelle des Gateways

Das GW besitzt eine lokale Schnittstelle, die im Rahmen der Phasen „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ durch den Integrator und „Installation + Vor-Ort-Inbetriebnahme des SMGW“ durch den Erstkonfigurator zum Datenimport/-export bzw. zur Administration genutzt werden kann. Aus Sicherheitsgründen ist diese lokale Schnittstelle möglichst schlank gehalten und wird auf die unbedingt erforderliche Funktionalität und den unbedingt notwendigen Datentransport beschränkt (siehe [TR-03109-1] und [TR-03109-1A]).

Für die lokale Schnittstelle des GW ist aus physikalischer Sicht die HAN-Schnittstelle des GW vorgesehen. Die Nutzung dieser lokalen Schnittstelle des GW in den Phasen „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ und „Installation + Vor-Ort-Inbetriebnahme des SMGW“ ist über einen geeigneten PIN-Mechanismus im GW abgesichert. Der Integrator bzw. der Erstkonfigurator hat sich für eine Nutzung der lokalen Schnittstelle des GW mittels PIN gegenüber dem GW zu authentisieren, wobei Integrator und Erstkonfigurator jeweils eine eigene PIN verwenden („Integrator-PIN“, „Erstkonfigurator-PIN“). Die Rolle des Erstkonfigurators erhält über die lokale Schnittstelle keine Integrator-Rechte zur Vor-Personalisierung und Integration des Sicherheitsmoduls.

Der PIN-Mechanismus ist allein Aufgabe des GW; das Sicherheitsmodul ist in diesen Mechanismus nicht involviert und leistet auch keinen weiteren funktionalen oder sicherheitstechnischen Beitrag zur Absicherung der lokalen Schnittstelle des GW. Die lokale Schnittstelle des GW und ihr PIN-Sicherungsmechanismus ist Gegenstand der CC-Zertifizierung des GW (CC-Aspekte ADV_ARC und ALC_DEL).

Die PIN-gesicherte lokale Schnittstelle des GW ist nur in den Phasen „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ und „Installation + Vor-Ort-Inbetriebnahme des SMGW“ verfügbar und wird mit Ende der Phase „Installation + Vor-Ort-Inbetriebnahme des SMGW“ deaktiviert.

Details zur PIN-gesicherten lokalen Schnittstelle des GW sind in [TR-03109-1] und [TR-03109-1A] geregelt.

2.1.3.2 Zertifikate

Zertifikate werden grundsätzlich im GW, nicht aber im Sicherheitsmodul gespeichert. Ausnahme bilden die Zertifikate der SM-PKI-Root sowie die Gütesiegel-Zertifikate des GW.

SM-PKI-Root-Zertifikate können grundsätzlich für eine selektive Prüfung von Zertifikaten eingesetzt werden (z.B. im Rahmen des Imports von Schlüsseln bzw. Zertifikaten aus der SM-PKI in das GW).

SM-PKI-Root-Zertifikate sollen im Sicherheitsmodul updatefähig sein, da die Laufzeit der Root-Zertifikate begrenzt ist, siehe [TR-03109-4]. Ein Update eines Root-Zertifikates soll aber nur unter Einhaltung geeigneter Randbedingungen möglich sein, da das initial eingebrachte, vor-personalisierte SM-PKI-Root-Zertifikat einen zentralen Sicherheitsanker bildet. Für SM-PKI-Root-Zertifikate ist mit Übergangszeiten zu rechnen, in denen alte wie auch neue Root-Zertifikate im Sicherheitsmodul verfügbar sein sollen, so dass entsprechender Speicherbereich im Filesystem des Sicherheitsmoduls vorzusehen ist. Siehe hierzu auch Kap. 3.2.5.

2.1.3.3 Life Cycle-Status

Das Sicherheitsmodul verwaltet einen eigenen Life Cycle-Status mit den Werten „nicht-initialisiert“ (d.h. das Sicherheitsmodul ist noch nicht initialisiert), „initialisiert“ (d.h. das Sicherheitsmodul ist initialisiert und mit den für das Smart Meter-System vorgesehenen initialen Strukturen und Daten bestückt) und „terminiert“ (d.h. das Sicherheitsmodul ist irreversibel außer Betrieb genommen), siehe Kap. 3.6.1.

Das GW kann zusätzlich einen eigenen Life Cycle-Status mitführen und Informationen dazu im Sicherheitsmodul hinterlegen. Eine Auswertung dieses Life Cycle-Status des GW durch das Sicherheitsmodul erfolgt jedoch nicht. Siehe Kap. 3.2.3 und 3.6.1.

2.1.3.4 Zugriffsregeln

Die in den einzelnen Phasen des Lebenszyklus-Modells relevanten Zugriffsregeln für die Restriktion des Zugriffs auf das Sicherheitsmodul, seine Datenfelder und Objekte sowie Kommandos werden über entsprechende sog. Security Environments im Sicherheitsmodul festgelegt, siehe Kap. 3.3.

2.2 Detailbeschreibung des Lebenszyklus-Modells

In den folgenden Abschnitten wird eine Detailbeschreibung des Lebenszyklus-Modells für Sicherheitsmodul und Gateway (GW) gegeben, und insbesondere werden die einzelnen Phasen des Lebenszyklus-Modells, die jeweils beteiligten Rollen und die jeweils durchzuführenden Aufgaben genauer beleuchtet.

In einem informativen Anhang zu dieser Technischen Richtlinie [TR-03109-2A] werden Anwendungsfälle (Use Cases) aufgeführt, die die Nutzung des Sicherheitsmoduls durch das SMGW beschreiben und das Zusammenspiel zwischen GW und Sicherheitsmodul in den verschiedenen Phasen des Lebenszyklus-Modells illustrieren. Mit den in [TR-03109-2A] beschriebenen Use Cases lassen sich die in den einzelnen Phasen des Lebenszyklus-Modells durchzuführenden Aufgaben realisieren.

2.2.1 Herstellungs- und Produktionsprozesse von Gateway und Sicherheitsmodul

Die Herstellungs- und Produktionsprozesse (sowie Zertifizierungsprozesse) für GW und Sicherheitsmodul werden bereits in [TR-03109-1], [TR-03109-1A], [PP 0073] und [PP 0077] betrachtet.

Im Rahmen der Herstellung und Produktion des GW erfolgt die Vergabe einer eindeutigen Geräte-ID zur Identifikation des GW (Produkt-ID).

Hinweis: Die Geräte-ID ist über eine spezielle DIN-Norm (siehe [TR-03109-1] und [TR-03109-1A]) geregelt und liefert eigentlich nur eine Identifikation des GW-Produktes, nicht aber der Kombination von GW und Sicherheitsmodul. Die Gütesiegel-Zertifikate (aus der SM-PKI) jedoch, siehe Kap. 2.2.2.1.3, beinhalten die Geräte-ID und schaffen indirekt eine Bindung zwischen der Geräte-ID des GW und dem Sicherheitsmodul, das die zu den Gütesiegel-Zertifikaten zugehörigen GW-Schlüsselpaare speichert.

Für die in den nachfolgenden Kapiteln dargestellten Phasen des Lebenszyklus-Modells werden die Herstellungs- und Produktionsprozesse von GW und Sicherheitsmodul als abgeschlossen vorausgesetzt.

Ausgangspunkt für die in den nachfolgenden Kapiteln dargestellten Phasen des Lebenszyklus-Modells ist genauer:

- Das Sicherheitsmodul liegt als initialisiertes Modul vor, d.h. das Sicherheitsmodul beinhaltet neben der Betriebssystem-Plattform mit den Kommandos insbesondere das für das Smart Meter-System vordefinierte File- und Objektsystem mit den vorgesehenen Vorbelegungen, siehe Kap. 3 und 4.
- Die Implementierung des GW liegt vor.

2.2.2 Vor-Personalisierung + Integration von Sicherheitsmodul und GW

Die Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ dient der Integration von GW und (initialisiertem) Sicherheitsmodul sowie dem Generieren und Aufbringen von initialem Schlüssel- und Zertifikatsmaterial.

Ein integriertes GW, also ein GW mit eingebautem und verbundenem Sicherheitsmodul wird im folgenden als SMGW, kurz für Smart Meter Gateway, bezeichnet.

Die Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ setzt sich aus den Teilphasen „Vor-Personalisierung“ und „Integration“ zusammen und wird beim Integrator durchgeführt. Die „Vor-Personalisierung“ gliedert sich dabei in zwei Teilphasen auf. Während in der Teilphase „Vor-Personalisierung 1“ Schlüssel- und Zertifikatsmaterial generiert und aufgebracht wird, das unabhängig vom späteren GW-Administrator ist, wird in der Teilphase „Vor-Personalisierung 2“ GW-Administrator spezifisches Schlüssel- und Zertifikatsmaterial importiert.

Hinsichtlich der Reihenfolge der Teilphasen „Vor-Personalisierung 1“, „Vor-Personalisierung 2“ und „Integration“ gilt:

- „Vor-Personalisierung 1“ findet vor „Vor-Personalisierung 2“ statt.
- „Integration“ findet vor „Vor-Personalisierung 2“ statt.
- Der Integrationsschritt des physikalisch-technischen Einbaus des Sicherheitsmoduls in das GW und der Verbindung der beiden Komponenten als Teilschritt der „Integration“ (siehe Kap. 2.2.2.1.2) kann alternativ vor „Vor-Personalisierung 1“ oder zwischen „Vor-Personalisierung 1“ und „Vor-Personalisierung 2“ stattfinden.
- Findet der Integrationsschritt des physikalisch-technischen Einbaus des Sicherheitsmoduls in das GW und der Verbindung der beiden Komponenten als Teilschritt der „Integration“ (siehe Kap. 2.2.2.1.2) vor „Vor-Personalisierung 1“ statt, so finden die weiteren

Prozessschritte der „Integration“ (siehe Kap. 2.2.2.1.2) alternativ vor „Vor-Personalisierung 1“ oder zwischen „Vor-Personalisierung 1“ und „Vor-Personalisierung 2“ statt.

Für den Import von Schlüsseln in dieser Phase werden sog. Import-Schlüssel verwendet, siehe Kap. 3.1.2. Initiale Import-Schlüssel werden bereits im Rahmen der Produktion (Initialisierung) des Sicherheitsmoduls aufgebracht. Zum Abschluss der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ müssen sämtliche Import-Schlüssel im Sicherheitsmodul (genauer deren Key Pair-Objekte und Public Key-Objekte) gelöscht werden.

2.2.2.1 Rollen und Aufgaben

Rolle: Integrator

Anforderungen an den Integrator:

- Beim Integrator besteht eine ausreichend durch technische, organisatorische und personelle Sicherheitsmaßnahmen gesicherte Umgebung.
- Sicherheitsmodul-spezifische weitere technische Sicherungsmaßnahmen sind zulässig. Ggf. wird dazu erforderliches Sicherungsmaterial (z.B. Keys, PINs, …) im Rahmen des Initialisierungsprozesses im Sicherheitsmodul hinterlegt und soweit erforderlich an den Integrator zur Nutzung ausgeliefert. Hierzu setzen viele Sicherheitsmodul-Hersteller auf übliche Sicherheitsmechanismen auf. In [TR-03109] und im Protection Profile [PP 0077] zum Sicherheitsmodul werden keine konkreten Sicherheitsmechanismen festgeschrieben, um den Herstellern von Sicherheitsmodulen genügend Freiraum für ihre spezifischen, ggf. bereits etablierten Sicherheitsmechanismen zu lassen.

2.2.2.1.1 Vor-Personalisierung 1

Aufgaben in der „Vor-Personalisierung 1“ von Sicherheitsmodul und GW:

- Import des SM-PKI-Root-Zertifikates (ROOT_WAN_SIG_CRT) in das Sicherheitsmodul.

  (Hinweis: Das SM-PKI-Root-Zertifikat stellt nachfolgend einen zentralen Sicherheitsanker dar; seine Integrität und Authentizität ist daher sicherzustellen. Es erfolgt eine vertrauenswürdige Übermittlung des Zertifikates an den Integrator. Die Authentizität wird z.B. über einen Abgleich eines Fingerprints über einen zweiten Kanal sichergestellt.)
- Onboard-Generierung von Key-Paaren für das GW im Sicherheitsmodul (vorläufige GW-Keys für die WAN-Kommunikation):
  - (GW_WAN_TLS_PRV_PRE, GW_WAN_TLS_PUB_PRE)
  - (GW_WAN_SIG_PRV_PRE, GW_WAN_SIG_PUB_PRE)
  - (GW_WAN_ENC_PRV_PRE, GW_WAN_ENC_PUB_PRE)
- Export der Public Keys:
  - GW_WAN_TLS_PUB_PRE
  - GW_WAN_SIG_PUB_PRE
  - GW_WAN_ENC_PUB_PRE

(Hinweis: Die zugehörigen privaten Keys verbleiben im Sicherheitsmodul und können nicht ausgelesen werden.)

- Erstellung des entsprechenden Zertifikatsrequest-Pakets für die Gütesiegel-Zertifikate zu den GW-Keys und Senden des Zertifikatsrequest-Pakets „an die SM-PKI“.

  (Hinweis: Hierzu benötigt der Integrator eine Verbindung zur SM-PKI. Es besteht hierbei die Möglichkeit, dass der Integrator selbst eine Sub-CA in der SM-PKI betreibt.)

- Import der Gütesiegel-Zertifikate in das Sicherheitsmodul:
  - GW_WAN_TLS_CRT_PRE
  - GW_WAN_SIG_CRT_PRE
  - GW_WAN_ENC_CRT_PRE

Hinweis: Sämtliches im Rahmen der „Vor-Personalisierung 1“ generiertes bzw. eingebrachtes Schlüssel- und Zertifikatsmaterial ist unabhängig vom späteren konkreten GW-Administrator.

Hinweis: Je nach Reihenfolge von „Integration“ und „Vor-Personalisierung 1“ ist ggf. die Nutzung der lokalen Schnittstelle des GW unter Verwendung der Integrator-PIN (siehe Kap. 2.1.3.1) erforderlich.

2.2.2.1.2 Integration

Aufgaben in der „Integration“ von Sicherheitsmodul und GW:

- Physikalisch-technischer Einbau des Sicherheitsmoduls in das GW und Verbindung der Komponenten.
- Generierung der Keys für die Speicherverschlüsselung des GW (sofern nicht schon im Rahmen der Produktion des GW erfolgt) und ggf. Import der Keys in das Sicherheitsmodul.
- Setzen der GW-System-PIN.

  Beim ersten Start des SMGW wird im GW eine GW-System-PIN generiert und im Sicherheitsmodul gesetzt. Über die PIN wird eine Bindung zwischen GW und SM, das sog. „Pairing zwischen GW und Sicherheitsmodul“, erreicht.

Der Prozessschritt des physikalisch-technischen Einbaus des Sicherheitsmoduls in das GW und der Verbindung der Komponenten kann im Rahmen der Teilphase „Integration“ in einem eigenen, von den übrigen o.g. Integrationsschritten abgetrennten Prozess durchgeführt werden. Das Setzen der GW-System-PIN setzt einen abgeschlossenen Prozessschritt des physikalisch-technischen Einbaus des Sicherheitsmoduls in das GW und der Verbindung der Komponenten voraus. Hinsichtlich der Eingliederung der Teilphase „Integration“ und ihrer einzelnen Integrationsschritte in die Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ siehe auch die Ausführungen in Kap. 2.2.2.

Ist auf Seiten des Integrators ein direkter Zugriff auf das Sicherheitsmodul für dessen „Vor-Personalisierung 1“ möglich, so kann die „Vor-Personalisierung 1“ ohne Nutzung der lokalen Schnittstelle des GW (siehe Kap. 2.1.3.1) erfolgen. Sollte hingegen auf Seiten des Integrators kein direkter Zugriff auf das Sicherheitsmodul für dessen „Vor-Personalisierung 1“ mehr möglich sein (z.B. im Fall, dass die „Vor-Personalisierung 1“ erst nach abgeschlossener „Integration“ von Sicherheitsmodul und GW stattfindet), so ist die lokale Schnittstelle des GW unter Verwendung der Integrator-PIN (siehe Kap. 2.1.3.1) für die „Vor-Personalisierung 1“ zu benutzen.

2.2.2.1.3 Vor-Personalisierung 2

Voraussetzungen für die „Vor-Personalisierung 2“ von Sicherheitsmodul und GW:

- „Vor-Personalisierung 1“ und „Integration“ von Sicherheitsmodul und GW sind abgeschlossen.
- Zum Zeitpunkt der „Vor-Personalisierung 2“ ist der konkrete GW-Administrator bekannt.
- Der GW-Administrator hat seine Schlüsselpaare
  - (GWADM_TLS_PRV, GWADM_TLS_PUB)
  - (GWADM_SIG_PRV, GWADM_SIG_PUB)
  - (GWADM_ENC_PRV, GWADM_ENC_PUB)
  - (GWADM_AUT_PRV, GWADM_AUT_PUB)

  generiert.

  Hinweis: An die Schlüsselgenerierung beim GW-Administrator bestehen spezifische Anforderungen, wie z.B. die Nutzung eines HSM (siehe TR-03109-6).
- Der GW-Administrator besitzt für seine zuvor genannten Schlüsselpaare jeweils ein Zertifikat der SM-PKI, also
  - GWADM_TLS_CRT
  - GWADM_SIG_CRT
  - GWADM_ENC_CRT
  - GWADM_AUT_CRT
- Der GW-Administrator erstellt eine sog. „Initiale Konfigurationsdatei“, die mindestens die folgenden Daten beinhaltet:
  - Kommunikationsparameter des GW-Administrators (GW-Administrator-Adresse)
  - Zertifikate GWADM_TLS_CRT, GWADM_SIG_CRT, GWADM_ENC_CRT und GWADM_AUT_CRT mit zugehöriger Zertifikatskette (aus der SM-PKI) exklusive des SM-PKI-Root-Zertifikates

  Anmerkung: Die Strukturierung und ggf. weiteren Inhalte der „Initialen Konfigurationsdatei“ sind über [TR-03109-1] geregelt.
- Der GW-Administrator signiert die „Initiale Konfigurationsdatei“ mit seinem Signaturschlüssel GWADM_SIG_PRV.

  Anmerkung: Die gesamte Konfigurationsdatei und damit insbesondere auch die darin enthaltenen Zertifikate der SM-PKI und ihrer Zertifikatsketten werden signiert, auch wenn dieses aus Sicherheitssicht nicht unbedingt erforderlich ist, da nachfolgend sowieso eine Prüfung der Zertifikate aus der SM-PKI und ihrer Zertifikatsketten bis zur SM-PKI-Root erfolgt. Ferner wird das Root-Zertifikat explizit nicht mit der Konfigurationsdatei ausgeliefert, um eine Prüfung der Zertifikatsketten gegen das im Sicherheitsmodul als Sicherheitsanker gespeicherte Root-Zertifikat „zu erzwingen“.
- Die signierte „Initiale Konfigurationsdatei“ wird an den Integrator übermittelt (z.B. vom GW-Administrator).

Aufgaben in der „Vor-Personalisierung 2“ von Sicherheitsmodul und GW:

- Einspielen der Kommunikationsparameter und Zertifikate des GW-Administrators unter Nutzung der lokalen Schnittstelle des SMGW (siehe Kap. 2.1.3.1), hierzu:
  - Der Integrator authentisiert sich gegenüber dem SMGW (unter Nutzung der Integrator-PIN) und schaltet damit die lokale Schnittstelle des SMGW für die „Vor-Personalisierung 2“ frei.
  - Der Integrator spielt über die lokale Schnittstelle die vom GW-Administrator signierte „Initiale Konfigurationsdatei“ ein. Es erfolgt eine Speicherung der in der „Initialen Konfigurationsdatei“ enthaltenen Zertifikate und Zertifikatsketten im GW.
  - Im GW erfolgt automatisch (unter Nutzung des Sicherheitsmoduls) die Überprüfung der in der „Initialen Konfigurationsdatei“ gelieferten Zertifikate des GW-Administrators aus der SM-PKI. Dies erfolgt unter Verwendung der mitgelieferten Zertifikatsketten und des SM-PKI-Root-Zertifikates, das im Rahmen der „Vor-Personalisierung 1“ als Sicherheitsanker integer und authentisch in das Sicherheitsmodul eingebracht wurde und dort hinterlegt ist (s.o.).

    Hinweis: Das Sicherheitsmodul unterstützt durch Bereitstellung der Kernroutine zur Signaturprüfung die Prüfung von X.509-Zertifikaten und -Zertifikatsketten. Die Prüfung von X.509-Zertifikaten und -Zertifikatsketten findet eigentlich im GW statt, wozu sich das GW der Signaturprüfungs-Routine des Sicherheitsmoduls bedient und die Daten für die Signaturprüfung im Sicherheitsmodul entsprechend aufbereitet. Das Sicherheitsmodul selbst stellt *keine* komplette Funktionalität zur Prüfung von X.509-Zertifikaten und -Zertifikatsketten bereit. (Üblicherweise prüfen Module/Chipkarten allenfalls CV-Zertifikate; für die Prüfung von X.509-Zertifikaten müssten proprietäre Routinen aufgesetzt werden.)
  - Im GW erfolgt automatisch (unter Nutzung des Sicherheitsmoduls) die Überprüfung der Signatur der „Initialen Konfigurationsdatei“.
  - Erst nach erfolgreicher Überprüfung der in der „Initialen Konfigurationsdatei“ gelieferten Zertifikate aus der SM-PKI und der Signatur der „Initialen Konfigurationsdatei“ erfolgt der Import der Public Keys des GW-Administrators (wie in der „Initialen Konfigurationsdatei“ enthalten) in das Sicherheitsmodul und werden die Kommunikationsdaten des GW-Administrators (wie in der „Initialen Konfigurationsdatei“ geliefert) für ihre weitere Verwendung freigeschaltet.

Die in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ erforderliche sog. „Initiale Konfigurationsdatei“ des GW-Administrators für den Integrator soll aus Sicherheitsgründen nur in dieser Phase verwendet werden können, da mit der „Initialen Konfigurationsdatei“ sicherheitskritische Schlüssel des GW-Administrators importiert werden.

2.2.2.2 Schlüssel- und Zertifikatsmaterial

Folgendes Schlüssel- und Zertifikatsmaterial liegt am Ende der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ (mindestens) im Sicherheitsmodul und GW vor:

Im Sicherheitsmodul:

- GW-System-PIN (Referenzwert)

- Ggf. Keys für die Speicherverschlüsselung des GW
- Zertifikat ROOT_WAN_SIG_CRT und darin enthaltener Public Key der SM-PKI-Root
- Vorläufige GW-Schlüsselpaare:
  - (GW_WAN_TLS_PRV_PRE, GW_WAN_TLS_PUB_PRE)
  - (GW_WAN_SIG_PRV_PRE, GW_WAN_SIG_PUB_PRE)
  - (GW_WAN_ENC_PRV_PRE, GW_WAN_ENC_PUB_PRE)
- Gütesiegel-Zertifikate:
  - GW_WAN_TLS_CRT_PRE
  - GW_WAN_SIG_CRT_PRE
  - GW_WAN_ENC_CRT_PRE
- Public Keys des GW-Administrators:
  - GWADM_TLS_PUB
  - GWADM_SIG_PUB
  - GWADM_ENC_PUB
  - GWADM_AUT_PUB

Im GW:

- GW-System-PIN
- Keys für die Speicherverschlüsselung des GW
- Zertifikate des GW-Administrators:
  - GWADM_TLS_CRT
  - GWADM_SIG_CRT
  - GWADM_ENC_CRT
  - GWADM_AUT_CRT
  - Jeweils inkl. der zugehörigen Zertifikatskette (aus der SM-PKI) exklusive des SM-PKI-Root-Zertifikates
- Kommunikationsparameter des GW-Administrators (GW-Administrator-Adresse)

2.2.3 Installation + Vor-Ort-Inbetriebnahme des SMGW

Die Phase „Installation + Vor-Ort-Inbetriebnahme des SMGW“ umfasst die Installation des SMGW beim Endverbraucher inklusive der dabei erforderlichen Installations-, Inbetriebnahme- und Konfigurationstätigkeiten.

2.2.3.1 Rollen und Aufgaben

Rolle: Erstkonfigurator

Aufgaben in der „Installation + Vor-Ort-Inbetriebnahme des SMGW“:

- Installations-Tätigkeiten (z.B. Einbau des SMGW beim Verbraucher usw.)
- Konfiguration des SMGW (z.B. Parametrierung und Konfiguration der physikalischen Kommunikationsschnittstellen des GW usw.)
- Weitere Inbetriebnahme-Tätigkeiten
- Falls erforderlich: Update der Kommunikationsadresse des GW-Administrators. Hierzu authentisiert sich der Erstkonfigurator gegenüber dem SMGW (unter Nutzung der Erstkonfigurator-PIN) und schaltet damit die lokale Schnittstelle des SMGW (siehe Kap. 2.1.3.1) für die Übergabe der neuen GW-Administrator-Adresse an das SMGW frei.
- Zum Abschluss der Phase „Installation + Vor-Ort-Inbetriebnahme des SMGW“ wird die lokale Schnittstelle des SMGW aus Sicherheitsgründen deaktiviert.

2.2.3.2 Schlüssel- und Zertifikatsmaterial

Folgendes Schlüssel- und Zertifikatsmaterial liegt am Ende der Phase „Installation + Vor-Ort-Inbetriebnahme des SMGW“ (mindestens) im Sicherheitsmodul und GW vor:

Siehe Kap. 2.2.2.2.

2.2.4 Personalisierung des SMGW

Die Phase „Personalisierung des SMGW“ beinhaltet insbesondere die initiale Konfiguration des SMGW durch den GW-Administrator und den Ersatz der vorläufigen GW-Schlüssel aus der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ durch seine Betriebsschlüssel.

2.2.4.1 Rollen und Aufgaben

Rolle: GW-Administrator

Aufgaben in der „Personalisierung des SMGW“:

- Das GW verbindet sich im WAN mit der im GW gespeicherten GW-Administrator-Adresse.
- Aufbau eines TLS-Kanals zwischen GW-Administrator und GW, hierbei:
  - Nutzung der vorläufigen GW-Keys (GW_WAN_TLS_PRV_PRE, GW_WAN_TLS_PUB_PRE) und des zugehörigen Gütesiegel-Zertifikats GW_WAN_TLS_CRT_PRE.
  - Nutzung der GW-Administrator-Keys für TLS. Das Zertifikat GWADM_TLS_CRT liegt bereits aus der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ im GW vor und wurde dort mit seiner Zertifikatskette bis zur Root geprüft. Ferner liegt der Public Key GWADM_TLS_PUB bereits aus der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ im Sicherheitsmodul vor.
- Hinweis: Aus der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ liegen auch bereits die Zertifikate GWADM_SIG_CRT, GWADM_ENC_CRT und GWADM_AUT_CRT im GW vor und wurden dort mit ihrer Zertifikatskette bis zur Root

geprüft, so dass diese ebenfalls nicht mehr personalisiert werden müssen. Auch liegen die zugehörigen Public Keys GWADM_SIG_PUB, GWADM_ENC_PUB und GWADM_AUT_PUB bereits aus der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ im Sicherheitsmodul vor.

- Authentisierung des GW-Administrators gegenüber dem Sicherheitsmodul für die folgenden Administrationstätigkeiten am Sicherheitsmodul unter Nutzung der GW-Administrator-Keys (GWADM_AUT_PRV, GWADM_AUT_PUB).
- Ersetzen der vorläufigen GW-Keys durch Betriebsschlüssel sowie Ersetzen der Gütesiegel-Zertifikate durch Betriebszertifikate aus der SM-PKI über den TLS-Kanal, hierzu:
  - Auf Seiten des GW-Administrators und des GW insbesondere Nutzung der GW-Administrator-Keys (GWADM_SIG_PRV, GWADM_SIG_PUB) und (GWADM_ENC_PRV, GWADM_ENC_PUB) und der zugehörigen SM-PKI-Zertifikate sowie Nutzung der vorläufigen GW-Keys (GW_WAN_SIG_PRV_PRE, GW_WAN_SIG_PUB_PRE) und (GW_WAN_ENC_PRV_PRE, GW_WAN_ENC_PUB_PRE) und der zugehörigen Gütesiegel-Zertifikate für die im folgenden benötigten Administrationskommandos (Sicherung der Kommandos mit Inhaltsdatenverschlüsselung und -signatur).
  - Onboard-Generierung von neuen Key-Paaren für das GW im Sicherheitsmodul (GW-Betriebsschlüssel): (GW_WAN_TLS_PRV, GW_WAN_TLS_PUB), (GW_WAN_SIG_PRV, GW_WAN_SIG_PUB), (GW_WAN_ENC_PRV, GW_WAN_ENC_PUB).
  - Export der Public Keys: GW_WAN_TLS_PUB, GW_WAN_SIG_PUB, GW_WAN_ENC_PUB (die zugehörigen privaten Keys verbleiben im Sicherheitsmodul und können nicht ausgelesen werden).
  - Erstellung des entsprechenden Zertifikatsrequest-Pakets für die Betriebszertifikate zu den neuen GW-Keys und Senden des Zertifikatsrequest-Pakets „an die SM-PKI“.
  - Import der „aus der SM-PKI“ gelieferten Betriebszertifikate GW_WAN_TLS_CRT, GW_WAN_SIG_CRT, GW_WAN_ENC_CRT in das GW und Speicherung im GW.
- Schließen des TLS-Kanals zwischen GW-Administrator und GW.

Mit Abschluss der Personalisierung liegt ein „scharfes“ SMGW vor, das für die Nutzung im Normalbetrieb des Smart Meter-Systems bereitsteht.

Der GW-Administrator kann nachfolgend weitere Administrationstätigkeiten am SMGW bzw. Sicherheitsmodul vornehmen, siehe Kap. 2.2.5.

2.2.4.2 Schlüssel- und Zertifikatsmaterial

Folgendes Schlüssel- und Zertifikatsmaterial liegt am Ende der Phase „Personalisierung des SMGW“ (mindestens) im Sicherheitsmodul und GW vor:

Im Sicherheitsmodul:

- GW-System-PIN (Referenzwert)
- Ggf. Keys für die Speicherverschlüsselung des GW

- Zertifikat ROOT_WAN_SIG_CRT und darin enthaltener Public Key der SM-PKI-Root
- Vorläufige GW-Schlüsselpaare:
  - (GW_WAN_TLS_PRV_PRE, GW_WAN_TLS_PUB_PRE)
  - (GW_WAN_SIG_PRV_PRE, GW_WAN_SIG_PUB_PRE)
  - (GW_WAN_ENC_PRV_PRE, GW_WAN_ENC_PUB_PRE)
- Gütesiegel-Zertifikate:
  - GW_WAN_TLS_CRT_PRE
  - GW_WAN_SIG_CRT_PRE
  - GW_WAN_ENC_CRT_PRE
- Public Keys des GW-Administrators:
  - GWADM_TLS_PUB
  - GWADM_SIG_PUB
  - GWADM_ENC_PUB
  - GWADM_AUT_PUB
- GW-Betriebsschlüsselpaare:
  - (GW_WAN_TLS_PRV, GW_WAN_TLS_PUB)
  - (GW_WAN_SIG_PRV, GW_WAN_SIG_PUB)
  - (GW_WAN_ENC_PRV, GW_WAN_ENC_PUB)

Im GW:

- GW-System-PIN
- Keys für die Speicherverschlüsselung des GW
- Zertifikate des GW-Administrators:
  - GWADM_TLS_CRT
  - GWADM_SIG_CRT
  - GWADM_ENC_CRT
  - GWADM_AUT_CRT
  - Jeweils inkl. der zugehörigen Zertifikatskette (aus der SM-PKI) exklusive des SM-PKI-Root-Zertifikates
- Kommunikationsparameter des GW-Administrators (GW-Administrator-Adresse)
- Betriebszertifikate:
  - GW_WAN_TLS_CRT
  - GW_WAN_SIG_CRT
  - GW_WAN_ENC_CRT

2.2.5 Normalbetrieb (End-Usage) des SMGW

2.2.5.1 Rollen und Aufgaben

Zum Normalbetrieb des SMGW gehören:

- **Administration des SMGW** (durch den GW-Administrator, z.B. Schlüsselmanagement, Aufbringen neuer Profile, ...)
- **Smart Meter Wirkbetrieb** (reguläre Kommunikation des GW mit externen Kommunikationspartnern im WAN, LMN, HAN usw., z.B. zur Übermittlung von Messdaten, …)

2.2.5.1.1 Administration des SMGW

Rolle: GW-Administrator (für die Administration des SMGW)

Administration des SMGW:

- Die Administration des SMGW erfolgt gesichert über einen TLS-Kanal im WAN zwischen GW-Administrator und SMGW.
- Die Administration des Sicherheitsmoduls erfordert zusätzlich eine Authentisierung des GW-Administrators gegenüber dem Sicherheitsmodul.
- Im Rahmen der Administration des SMGW erfolgt die Einrichtung und das weitere Management des für die weiteren Kommunikationsteilnehmer (im WAN, LMN, HAN usw.) erforderlichen Schlüssel- und Zertifikatsmaterials im GW bzw. Sicherheitsmodul.

Hinweis: Bei einem Wechsel des GW-Administrators sind die Schlüssel des neuen GW-Administrators durch den alten GW-Administrator in das GW bzw. Sicherheitsmodul zu importieren.

Bzgl. des Updates von SM-PKI-Root-Zertifikaten: Das Root-Zertifikat stellt einen zentralen Sicherheitsanker dar; seine Integrität und Authentizität ist daher sicherzustellen. Es erfolgt eine vertrauenswürdige Übermittlung des Zertifikates an den GW-Administrator, damit dieser das Update im Sicherheitsmodul vornehmen kann. Die Authentizität des neuen Root-Zertifikates wird über ein sog. Link-Zertifikat sichergestellt, siehe [TR-03109-4].

2.2.5.1.2 Smart Meter Wirkbetrieb

Rolle: Kommunikationspartner im WAN, LMN bzw. HAN (für den Smart Meter Wirkbetrieb)

Für notwendige Aktivitäten eines Service Technikers in der Phase „Normalbetrieb (End-Usage) des SMGW“ kann der Service Techniker die reguläre HAN-Schnittstelle (TLS-gesicherte Schnittstelle) nutzen, wozu zuvor ein entsprechendes Nutzer-Profil mit zugehörigem Schlüsselmaterial durch den GW-Administrator einzurichten ist.

2.2.5.2 Schlüssel- und Zertifikatsmaterial

2.2.5.2.1 Administration des SMGW

Für die Administration des SMGW verwendetes Schlüssel- und Zertifikatsmaterial:

- Betriebsschlüssel und -zertifikate des GW für den TLS-Kanal ins WAN sowie für die Inhaltsdatenverschlüsselung und -signatur (im Sicherheitsmodul bzw. GW hinterlegt)
- Schlüssel und Zertifikate des GW-Administrators (im Sicherheitsmodul bzw. GW hinterlegt)

2.2.5.2.2 Smart Meter Wirkbetrieb

Für den Smart Meter Wirkbetrieb verwendetes Schlüssel- und Zertifikatsmaterial:

- Betriebsschlüssel und -zertifikate des GW für TLS-Kanäle ins WAN, LMN bzw. HAN sowie für die Inhaltsdatenverschlüsselung und -signatur (im Sicherheitsmodul bzw. GW hinterlegt)
- Schlüssel und Zertifikate der externen Kommunikationspartner im WAN, LMN bzw. HAN (im Sicherheitsmodul bzw. GW hinterlegt)

2.2.6 Außerbetriebnahme des SMGW

Für Details zur Außerbetriebnahme des SMGW siehe [TR-03109-1].

Das Sicherheitsmodul stellt für die Außerbetriebnahme des SMGW die Funktionalität zur (irreversiblen) Außerbetriebnahme des Sicherheitsmoduls zur Verfügung, siehe Kap. 3.4.10 und 4.9.

3 File- und Objektsystem, Zugriffsregeln und Kommandoset des Sicherheitsmoduls

3.1 Initialisierung des Sicherheitsmoduls

3.1.1 Initialisierungsverfahren und -kommandos

Die Initialisierung des Sicherheitsmoduls umfasst das Laden fester und kunden- bzw. personenunabhängiger Daten in das Sicherheitsmodul. Insbesondere wird mit dem Initialisierungsfile das initiale File- und Objektsystem der SMGW-Applikation mit seinen Sicherheitsstrukturen und Zugriffsregeln geladen. Ggf. beinhaltet das Initialisierungsfile auch im Rahmen der CC-Zertifizierung des Sicherheitsmoduls evaluierte Patches des Sicherheitsmodul-Betriebssystems.

Initialisierungsprozesse und -kommandos für das Sicherheitsmodul werden Hersteller-spezifisch aufgesetzt und implementiert. Zu den Initialisierungsprozessen und -kommandos erfolgen keine funktionalen oder technischen Vorgaben von Seiten der vorliegenden TR oder anderer Teile der [TR-03109]. Gleichwohl sind aber die Initialisierungsprozesse und -kommandos Gegenstand der CC-Zertifizierung des Sicherheitsmoduls, siehe [PP 0077].

Das Sicherheitsmodul verwaltet einen eigenen Life Cycle-Status. Zum Abschluss der Initialisierungsphase erfolgt für das Sicherheitsmodul der Übergang vom Status „nicht-initialisiert“ zu „initialisiert“ (Hersteller-spezifische Implementierung), siehe Kap. 3.6.1.

3.1.2 Initialisierungsfile

Der Hersteller des Sicherheitsmoduls erstellt ein Initialisierungsfile, mit dem das Sicherheitsmodul initialisiert wird. Das Initialisierungsfile enthält ein auf das Smart Meter-System und das Lebenszyklus-Modell von Sicherheitsmodul und GW zugeschnittenes initiales File- und Objektsystem (SMGW-Applikation) mit vordefinierten DFs, EFs, Key- und PIN-Objekten, die z.T. eine spezielle Vorbelegung aufweisen (siehe unten).

Im Rahmen der Produktion des Sicherheitsmoduls generiert der Hersteller des Sicherheitsmoduls ein Schlüsselpaar (IMP_PRV_TRANS, IMP_PUB_TRANS), mit dem der Import von Public Keys in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ realisiert wird. Dieses Schlüsselpaar ist im Initialisierungsfile hinterlegt und kann je nach Hersteller kundenspezifisch gewählt sein.

Im Auslieferungszustand enthält das initialisierte Sicherheitsmodul das Schlüsselpaar (IMP_PRV_TRANS, IMP_PUB_TRANS) (in einem Key Pair-Objekt) sowie zusätzlich den Public Key IMP_PUB_TRANS (in einem Public Key-Objekt). Hinweis: Grund für die doppelte Speicherung des Public Key ist das Set der verfügbaren Kommandos des Sicherheitsmoduls.

Hinweis: Das Schlüsselpaar (IMP_PRV_TRANS, IMP_PUB_TRANS) hat den Charakter eines Transport-Schlüsselpaares und kann in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ durch den Integrator gegen ein Integrator-eigenes Import-Schlüsselpaar (IMP_PRV, IMP_PUB) ausgetauscht werden. Hierfür sind entsprechende Key-Objekte im Initialisierungsfile vorbereitet.

Für eine detaillierte Beschreibung des Imports von Public Keys in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ siehe [TR-03109-2A].

Mit dem Initialisierungsfile für die Initialisierung des Sicherheitsmoduls wird bzgl. der SMGW-Applikation mindestens folgendes initiale File- und Objektsystem mit folgenden DFs, EFs, Key- und PIN-Objekten geladen:

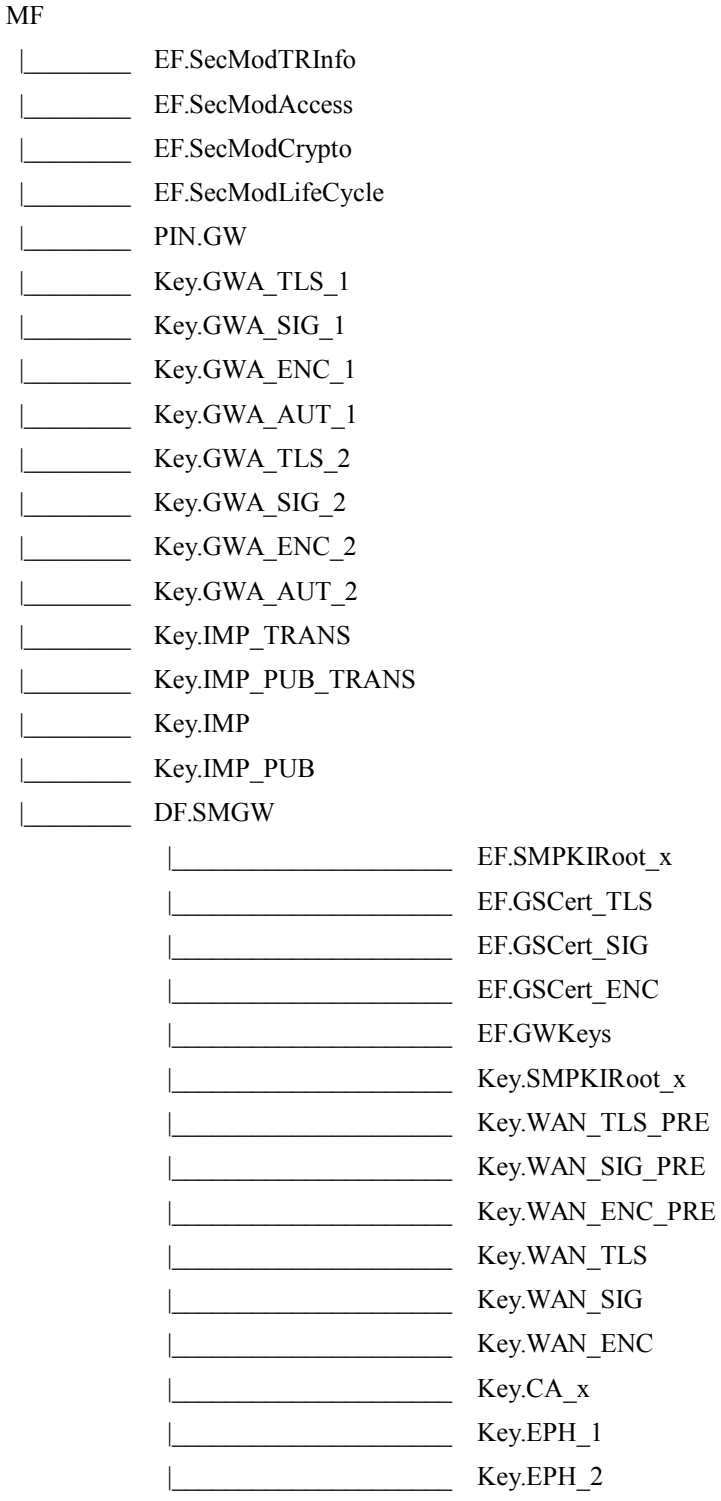

Abbildung 2: Initiales File- und Objektsystem der SMGW-Applikation

Für die vorstehend verwendeten Bezeichnungen für die DFs, EFs, Key- und PIN-Objekte sei auf die folgenden Erklärungen bzw. Kap. 3.2.4, 3.2.5, 3.2.6 und 3.2.7 verwiesen. Für die Beschreibung der im folgenden genannten Key- und PIN-Attribute siehe ebenfalls Kap. 3.2.4 und 3.2.7.

Einträge der Art 'XY' in den folgenden Tabellen kennzeichnen Angaben im HEX-Format.

Hinsichtlich der Vorbelegung der DFs und EFs im Initialisierungsfile gilt:

MF/DF/EF	Beschreibung	File-ID	SFI	Vorbelegung
Masterfile (MF)				
MF	Masterfile.	'3F 00'	---	LCSI: „operational state - activated“
EF.SecModTRInfo	Technisches Datenfeld für die Sicherheitsmodul-relevante Spezifikation. Siehe Kap. 3.2.3.	'01 1A'	'1A'	Nutzdaten: … (Eintrag der zutreffenden Information) LCSI: „operational state - activated“
EF.SecModAccess	Technisches Datenfeld für die Sicherheitsmodul-relevante PACE-Funktionalität. Siehe Kap. 3.2.3, 3.2.3.1.	'01 1B'	'1B'	Nutzdaten: Siehe [TR-03110-3]. LCSI: „operational state - activated“
EF.SecModCrypto	Technisches Datenfeld für die Sicherheitsmodul-relevante Krypto-Funktionalität. Siehe Kap. 3.2.3, 3.2.3.2.	'01 1C'	'1C'	Nutzdaten: … (Eintrag der zutreffenden Information) LCSI: „operational state - activated“
EF.SecModLifeCycle	Technisches Datenfeld für Life Cycle Status-Informationen. Siehe Kap. 3.2.3.	'01 1D'	'1D'	Nutzdaten: leer LCSI: „operational state - activated“
DF.SMGW				
DF.SMGW	DF für die SMGW-Applikation.	'10 01' AID: siehe unten	---	LCSI: „operational state - activated“
EF.SMPKIRoot_x	Datenfelder für die SM-PKI-Root-Zertifikate. Siehe Kap. 3.2.5. x = Nummer des EFs, wobei 1≤x≤10.	Fortlaufende Nummerierung, beginnend mit '01 01', '01 02' usw.	Fortlaufende Nummerierung, beginnend mit '01', '02' usw.	Nutzdaten: leer LCSI: „operational state - activated“
EF.GSCert_TLS	Datenfeld für das Gütesiegel-Zertifikat bzgl. TLS. Siehe Kap. 3.2.5.	'01 11'	'11'	Nutzdaten: leer LCSI: „initialisation“
EF.GSCert_SIG	Datenfeld für das Gütesiegel-Zertifikat bzgl. SIG. Siehe Kap. 3.2.5.	'01 12'	'12'	Nutzdaten: leer LCSI: „initialisation“
EF.GSCert_ENC	Datenfeld für das Gütesiegel-Zertifikat bzgl. ENC. Siehe Kap. 3.2.5.	'01 13'	'13'	Nutzdaten: leer LCSI: „initialisation“

MF/DF/EF	Beschreibung	File-ID	SFI	Vorbelegung
EF.GWKeys	Datenfeld für die symmetrischen GW-Keys. Siehe Kap. 3.2.6.	'01 14'	'14'	Nutzdaten: leer LCSI: „operational state - activated“

Tabelle 3: Initialisierungsfile – MF/DFs/EFs

Das DF.SMGW ist auch direkt über einen Application Identifier (AID) adressierbar. Der AID basiert auf der OID 0.4.0.127.0.7.3.4, so dass sich für den AID des DF.SMGW der Wert 'E8 07 04 00 7F 00 07 03 04' ergibt.

Hinsichtlich der Vorbelegung der Key-Objekte im Initialisierungsfile gilt:

Key-Objekt	Beschreibung	Key-ID / Key-Name	Vorbelegung
Masterfile (MF)			
Key.GWA_TLS_1	Public Key-Objekt für GW-Admin-Schlüssel bzgl. TLS, also GWADM_TLS_PUB.	'00 00 00 11'	Key-Daten: leer Key-Type: „ECC Public Key“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden Key-Curve: leer Key-Usage: DST Key-CryptoAlg: ECDSA ohne Hashing Key-SEID: 01 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.GWA_SIG_1	Public Key-Objekt für GW-Admin-Schlüssel bzgl. SIG, also GWADM_SIG_PUB.	'00 00 00 12'	Key-Daten: leer Key-Type: „ECC Public Key“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden Key-Curve: leer Key-Usage: DST Key-CryptoAlg: ECDSA ohne Hashing und ECDSA mit Hashing Key-SEID: 01 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.GWA_ENC_1	Public Key-Objekt für GW-Admin-Schlüssel bzgl. ENC, also GWADM_ENC_PUB.	'00 00 00 13'	Key-Daten: leer Key-Type: „ECC Public Key“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden

Key-Objekt	Beschreibung	Key-ID / Key-Name	Vorbelegung
			Key-Curve: leer Key-Usage: AT Key-CryptoAlg: ECKA-EG Key-SEID: 01 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.GWA_AUT_1	Public Key-Objekt für GW-Admin-Schlüssel bzgl. AUT, also GWADM_AUT_PUB.	'00 00 00 10'	Key-Daten: leer Key-Type: „ECC Public Key“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden Key-Curve: leer Key-Usage: AT Key-CryptoAlg: ECDSA mit Hashing Key-SEID: 01 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.GWA_TLS_2	Public Key-Objekt für GW-Admin-Schlüssel bzgl. TLS, also GWADM_TLS_PUB.	'00 00 00 21'	Key-Daten: leer Key-Type: „ECC Public Key“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden Key-Curve: leer Key-Usage: DST Key-CryptoAlg: ECDSA ohne Hashing Key-SEID: 01 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.GWA_SIG_2	Public Key-Objekt für GW-Admin-Schlüssel bzgl. SIG, also GWADM_SIG_PUB.	'00 00 00 22'	Key-Daten: leer Key-Type: „ECC Public Key“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden Key-Curve: leer Key-Usage: DST Key-CryptoAlg: ECDSA ohne Hashing und ECDSA mit Hashing Key-SEID: 01 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.GWA_ENC_2	Public Key-Objekt für GW-Admin-Schlüssel bzgl. ENC,	'00 00 00 23'	Key-Daten: leer Key-Type: „ECC Public Key“

Key-Objekt	Beschreibung	Key-ID / Key-Name	Vorbelegung
	also GWADM_ENC_PUB.		Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden Key-Curve: leer Key-Usage: AT Key-CryptoAlg: ECKA-EG Key-SEID: 01 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.GWA_AUT_2	Public Key-Objekt für GW-Admin-Schlüssel bzgl. AUT, also GWADM_AUT_PUB.	'00 00 00 20'	Key-Daten: leer Key-Type: „ECC Public Key“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden Key-Curve: leer Key-Usage: AT Key-CryptoAlg: ECDSA mit Hashing Key-SEID: 01 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.IMP_TRANS	Key Pair-Objekt für Transport-Import-Schlüsselpaar, also (IMP_PRV_TRANS, IMP_PUB_TRANS).	'31'	Key-Daten: ... Key-Type: „ECC Key Pair“ Key-LifeCycleStatus: „operational state - activated“ Key-Storage: persistent Key-UsageCounterInit: … Key-UsageCounter: ... Key-Curve: ... Key-Usage: DST Key-CryptoAlg: ECDSA ohne Hashing Key-SEID: 02 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.1, insbesondere Tabelle 15
Key.IMP_PUB_TRANS	Public Key-Objekt für Transport-Import-Schlüssel, also IMP_PUB_TRANS.	'00 00 00 31'	Key-Daten: ... Key-Type: „ECC Public Key“ Key-LifeCycleStatus: „operational state - activated“ Key-Storage: persistent Key-UsageCounterInit: … Key-UsageCounter: ... Key-Curve: ... Key-Usage: DST Key-CryptoAlg: ECDSA mit Hashing Key-SEID: 02 Zugriff über Key Management- und

Key-Objekt	Beschreibung	Key-ID / Key-Name	Vorbelegung
			Krypto-Kommandos: siehe Kap. 3.3.3.1, insbesondere Tabelle 15
Key.IMP	Key Pair-Objekt für Import-Schlüsselpaar (Integrator-spezifisch), also (IMP_PRV, IMP_PUB).	'32'	Key-Daten: leer Key-Type: „ECC Key Pair“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: … Key-UsageCounter: ... Key-Curve: leer Key-Usage: DST Key-CryptoAlg: ECDSA ohne Hashing Key-SEID: 02 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.1, insbesondere Tabelle 15
Key.IMP_PUB	Public Key-Objekt für Import-Schlüssel (Integrator-spezifisch), also IMP_PUB.	'00 00 00 32'	Key-Daten: leer Key-Type: „ECC Public Key“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: … Key-UsageCounter: ... Key-Curve: leer Key-Usage: DST Key-CryptoAlg: ECDSA mit Hashing Key-SEID: 02 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.1, insbesondere Tabelle 15
DF.SMGW			
Key.SMPKIRoot_x	Public Key-Objekte für die SM-PKI-Root-Keys. x = Nummer des Key-Objektes, wobei 1≤x≤10. Die Anzahl der Key-Objekte entspricht der Anzahl der Datenfelder EF.SMPKIRoot_x (siehe Tabelle 3).	Fortlaufende Nummerierung, beginnend mit '00 00 00 01', '00 00 00 02' usw.	Key-Daten: leer Key-Type: „ECC Public Key“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden Key-Curve: leer Key-Usage: DST Key-CryptoAlg: ECDSA ohne Hashing Key-SEID: 01 und 02 (für x=1) bzw. 01 (für x≥2) Zugriff über Key Management- und Krypto-Kommandos: für SEID = 02 siehe Kap. 3.3.3.1, insbesondere Tabelle 15, für SEID = 01 siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.WAN_TLS_PRE	Key Pair-Objekt für vorläufiges GW-Schlüsselpaar bzgl. TLS, also (GW_WAN_TLS_PRV_PRE, GW_WAN_TLS_PUB_PRE).	'01'	Key-Daten: leer Key-Type: „ECC Key Pair“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent

Key-Objekt	Beschreibung	Key-ID / Key-Name	Vorbelegung
			Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden Key-Curve: leer Key-Usage: DST Key-CryptoAlg: ECDSA ohne Hashing Key-SEID: 01 und 02 Zugriff über Key Management- und Krypto-Kommandos: für SEID = 02 siehe Kap. 3.3.3.1, insbesondere Tabelle 15, für SEID = 01 siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.WAN_SIG_PRE	Key Pair-Objekt für vorläufiges GW-Schlüsselpaar bzgl. SIG, also (GW_WAN_SIG_PRV_PRE, GW_WAN_SIG_PUB_PRE).	'02'	Key-Daten: leer Key-Type: „ECC Key Pair“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden Key-Curve: leer Key-Usage: DST Key-CryptoAlg: ECDSA ohne Hashing Key-SEID: 01 und 02 Zugriff über Key Management- und Krypto-Kommandos: für SEID = 02 siehe Kap. 3.3.3.1, insbesondere Tabelle 15, für SEID = 01 siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.WAN_ENC_PRE	Key Pair-Objekt für vorläufiges GW-Schlüsselpaar bzgl. ENC, also (GW_WAN_ENC_PRV_PRE, GW_WAN_ENC_PUB_PRE).	'03'	Key-Daten: leer Key-Type: „ECC Key Pair“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden Key-Curve: leer Key-Usage: AT Key-CryptoAlg: ECKA-EG und ECDSA ohne Hashing Key-SEID: 01 und 02 Zugriff über Key Management- und Krypto-Kommandos: für SEID = 02 siehe Kap. 3.3.3.1, insbesondere Tabelle 15, für SEID = 01 siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.WAN_TLS	Key Pair-Objekt für GW-Betriebsschlüsselpaar bzgl. TLS, also (GW_WAN_TLS_PRV, GW_WAN_TLS_PUB).	'04'	Key-Daten: leer Key-Type: „ECC Key Pair“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden

Key-Objekt	Beschreibung	Key-ID / Key-Name	Vorbelegung
			Key-Curve: leer Key-Usage: DST Key-CryptoAlg: ECDSA ohne Hashing Key-SEID: 01 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.WAN_SIG	Key Pair-Objekt für GW-Betriebsschlüsselpaar bzgl. SIG, also (GW_WAN_SIG_PRV, GW_WAN_SIG_PUB).	'05'	Key-Daten: leer Key-Type: „ECC Key Pair“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden Key-Curve: leer Key-Usage: DST Key-CryptoAlg: ECDSA ohne Hashing Key-SEID: 01 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.WAN_ENC	Key Pair-Objekt für GW-Betriebsschlüsselpaar bzgl. ENC, also (GW_WAN_ENC_PRV, GW_WAN_ENC_PUB).	'06'	Key-Daten: leer Key-Type: „ECC Key Pair“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden Key-Curve: leer Key-Usage: AT Key-CryptoAlg: ECKA-EG und ECDSA ohne Hashing Key-SEID: 01 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.CA_x	Public Key-Objekte für CA-Public Key (relevant für die Prüfung von Zertifikatsketten im Rahmen der SM-PKI). x = Nummer des Key-Objektes, wobei 1≤x≤10.	Fortlaufende Nummerierung, beginnend mit '00 00 01 01', '00 00 01 02' usw.	Key-Daten: leer Key-Type: „ECC Public Key“ Key-LifeCycleStatus: „initialisation“ Key-Storage: persistent Key-UsageCounterInit: nicht vorhanden Key-UsageCounter: nicht vorhanden Key-Curve: leer Key-Usage: DST Key-CryptoAlg: ECDSA ohne Hashing Key-SEID: 01 und 02 Zugriff über Key Management- und Krypto-Kommandos: für SEID = 02 siehe Kap. 3.3.3.1, insbesondere

Key-Objekt	Beschreibung	Key-ID / Key-Name	Vorbelegung
			Tabelle 15, für SEID = 01 siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.EPH_1	Key Pair-Objekt für ephemerales Schlüsselpaar des GW im Rahmen von ECKA-DH (Variante 2.2), siehe Kap. 4.5.5 a), also (GW_PRV_EPH, GW_PUB_EPH).	'7E'	Key-Daten: leer Key-Type: „ECC Key Pair“ Key-LifeCycleStatus: „operational state - activated“ Key-Storage: temporär Key-UsageCounterInit: Wert 1 Key-UsageCounter: Wert 1 Key-Curve: leer Key-Usage: AT Key-CryptoAlg: ECKA-DH Key-SEID: 01 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.3, insbesondere Tabelle 17
Key.EPH_2	Key Pair-Objekt für ephemerales Schlüsselpaar des GW im Rahmen von ECKA-DH (Variante 2.2), siehe Kap. 4.5.5 a), also (GW_PRV_EPH, GW_PUB_EPH).	'7F'	Key-Daten: leer Key-Type: „ECC Key Pair“ Key-LifeCycleStatus: „operational state - activated“ Key-Storage: temporär Key-UsageCounterInit: Wert 1 Key-UsageCounter: Wert 1 Key-Curve: leer Key-Usage: AT Key-CryptoAlg: ECKA-DH Key-SEID: 01 Zugriff über Key Management- und Krypto-Kommandos: siehe Kap. 3.3.3.3, insbesondere Tabelle 17

Tabelle 4: Initialisierungsfile - Key-Objekte

Zur Notation in vorstehender Tabelle 4:

1) „Key-Daten: ...“ bedeutet, dass konkrete Schlüsseldaten einzutragen sind (Hersteller-spezifische Wahl).
2) „Key-Curve: ...“ bedeutet, dass eine konkrete Elliptische Kurve einzutragen ist (Hersteller-spezifische Wahl).
3) „Key-UsageCounterInit: ...“, „Key-UsageCounter: ...“ bedeuten, dass ein konkreter Wert für den Usage Counter und seinen Initialwert einzutragen ist (Hersteller-spezifische Wahl).

Hinweis: Um im laufenden Betrieb des GW bzw. Sicherheitsmoduls einen Wechsel des GW-Administrators technisch zu unterstützen, werden im Initialisierungsfile zwei Sätze von Public Key-Objekten für den GW-Administrator angelegt (Key.GWA_TLS_x, Key.GWA_SIG_x, Key.GWA_ENC_x, Key.GWA_AUT_x, jeweils x = 1 bzw. 2).

Hinsichtlich der Vorbelegung der PIN-Objekte im Initialisierungsfile gilt:

PIN-Objekt	Beschreibung	PIN-ID	Vorbelegung
PIN.GW	PIN-Objekt für GW-System-PIN (PACE-PIN).	'01'	PIN-Daten: leer PIN-LifeCycleStatus: „initialisation“ PIN-Mindestlänge: 10 Dezimalziffern SEID: 01 und 02 Zugriff über PIN Management- und Krypto-Kommandos: für SEID = 02 siehe Kap. 3.3.3.1, insbesondere Tabelle 15, für SEID = 01 siehe Kap. 3.3.3.3, insbesondere Tabelle 17

Tabelle 5: Initialisierungsfile - PIN-Objekte

3.2 File- und Objektsystem des Sicherheitsmoduls

3.2.1 Übersicht über das File- und Objektsystem

Das Sicherheitsmodul stellt eine SMGW-Applikation, die auf der unterliegenden Betriebssystem-Plattform des Sicherheitsmoduls (HW/SW) mit ihren Funktionalitäten und Sicherheitsmechanismen aufsetzt und auf die Belange des GW bzw. des Smart Meter-Systems zugeschnitten ist, bereit. Diese SMGW-Applikation mit ihren Sicherheitsstrukturen, Zugriffsregeln, Ordnern, Datenfeldern, PIN- und Key-Objekten ist im MF sowie DF.SMGW (und eventuellen späteren zugehörigen Unterordnern), das wiederum direkt unterhalb des MF liegt, angesiedelt.

Eine Übersicht über das File- und Objektsystem des Sicherheitsmoduls im initialen Zustand (mit den mindestens enthaltenen Ordnern, Datenfeldern und Key- und PIN-Objekten sowie ihrer Vorbelegung) wird in Kap. 3.1.2 gegeben. Insbesondere befinden sich direkt im MF das PIN-Objekt für die GW-System-PIN sowie die Key-Objekte für die Administrationsschlüssel des GW-Administrators, während im DF.SMGW (und eventuellen späteren Unterordnern des DF.SMGW) die Key-Objekte für den Wirkbetrieb des SMGW gesammelt werden.

Das DF.SMGW kann über einen FID oder über einen spezifischen AID adressiert werden.

Für Zugriffe auf die im MF bzw. DF.SMGW (und ggf. Unterordnern) liegenden Ordner, Datenfelder, Key- und PIN-Objekte stellt das Sicherheitsmodul ein auf die Belange des SMGW bzw. Smart Meter-Systems zugeschnittenes Set an Kommandos zur Verfügung. Siehe hierzu Kap. 3.4 und 4.

Insbesondere enthält das Sicherheitsmodul im MF durch das GW bzw. den GW-Administrator auslesbare bzw. schreibbare Datenfelder, in denen technische Informationen zum Sicherheitsmodul und seiner SMGW-Applikation hinterlegt sind. Siehe hierzu Kap. 3.2.3.

Für die für das Sicherheitsmodul und seine Ordner, Datenfelder und Key- und PIN-Objekte gesetzten Zugriffsregeln siehe Kap. 3.3.

3.2.2 Ordner und Datenfelder

Das Sicherheitsmodul verwaltet ein hierarchisches, aus Ordnern (MF, DFs), Datenfeldern (EFs) und Key- und PIN-Objekten bestehendes File- und Objektsystem, das sich konform zur [ISO 7816-4] verhält.

Für jedes DF und EF wird ein LCSI gemäß [ISO 7816-4] mitgeführt, der (mindestens) folgende Werte annehmen kann: „initialisation“ / „operational state – activated“ / „operational state – deactivated“ / „terminated“. Von den Zuständen „operational state – activated“ und „operational state – deactivated“ gibt es keinen Schritt zurück auf den Wert „initialisation“, und vom Zustand „terminated“ gibt es keinen Schritt zurück auf einen der übrigen Werte.

Übergänge zwischen den verschiedenen Werten für den LCSI eines DFs oder EFs durch die Ausführung eines auf das betreffende Objekt zugreifenden Kommandos werden abgesehen von den zuvor gemachten Vorgaben und sofern von Seiten der TR in den Kap.3.4 und 4 zu den Kommando-Spezifikationen keine weitere explizite Vorgabe erfolgt, Hersteller-spezifisch festgelegt und implementiert.

3.2.3 Technische Datenfelder

Das Sicherheitsmodul enthält im MF insbesondere durch das GW bzw. den GW-Administrator auslesbare bzw. schreibbare Datenfelder, in denen technische Informationen zum Sicherheitsmodul und seiner SMGW-Applikation hinterlegt sind:

- EF.SecModTRInfo: Record-orientiertes Datenfeld mit technischen Informationen zu der für das vorliegende Sicherheitsmodul relevanten Spezifikation (insbesondere Angabe der Version der vorliegenden TR, auf Basis derer das vorliegende Sicherheitsmodul implementiert wurde)
- EF.SecModAccess: Transparentes Datenfeld mit technischen Informationen zu der vom vorliegenden Sicherheitsmodul angebotenen PACE-Funktionalität; hier Speicherung der SecurityInfos-Datei (für weitere Detailinformationen zu diesem Datenfeld siehe Kap. 3.2.3.1)
- EF.SecModCrypto: Transparentes Datenfeld mit technischen Informationen zu der vom vorliegenden Sicherheitsmodul angebotenen Krypto-Funktionalität; hier Speicherung der KryptoSecurityInfos-Datei (für weitere Detailinformationen zu diesem Datenfeld siehe Kap. 3.2.3.2)
- EF.SecModLifeCycle: Record-orientiertes Datenfeld mit technischen Informationen zum Life Cycle-Status des vorliegenden Sicherheitsmoduls (für weitere Detailinformationen zu diesem Datenfeld siehe Kap. 3.6.1)

Die Technischen Datenfelder EF.SecModTRInfo, EF.SecModAccess und EF.SecModCrypto zum Hinterlegen von Informationen zu der für das Sicherheitsmodul relevanten Spezifikation und zu der vom Sicherheitsmodul unterstützten PACE- und Krypto-Funktionalität werden insbesondere im Hinblick auf mögliche spätere Migrationsschritte im Smart Meter-System als sinnvoll erachtet. Für die Verwendung des Technischen Datenfeldes EF.SecModLifeCycle siehe Kap. 3.6.1.

Der Zugriff auf die Technischen Datenfelder erfolgt über die üblichen Kommandos zum Zugriff auf Datenfelder, siehe Kap. 3.4.4.

Für die für die Technischen Datenfelder gesetzten Zugriffsregeln siehe Kap. 3.3.

3.2.3.1 Technisches Datenfeld zur PACE-Funktionalität

Das Sicherheitsmodul beinhaltet im MF eine auslesbare Info-Datei zu der vom Sicherheitsmodul unterstützten PACE-Funktionalität und ihren Protokollparametern.

Für die im folgenden verwendeten Bezeichnungen der Info-Datei siehe [TR-03110-3], Abschnitt A.1.

SecurityInfo-Datei:

- Die Datei enthält ein oder mehrere PACEInfo-Dateien (siehe unten).
- Die SecurityInfo-Datei kann vom GW ausgelesen werden, um Informationen darüber zu erhalten, welche PACE-Protokollparameter das Sicherheitsmodul prinzipiell unterstützt.

Das Sicherheitsmodul kann mehrere SecurityInfo-Dateien enthalten, die in der SecurityInfos-Datei zusammengefasst sind. Die SecurityInfos-Datei wird im EF.SecModAccess (siehe Kap. 3.2.3) gespeichert.

PACEInfo-Datei:

- Die Datei beinhaltet Informationen zur Implementierung von PACE und seinen Protokollparametern.
- Inhalte der Datenstruktur:
  - OID des PACE-Protokolls mit seinen Protokollparametern (siehe auch [TR-03109-3])
  - Protokoll-Version
  - Parameter-ID = ID der Elliptischen Kurve (Standardized Domain Parameter, gemäß [TR-03110-3], Abschnitt A.2.1.1)

Hinweis: Das für das Smart Meter-System verwendete PACE-Protokoll nutzt ausschließlich das sog. Generic Mapping, siehe auch [TR-03109-3] bzw. [TR-03116-3].

Es ist keine Signatur über die SecurityInfos-Datei und kein alternativer Sicherungsmechanismus für die Authentizität dieser Datei explizit vorgesehen. Jedoch ist die SecurityInfos-Datei Betriebssystem-intern integritätsgeschützt abzulegen (Hersteller-spezifische Implementierung).

3.2.3.2 Technisches Datenfeld zur Krypto-Funktionalität

Das Sicherheitsmodul beinhaltet im MF eine auslesbare Info-Datei zu den vom Sicherheitsmodul unterstützten Krypto-Algorithmen und Elliptischen Kurven.

KryptoSecurityInfo-Datei:

- Die Datei enthält ein oder mehrere KryptoInfo-Dateien (siehe unten).
- Die KryptoSecurityInfo-Datei kann vom GW ausgelesen werden, um Informationen darüber zu erhalten, welche Krypto-Algorithmen und Elliptischen Kurven das Sicherheitsmodul prinzipiell unterstützt.

Das Sicherheitsmodul kann mehrere KryptoSecurityInfo-Dateien enthalten, die in der KryptoSecurityInfos-Datei zusammengefasst sind. Die KryptoSecurityInfos-Datei wird im EF.SecModKrypto (siehe Kap. 3.2.3) gespeichert.

KryptoInfo-Datei:

- Die Datei beinhaltet Informationen zum Krypto-Algorithmus und zur Elliptischen Kurve.
- Inhalte der Datenstruktur:

- OID des Krypto-Algorithmus
- Parameter-ID = OID der Elliptischen Kurve (Standardized Domain Parameter)

Hinweis: Die SecurityInfos-Datei ist als eigenständige Info-Datei ausspezifiziert, und es ist keine Zusammenfassung der KryptoSecurityInfos-Datei und SecurityInfos-Datei (siehe Kap. 3.2.3.1) vorgesehen. Die KryptoSecurityInfos-Datei wird als eigenständige Datei spezifiziert (in Anlehnung an die Spezifikation der SecurityInfos-Datei).

Es ist keine Signatur über die KryptoSecurityInfos-Datei und kein alternativer Sicherungsmechanismus für die Authentizität dieser Datei explizit vorgesehen. Jedoch ist die KryptoSecurityInfos-Datei Betriebssystem-intern integritätsgeschützt abzulegen (Hersteller-spezifische Implementierung).

3.2.4 Sicherheitsmodul als Speicher und Nutzer asymmetrischer Schlüssel

3.2.4.1 Schlüsselkonzept

Für das Smart Meter-System wird die Zielsetzung eines möglichst flexiblen Schlüsselkonzeptes verfolgt. Das Sicherheitsmodul setzt hierzu auf der Idee von Key-Objekten auf, für die das Sicherheitsmodul entsprechende Key Management-Funktionen, insbesondere zum Anlegen, Löschen, Aktivieren und Deaktivieren von Key-Objekten bereitstellt.

Prinzipiell geht das GW mit folgendem Schlüsselmaterial um:

a) GW-Key-Paare:
   - Die Schlüsselpaare des GW werden im Sicherheitsmodul gespeichert, wohingegen die Zertifikate der öffentlichen Keys des GW – mit Ausnahme der Gütesiegel-Zertifikate – im GW gespeichert werden. Insbesondere werden die Zertifikate des GW zu seinen Betriebsschlüsseln im GW hinterlegt.
   - Alle GW-Schlüsselpaare werden onboard im Sicherheitsmodul generiert.
   - Die GW-Schlüsselpaare für TLS (WAN), SIG und ENC werden in der SM-PKI zertifiziert.

b) Keys der externen Welt (z.B. des GW-Administrators, der WAN-Kommunikationsteilnehmer usw.):
   - Die Schlüsselpaare der externen Welt für TLS, SIG, ENC und AUT werden durch diese selbst generiert.
   - Die Zertifikate zu den Schlüsseln entstammen der jeweils zugehörigen PKI (z.B. für die WAN-Kommunikation also der SM-PKI). Sämtliche Schlüssel des GW-Administrators werden in der SM-PKI zertifiziert.
   - Die Zertifikate mit den öffentlichen Keys der externen Welt werden im GW gespeichert; die privaten Keys verbleiben bei der externen Welt.
   - Für die Nutzung der öffentlichen Keys der externen Welt in Krypto-Operationen werden diese Keys in das Sicherheitsmodul importiert, entweder innerhalb des jeweiligen Krypto-Kommandos oder aber vorab durch ein entsprechendes Import-Kommando.

c) Über die zuvor in a) und b) genannten Schlüssel hinaus generiert und verarbeitet das Sicherheitsmodul (im Rahmen der Unterstützung des GW beim TLS Handshake) Diffie-Hellman-Schlüssel (DH).

3.2.4.2 Key-Objekte

Für die im Sicherheitsmodul gespeicherten und verwalteten Schlüssel gilt Folgendes:

- Speicherformen von Keys:
  - Persistent oder temporär im Sicherheitsmodul gespeicherte Key-Paare und Public Keys werden vom Betriebssystem des Sicherheitsmoduls in Form von Key-Objekten abgelegt.
  - Hinweise: Die Speicherorganisation der im Sicherheitsmodul gespeicherten Key-Objekte fällt generell unter die Administration des Sicherheitsmoduls.

    Die im Smart Meter-System *im Wirkbetrieb* des SMGW verwendeten Key-Paare und Public Keys werden der SMGW-Applikation des Sicherheitsmoduls zugeordnet. Die Ablage persistent gespeicherter Keys erfolgt hierbei direkt im DF.SMGW selbst und/oder in Unterordnern des DF.SMGW.

    Im Rahmen der Administration des Sicherheitsmoduls ist darauf zu achten, dass eine Ablage der zum Smart Meter-System zugehörigen Key-Objekte für den Wirkbetrieb im DF.SMGW selbst und/oder in geeigneten Unterordnern des DF.SMGW erfolgt. Hierbei kann durch den GW-Administrator eine bedarfsgerechte Verwaltung der Ordnerstruktur für die Key-Objekte im Rahmen seiner Administration des SMGW erfolgen. Auch kann unter gewissen Randbedingungen (siehe hierzu Kap. 5) das Initialisierungsfile wie es in Kap. 3.1.2 vordefiniert ist bereits um weitere Ordner und Key-Objekte geeignet für den späteren Wirkbetrieb ergänzt werden.

    Ferner ist im Rahmen der Administration des Sicherheitsmoduls darauf zu achten, dass außerhalb des Smart Meter-Systems verwendete Key-Objekte außerhalb der SMGW-Applikation abgelegt werden.

    Die im Smart Meter-System *für die Administration* des Sicherheitsmoduls verwendeten Public Keys des GW-Administrators werden zentral dem MF zugeordnet.
  - Unterschieden werden je nach Speicherort globale und lokale Key-Objekte (im Sinne der [ISO 7816-4]).
  - Die Sicherheitsmodul-interne Speicherung der Key-Objekte erfolgt Betriebssystem-spezifisch, d.h. es erfolgt keine weitere technische Vorgabe zu den Betriebssystem-internen Speicherstrukturen und -mechanismen von Seiten der vorliegenden TR.
- Key-Objekte werden im Sicherheitsmodul integritätsgeschützt abgelegt. Vor der Nutzung eines Key-Objektes wird dieses vom Betriebssystem auf Integrität geprüft. Siehe hierzu auch die Anforderung nach Integritätsschutz im Protection Profile [PP 0077] für das Sicherheitsmodul.
- Folgende Typen von Key-Objekten werden unterschieden:
  - Key Pair-Objekte (zur persistenten/temporären Speicherung von Schlüsselpaaren)
  - Public Key-Objekte (zur persistenten/temporären Speicherung von öffentlichen Schlüsseln)

- Referenzierung von Key-Objekten:
  - Key Pair-Objekte: Key-ID von 1 Byte Länge.
  - Public Key-Objekte: Key-Name von 4-8 Byte Länge.
  - Für Kommandos, die auf ein Key-Objekt zugreifen, wird das betreffende Key-Objekt über die sog. Key Reference referenziert. Für den Zusammenhang zwischen Key Reference und Key-ID bei Key Pair-Objekten bzw. Key-Name bei Public Key-Objekten siehe Kap. 3.2.4.2.1 bzw. 3.2.4.2.2.
- Schlüsselsuche von Key-Objekten:
  - Die Schlüsselsuche erfolgt über die Key Reference und ggf. die Anwendungsklasse des Key-Objektes (je nach Hersteller-spezifischer Implementierung des Betriebssystems des Sicherheitsmoduls).
  - Die Implementierung der Schlüsselsuche erfolgt Betriebssystem-spezifisch.
  - In Kommandos, in denen Key-Objekte zu referenzieren sind, wird neben der Key Reference auch der Parameter AT / DST zur Anzeige der Anwendungsklasse des betreffenden Key-Objektes mitgegeben. Ob das jeweilige Betriebssystem des Sicherheitsmoduls diese Information bei der Schlüsselsuche auswertet oder nicht, hängt von der Implementierung des Betriebssystems ab.
  - Die Schlüsselsuche beginnt bei lokaler Suche im aktuell selektierten DF. Wird das gesuchte Key-Objekt unter der Key Reference, ggf. zusammen mit der Anwendungsklasse, im selektierten DF nicht gefunden, so bricht die Suche ab oder wird sukzessive in den übergeordneten DFs fortgesetzt (Hersteller-spezifische Implementierung).

    Bei globaler Suche erfolgt die Schlüsselsuche im MF, ebenfalls unter der Key Reference, ggf. zusammen mit der Anwendungsklasse.

    Für Key Pair-Objekte ist sowohl eine lokale wie auch globale Suche möglich (Anzeige über die Key Reference).

    Für Public Key-Objekte beginnt die Suche immer im aktuell selektierten DF (lokale Suche).
  - Bei der Referenzierung von Key-Objekten in Kommandos werden Key Pair-Objekte mit Tag '84' und Public Key-Objekte mit Tag '83' referenziert. Die Anwendungsklasse des Key-Objektes wird über ein entsprechendes CRT-Template (AT / DST) angezeigt.
- Management von Key-ID/Key-Name:
  - Für Key Pair-Objekte:

    Das Betriebssystem erkennt und lehnt i.a. die mehrfache Vergabe von Key-IDs innerhalb desselben Ordners für im Sicherheitsmodul gespeicherte Key Pair-Objekte ab. Hersteller-spezifisch können ggf. aber auch doppelt vergebene Key-IDs für Key Pair-Objekte im selben Ordner zulässig sein, sofern sich die Keys in ihrer Anwendungsklasse (AT / DST im Key-Attribut Key-Usage) unterscheiden.
  - Für Public Key-Objekte:

    Das Betriebssystem erkennt die doppelte Vergabe von Key-Names im Pfad eines Public Key-Objektes (auch über Anwendungsklassen von Public Key-Objekten hinweg) und

lehnt eine solche doppelte Vergabe bei der Ausführung des Kommandos CREATE KEY ab.

- Hinweis: Zum Auffinden von im Sicherheitsmodul persistent gespeicherten Key-Objekten kann z.B. eine Cryptographic Information Application (DF.CIA) nach ISO/IEC 7816-15 (siehe auch [EN 14890-1]) verwendet werden (optional). Im DF.CIA kann eine Übersicht über die im Sicherheitsmodul vergebenen Key-ID/Key-Name hinterlegt werden.

Die beiden folgenden Kapitel enthalten eine detailliertere Beschreibung der Key-Objekte, gegliedert nach den beiden Typen Key Pair-Objekt und Public Key-Objekt.

3.2.4.2.1 Key Pair-Objekte

Für die im Sicherheitsmodul gespeicherten und verwalteten Key Pair-Objekte gilt Folgendes:

- Es erfolgt ausschließlich eine onboard-Generierung von Key-Paaren im Sicherheitsmodul, die dann in einem Key Pair-Objekt abgelegt werden.
- Key Pair-Objekte werden persistent oder temporär im Sicherheitsmodul gespeichert.

  Hinweis: Im Smart Meter-System *im Wirkbetrieb* verwendete Key Pair-Objekte werden im DF.SMGW selbst und/oder in Unterordnern des DF.SMGW abgelegt.
- Im Key Pair-Objekt gespeichert werden die Daten des Key-Paares:
  - Private Key-Daten (Zufallszahl aus dem zugelassenen Wertebereich)
  - Public Key-Daten (Kurvenpunkt) (Speicherung optional)
- Bei der Speicherung von Key-Paaren erfolgt Hersteller-spezifisch die Ablage des Public Key-Parts oder nicht. Erfolgt keine Speicherung des Public Key im Sicherheitsmodul, so ist dieser, wenn er z.B. in Krypto-Kommandos benötigt wird, vom Betriebssystem erneut aus dem Private Key-Part zu berechnen.
- Gespeichert werden mindestens folgende Key-Paar-Zusatzinformationen (Key-Attribute):
  - Key-ID (1 Byte Länge): zur Identifikation des Key Pair-Objektes, relevant für die Schlüsselsuche

    Für die eigentliche Schlüsselnummer stehen nur die untersten 7 Bit in der Key-ID zur Verfügung. Wird in einem Kommando ein Key Pair-Objekt referenziert, so wird im Sinne der [ISO 7816-4] nicht die Key-ID, sondern die sog. Key Reference übergeben. Die Key Reference stimmt in ihren untersten 7 Bit mit den untersten 7 Bit der Key-ID überein, während über das MSBit der Key Reference die Steuerung der Schlüsselsuche (1 für lokale Suche, 0 für globale Suche) erfolgt.
  - Key-Type: hier „ECC Key Pair“
  - Key-LifeCycleStatus: Key-Paar „initialisation“ / „operational state – activated“ / „operational state – deactivated“ (LCSI des Key Pair-Objektes)

    Für temporäre Key-Paare wird im Sicherheitsmodul nur der Wert „operational state – activated“ genutzt.
  - Key-Storage: persistente / temporäre Speicherung des Key-Paars

- Key-UsageCounterInit: Initialwert eines Bedienungszählers für das Key-Paar (optionales Attribut)
- Key-UsageCounter: Aktueller Wert des Bedienungszählers für das Key-Paar (optionales Attribut)
- Key-Curve: Informationen zu den zum Key-Paar zugehörigen Kurvenparametern (OID oder Hersteller-spezifische Codierung/Speicherung, siehe hierzu auch Kap. 3.2.4.4 und 3.4.6.1)
- Key-Usage: Informationen zum Verwendungszweck bzw. zur Anwendungsklasse des Key-Paares, hier AT = authentication bzw. DST = digital signature; ggf. relevant für die Schlüsselsuche (Hersteller-spezifische Implementierung der Schlüsselsuche)
- Key-CryptoAlg: Informationen zum Krypto-Algorithmus (ohne Bezug zur Elliptischen Kurve bzw. Schlüssellänge), für den das Key-Paar eingesetzt werden darf; hier ECDSA ohne Hashing / ECKA-EG / ECKA-DH, siehe Kap. 3.2.4.3, insbesondere Tabelle 6, und Kap. 3.2.4.5, [TR-03109-3], [TR-03111] bzw. [EN 14890-1] (OID oder Hersteller-spezifische Codierung/Speicherung, siehe hierzu auch Kap. 3.4.6.1)

  Für ein Key Pair-Objekt müssen in Key-CryptoAlg mehrere Krypto-Algorithmen eingetragen werden können.
- Key-SEID: Information, in welchem Security Environment (SE) das Key-Paar nutzbar ist

- Für temporäre Key Pair-Objekte, deren Key-UsageCounter zu Beginn der Ausführung eines dieses Key Pair-Objekt nutzenden Kommandos einen Wert trägt, der eine einmalige weitere Nutzung dieses Key Pair-Objektes erlaubt, werden die Schlüsseldaten des Key Pair-Objektes nach ihrer Nutzung im Rahmen der betreffenden Kommando-Ausführung gelöscht.
- Eine Auswertung des Key-Attributs Key-CryptoAlg eines Key Pair-Objektes erfolgt im Rahmen der Ausführung des auf das Key Pair-Objekt zugreifenden Krypto-Kommandos bzw. des vorhergehenden Kommandos zum Setzen der Schlüsselreferenz.
- Unterstützt werden die Krypto-Kommandos GENERATE ASYMMETRIC KEY PAIR, PSO COMPUTE DIGITAL SIGNATURE, GENERAL AUTHENTICATE und INTERNAL AUTHENTICATE.

3.2.4.2.2 Public Key-Objekte

Für die im Sicherheitsmodul gespeicherten und verwalteten Public Key-Objekte gilt Folgendes:

- Hinweis: Die Generierung von Key-Paaren, deren Public Key-Part in einem Public Key-Objekt im Sicherheitsmodul abgelegt wird, erfolgt ausschließlich extern.
- Public Key-Objekte werden persistent oder temporär im Sicherheitsmodul gespeichert.

  Hinweis: Für im Smart Meter-System *im Wirkbetrieb* verwendete Public Key-Objekte erfolgt bei persistenter Ablage die Speicherung im DF.SMGW selbst und/oder in Unterordnern des DF.SMGW; Administrationsschlüssel des GW-Administrators *für die Administration* des Sicherheitsmoduls werden zentral im MF angesiedelt.
- Im Public Key-Objekt gespeichert werden die Daten des Public Key:
  - Public Key-Daten (Kurvenpunkt)

- Für persistente Public Key-Objekte werden mindestens folgende Public Key-Zusatzinformationen (Key-Attribute) gespeichert:
  - Key-Name (4-8 Byte Länge): zur Identifikation des Public Key-Objektes, relevant für die Schlüsselsuche

    Für die Schlüsselnummer steht Key-Name komplett zur Verfügung. Wird in einem Kommando ein Public Key-Objekt referenziert, so wird im Sinne der [ISO 7816-4] nicht Key-Name, sondern die sog. Key Reference übergeben. Die Key Reference stimmt aber mit Key-Name überein.
  - Key-Type: hier „ECC Public Key“
  - Key-LifeCycleStatus: Public Key „initialisation“ / „operational state – activated“ / „operational state – deactivated“ (LCSI des Public Key-Objektes)
  - Key-Storage: persistente / temporäre Speicherung des Public Keys
  - Key-UsageCounterInit: Initialwert eines Bedienungszählers für den Public Key (optionales Attribut)
  - Key-UsageCounter: Aktueller Wert des Bedienungszählers für den Public Key (optionales Attribut)
  - Key-Curve: Informationen zu den zum Public Key zugehörigen Kurvenparametern (OID oder Hersteller-spezifische Codierung/Speicherung, siehe hierzu auch Kap. 3.2.4.4 und 3.4.6.1)
  - Key-Usage: Informationen zum Verwendungszweck bzw. zur Anwendungsklasse des Key-Paares, hier AT = authentication bzw. DST = digital signature; ggf. relevant für die Schlüsselsuche (Hersteller-spezifische Implementierung der Schlüsselsuche)
  - Key-CryptoAlg: Informationen zum Krypto-Algorithmus (ohne Bezug zur Elliptischen Kurve bzw. Schlüssellänge, für den der Public Key eingesetzt werden darf; hier ECDSA ohne Hashing / ECDSA mit Hashing / ECKA-EG / ECKA-DH, siehe Kap. 3.2.4.3, insbesondere Tabelle 6, und Kap. 3.2.4.5, [TR-03109-3], [TR-03111] bzw. [EN 14890-1] (OID oder Hersteller-spezifische Codierung/Speicherung, siehe hierzu auch Kap. 3.4.6.1)

    Für ein Public Key-Objekt müssen in Key-CryptoAlg mehrere Krypto-Algorithmen eingetragen werden können.
  - Key-SEID: Information, in welchem Security Environment (SE) der Public Key nutzbar ist
- Eine Auswertung des Key-Attributs Key-CryptoAlg eines Public Key-Objektes erfolgt im Rahmen der Ausführung des auf das Public Key-Objekt zugreifenden Krypto-Kommandos bzw. des vorhergehenden Kommandos zum Setzen der Schlüsselreferenz.
- Unterstützt werden die Krypto-Kommandos PSO VERIFY DIGITAL SIGNATURE, PSO VERIFY CERTIFICATE, GENERAL AUTHENTICATE und EXTERNAL AUTHENTICATE.

3.2.4.2.3 Key-LifeCycleStatus

Mittels des Kommandos CREATE KEY wird ein persistentes Key-Objekt mit dem Wert „initialisation“ für das Key-Attribut Key-LifeCycleStatus angelegt; das Key-Objekt ist hierbei mit

seinen Betriebssystem-internen Speicherstrukturen angelegt, aber noch nicht mit Schlüsseldaten gefüllt. Im Falle eines Key Pair-Objektes wird dieses über das Kommando GENERATE ASYMMETRIC KEY PAIR mit Schlüsseldaten gefüllt und in den Status „operational state – activated“ versetzt. Im Falle eines Public Key-Objektes wird dieses über das Kommando PSO VERIFY CERTIFICATE mit Schlüsseldaten gefüllt und in den Status „operational state – activated“ überführt. Über das Kommando DEACTIVATE KEY kann ein solches (gefülltes) Key-Objekt explizit deaktiviert werden.

Ein persistentes Key Pair-Objekt im Zustand „initialisation“ ist für die Verwendung in Krypto-Kommandos mit Ausnahme des Kommandos GENERATE ASYMMETRIC KEY PAIR (in der Variante Schlüsselgenerierung mit/ohne Ausgabe des Public Key) gesperrt. Ferner ist ein persistentes Key Pair-Objekt im Zustand „operational state – deactivated“ für die Verwendung in Krypto-Kommandos mit Ausnahme des Kommandos GENERATE ASYMMETRIC KEY PAIR (in der Variante Schlüsselgenerierung mit/ohne Ausgabe des Public Key sowie in der Variante Ausgabe des Public Key ohne Schlüsselgenerierung) gesperrt. Über ein GENERATE ASYMMETRIC KEY PAIR kann ein persistentes Key Pair-Objekt im Zustand „initialisation“ oder „operational state – deactivated“ mit (neuen) Schlüsseldaten gefüllt werden. Ein Befüllen eines persistenten Key Pair-Objektes im Zustand „operational state – activated“ ist nicht möglich.

Ein persistentes Public Key-Objekt im Zustand „initialisation“ oder „operational state – deactivated“ ist für die Verwendung in Krypto-Kommandos mit Ausnahme des Kommandos PSO VERIFY CERTIFICATE, sofern es sich um das zu befüllende Public Key-Objekt handelt, gesperrt. Über ein PSO VERIFY CERTIFICATE kann ein persistentes Public Key-Objekt im Zustand „initialisation“ oder „operational state – deactivated“ mit (neuen) Schlüsseldaten gefüllt werden. Ein Befüllen eines persistenten Public Key-Objektes im Zustand „operational state – activated“ ist nicht möglich.

Für temporäre Key Pair-Objekte wird nach Kap. 3.2.4.2.1 nur der Key-LifeCycleStatus „operational state – activated“ genutzt, so dass diese Key Pair-Objekte stets über das Kommando GENERATE ASYMMETRIC KEY PAIR mit (neuen) Schlüsseldaten gefüllt werden können.

Von den Zuständen „operational state – activated“ und „operational state – deactivated“ gibt es keinen Schritt zurück auf den Wert „initialisation“.

3.2.4.3 Klassifikation der Schlüssel

Folgende Schlüsselarten werden im Smart Meter-System bzgl. ihrer Verwendung unterschieden:

TLS-Keys: Schlüssel für TLS

SIG-Keys: Schlüssel für Inhaltsdatensignatur und sonstige Signaturen (z.B. der PKI-Root-CA oder von Sub-CAs der PKI)

ENC-Keys: Schlüssel für Inhaltsdatenverschlüsselung (ElGamal Key Agreement)

AUT-Keys: Schlüssel für (externe) Authentisierung

DH-Keys: Schlüssel für Diffie-Hellman Key Agreement

Folgende Tabelle 6 gibt eine detaillierte Übersicht über die im Sicherheitsmodul verwendeten Schlüsselarten, deren Klassifikation und diesbzgl. relevanten Key-Attribute:

Schlüsselart	**Schlüsseltyp (Key-Attribut Key-Type)**	**Speicherart (Key-Attribut Key-Storage)**	**Anwendungsklasse (Key-Attribut Key-Usage)**	**Krypto-Algorithmus (Key-Attribut Key-CryptoAlg) (siehe Tabelle 7)**
Administrationsschlüssel des GW-Administrators				
TLS-Key	Public Key-Objekt	persistent	DST	ECDSA ohne Hashing: id-ecdsa-plain-signatures
SIG-Key	Public Key-Objekt	persistent	DST	ECDSA ohne Hashing: id-ecdsa-plain-signatures und ECDSA mit Hashing: id-ecdsa-plain-SHA
ENC-Key	Public Key-Objekt	persistent	AT	ECKA-EG: id-ecka-eg
AUT-Key	Public Key-Objekt	persistent	AT	ECDSA mit Hashing: id-ecdsa-plain-SHA
Schlüssel des SMGW				
TLS-Key	Key Pair-Objekt	persistent	DST	ECDSA ohne Hashing: id-ecdsa-plain-signatures
SIG-Key	Key Pair-Objekt	persistent	DST	ECDSA ohne Hashing: id-ecdsa-plain-signatures
ENC-Key	Key Pair-Objekt	persistent	AT	ECKA-EG: id-ecka-eg und ECDSA ohne Hashing: id-ecdsa-plain-signatures
DH-Key	Key Pair-Objekt	temporär Hinweis: Verwendung nur für die Kommandos GENERATE ASYMMETRIC KEY PAIR und GENERAL AUTHENTICATE / ECKA-DH (Var. 2.2)	AT	ECKA-DH: id-ecka-dh
Schlüssel der externen Welt (WAN, LMN, HAN)				
TLS-Key	Public Key-Objekt	persistent oder Übergabe im Kommando	DST	ECDSA ohne Hashing: id-ecdsa-plain-signatures
SIG-Key	Public Key-Objekt	persistent oder Übergabe im Kommando	DST	ECDSA ohne Hashing: id-ecdsa-plain-signatures
ENC-Key	Public Key-Objekt	persistent oder Übergabe im Kommando	AT	ECKA-EG: id-ecka-eg

Schlüsselart	Schlüsseltyp (Key-Attribut Key-Type)	Speicherart (Key-Attribut Key-Storage)	Anwendungsklasse (Key-Attribut Key-Usage)	Krypto-Algorithmus (Key-Attribut Key-CryptoAlg) (siehe Tabelle 7)
DH-Key	Public Key-Objekt	Übergabe im Kommando GENERAL AUTHENTICATE / ECKA-DH (Var. 2.1 und 2.2)	AT	ECKA-DH: id-ecka-dh
PKI-Schlüssel (Root-CA, Sub-CAs)				
SIG-Key	Public Key-Objekt	persistent	DST	ECDSA ohne Hashing: id-ecdsa-plain-signatures
Import-Schlüssel (nur relevant für SEID = 02)				
SIG-Key	Key Pair-Objekt	persistent	DST	ECDSA ohne Hashing: id-ecdsa-plain-signatures
SIG-Key	Public Key-Objekt	persistent	DST	ECDSA mit Hashing: id-ecdsa-plain-SHA

Tabelle 6: Klassifikation der Schlüssel

Zur Notation in vorstehender Tabelle 6:

In der Spalte „Krypto-Algorithmus“ ist für das Key-Attribut Key-CryptoAlg nur der jeweils relevante OID aus Tabelle 7 vermerkt. Wie in den Kap. 3.2.4.2.1 und 3.2.4.2.2 angegeben, kann der Eintrag im Key-Attribut Key-CryptoAlg aber auch durch eine entsprechende Hersteller-spezifische Codierung realisiert werden.

Hinweis: Für die Erstellung der Zertifikatsrequests zu TLS- und SIG-Keys des SMGW wird für die Erzeugung der inneren Signatur das Kommando PSO COMPUTE DIGITAL SIGNATURE verwendet. Für die Erstellung der Zertifikatsrequests zu ENC-Keys des SMGW wird für die Erzeugung der inneren Signatur das Kommando INTERNAL AUTHENTICATE verwendet. Die Erstellung der Zertifikatsrequests zu TLS-, SIG-, ENC- und AUT-Keys der externen Welt (GW-Administrator, WAN, LMN, HAN) erfolgt außerhalb von GW und Sicherheitsmodul.

3.2.4.4 Domain Parameter Elliptischer Kurven

Für die im Sicherheitsmodul verwendeten Elliptischen Kurven gilt:

- Implementiert werden im Sicherheitsmodul alle Elliptischen Kurven wie in [TR-03109-3] bzw. [TR-03116-3] vorgesehen.
- Die Domain Parameter einer Elliptischen Kurve umfassen: Primzahl, erster Koeffizient, zweiter Koeffizient, Basispunkt, Ordnung des Basispunktes, Cofaktor.
- Es erfolgt Sicherheitsmodul-intern eine Betriebssystem-spezifische Ablage der Domain Parameter (Hersteller-abhängige Codierung und Speicherung).
- Für eine *gesicherte* Ablage der Domain Parameter im Sicherheitsmodul ist von Seiten des Betriebssystems zu sorgen (insbesondere Integritätsschutz).

- In Krypto-Kommandos erfolgt eine Referenzierung der Elliptischen Kurven ausschließlich über ihre OID für sog. Standardized Domain Parameter, siehe [TR-03111], Kap. 6 und [RFC 5114]. Ausnahme bildet das MSE-Kommando (SET-Variante mit AT-Template) zum Setzen der Elliptischen Kurve für das Kommando GENERAL AUTHENTICATE / Variante PACE; hier wird im MSE-Kommando die Elliptische Kurve über ihre ID wie in [TR-03110-3], Abschnitt A.2.1.1 angegeben referenziert.

3.2.4.5 Object Identifier (OID)

Im Zusammenhang mit Krypto-Funktionalität werden im Sicherheitsmodul folgende Object Identifier (OIDs) verwendet:

Elliptische Kurven / Krypto-Algorithmen / Krypto-Protokolle	**OID**
Elliptische Kurven	
Domain Parameter für Elliptische Kurven: Parameter wie in [TR-03116-3], Kap. 2.2 vorgegeben.	Siehe [TR-03111], Kap. 6 für Brainpool-Kurven bzw. [RFC 5114] für NIST-Kurven.
Krypto-Algorithmen	
ECDSA ohne Hashing: ECDSA-Parameter wie für die Implementierung der Signaturerzeugung und -verifikation im GW benötigt (Inhaltsdatensignatur, TLS, sonstige Signaturen) und in [TR-03116-3], Kap. 2, 3, 4, 5 und 6 vorgegeben.	id-ecdsa-plain-signatures (0.4.0.127.0.7.1.1.4.1) Siehe [TR-03111], Kap. 5.2.1.1.
ECDSA mit Hashing: ECDSA- und Hash-Parameter wie in [TR-03116-3], Kap. 2 vorgegeben.	id-ecdsa-plain-SHA (0.4.0.127.0.7.1.1.4.1.7) id-ecdsa-plain-SHA256 (0.4.0.127.0.7.1.1.4.1.3) id-ecdsa-plain-SHA384 (0.4.0.127.0.7.1.1.4.1.4) id-ecdsa-plain-SHA512 (0.4.0.127.0.7.1.1.4.1.5) Siehe [TR-03111], Kap. 5.2.1.1. id-ecdsa-plain-SHA wird nur im Key-Attribut Key-CryptoAlg verwendet, um anzuzeigen, dass bei Verwendung des Key-Objektes (hier: im Kommando PSO VERIFY CERTIFICATE bzw. EXTERNAL AUTHENTICATE) Sicherheitsmodul-intern ein Hashing stattfindet. Der zu verwendende Hash-Algorithmus wird über ein dem betreffenden Krypto-Kommando vorhergehendes MSE SET-Kommando ausgewählt, wozu id-ecdsa-plain-SHA256, id-ecdsa-plain-SHA384 bzw. id-ecdsa-plain-SHA512 zur Anzeige von SHA-256, SHA-384 bzw. SHA-512 zur Verfügung steht.
ECKA-EG (ohne KDF): Parameter wie für die Implementierung der Inhaltsdatenverschlüsselung im GW benötigt und in [TR-03116-3], Kap. 2 und 8 vorgegeben.	id-ecka-eg (0.4.0.127.0.7.1.1.5.1) Siehe [TR-03111], Kap. 5.3.1.
ECKA-DH (ohne KDF):	id-ecka-dh

Elliptische Kurven / Krypto-Algorithmen / Krypto-Protokolle	**OID**
Parameter wie für die Implementierung von TLS im GW benötigt (TLS Handshake) und in [TR-03116-3], Kap. 2, 3, 4, 5 und 6 vorgegeben.	(0.4.0.127.0.7.1.1.5.2) Siehe [TR-03111], Kap. 5.3.1.
Krypto-Protokolle	
PACE: Protokoll-Parameter wie für die Implementierung von PACE im GW benötigt und in [TR-03116-3], Kap. 9 vorgegeben.	id-PACE-ECDH-GM-AES-CBC-CMAC-128 (0.4.0.127.0.7.2.2.4.2.2) id-PACE-ECDH-GM-AES-CBC-CMAC-192 (0.4.0.127.0.7.2.2.4.2.3) id-PACE-ECDH-GM-AES-CBC-CMAC-256 (0.4.0.127.0.7.2.2.4.2.4) Siehe [TR-03110-3] und [TR-03111], Kap. 5.4.1.
ECKA-EG (ohne KDF): • Protokoll-Variante [TR-03111], Kap. 4.3.2.2 mit SMGW/Sicherheitsmodul als Recipient (Protokoll-Variante 1.1 in Kap. 4.5.5 a)) • Protokoll-Variante [TR-03111], Kap. 4.3.2.2 mit SMGW/Sicherheitsmodul als Initiator (Protokoll-Variante 1.2 in Kap. 4.5.5 a)) Protokoll-Parameter wie für die Implementierung der Inhaltsdatenverschlüsselung im GW benötigt und in [TR-03116-3], Kap. 2 und 8 vorgegeben.	id-ECKA-EG-REC-woKDF (0.4.0.127.0.7.2.2.9.1.1) id-ECKA-EG-INI-woKDF (0.4.0.127.0.7.2.2.9.2.1)
ECKA-DH (ohne KDF): • Protokoll-Variante [TR-03111], Kap. 4.3.2.1 mit SMGW/Sicherheitsmodul als Initiator oder Recipient (TLS-Client, Protokoll-Variante 2.1 in Kap. 4.5.5 a)) • Protokoll-Variante [TR-03111], Kap. 4.3.2.1 mit SMGW/Sicherheitsmodul als Initiator oder Recipient (TLS-Server, Protokoll-Variante 2.2 in Kap. 4.5.5 a)) Protokoll-Parameter wie für die TLS-Implementierung im GW benötigt und in [TR-03116-3], Kap. 2, 4, 5 und 6 vorgegeben.	id-ECKA-DH-CLT-woKDF (0.4.0.127.0.7.2.2.10.1.1) id-ECKA-DH-SRV-woKDF (0.4.0.127.0.7.2.2.10.2.1)

Tabelle 7: Object Identifier (OID)

OIDs für Krypto-Algorithmen werden im Key-Attribut Key-CryptoAlg der Key-Objekte sowie in Kommandos zum Setzen relevanter Krypto-Informationen (z.B. zur Auswahl der zu verwendenden Hash-Funktion) für ein nachfolgendes Krypto-Kommando verwendet. OIDs für Krypto-Protokolle hingegen werden für die Auswahl der Protokoll-Variante eines Krypto-Kommandos (hier: Variante des Kommandos GENERAL AUTHENTICATE) eingesetzt.

3.2.5 Sicherheitsmodul als Speicher für Zertifikate

Zertifikate werden grundsätzlich im GW gespeichert, mit Ausnahme des SM-PKI-Root-Zertifikates sowie der Gütesiegel-Zertifikate, die im Rahmen der Vor-Personalisierung im Sicherheitsmodul hinterlegt werden.

Es ist ausreichend Speicher für die Speicherung mehrerer SM-PKI-Root-Zertifikate im Sicherheitsmodul vorzusehen, da im Normalbetrieb des SMGW ein Austausch des Root-Zertifikates möglich sein muss und aus Migrationsgründen zeitweise evtl. alte wie auch neue Zertifikate im Sicherheitsmodul vorhanden sein müssen.

Ebenso ist ausreichend viel Speicher für die Gütesiegel-Zertifikate (3) im Sicherheitsmodul vorzusehen. Ein Update der Gütesiegel-Zertifikate ist nicht vorgesehen.

Die Speicherung der Zertifikate erfolgt in entsprechenden Datenfeldern im DF.SMGW.

Die Zertifikate werden jeweils in transparenten EFs gespeichert. Für die SM-PKI-Root-Zertifikate sind die Datenfelder EF.SMPKIRoot_x (mit $1 \leq x \leq 10$), für die Gütesiegel-Zertifikate die Datenfelder EF.GSCert_TLS, EF.GSCert_SIG und EF.GSCert_ENC vorgesehen.

Das Schreiben der Zertifikate erfolgt über das Kommando UPDATE BINARY, das Auslesen über das Kommando READ BINARY. Benötigt wird ferner das Kommando SELECT zur vorhergehenden Auswahl des jeweiligen Zertifikats-EFs, oder aber das Zertikats-EF wird im UPDATE BINARY bzw. READ BINARY-Kommando via SFI referenziert. (Zu beachten: SFIs sind auf den Wertebereich von 1 – 30 (dez.) beschränkt.) Siehe Kap. 3.4.3.1, 3.4.4.1 und 3.4.4.2.

Neben dem SM-PKI-Root-Zertifikat selbst wird auch der Root-Public Key aus dem genannten Zertifikat im Sicherheitsmodul in Form eines Public Key-Objektes abgelegt. Grund hierfür ist, dass das Sicherheitsmodul in Krypto-Operationen, die den Root-Public Key benötigen, nicht direkt auf den Public Key im SM-PKI-Root-Zertifikat zugreifen kann. Im Rahmen der Vor-Personalisierung des Sicherheitsmoduls wird das initiale SM-PKI-Root-Zertifikat sowie der darin enthaltene initiale Root-Public Key abgelegt. Erfolgt in nachfolgenden Phasen des Lebenszyklus-Modells ein Update des SM-PKI-Root-Zertifikates, so ist sowohl das neue SM-PKI-Root-Zertifikat wie auch der darin enthaltene neue Root-Public Key im Sicherheitsmodul zu speichern. Hierbei ist jeweils auf die Übereinstimmung der Public Key-Daten im SM-PKI-Root-Zertifikat und im zugehörigen Public Key-Objekt zu achten.

Das SM-PKI-Root-Zertifikat wird im Rahmen der Prüfung von Zertifikaten bzw. Zertifikatsketten der SM-PKI als Sicherheitsanker verwendet. Genauer bedient sich das GW hierzu des Sicherheitsmoduls und nutzt für die erforderlichen Signaturprüfungen gegen die Root das Public Key-Objekt, das den im SM-PKI-Root-Zertifikat enthaltenen Root-Public Key speichert. (An dieser Stelle ist die Übereinstimmung der Public Key-Daten im SM-PKI-Root-Zertifikat und im zugehörigen Public Key-Objekt wesentlich.)

Hinweis: Das Sicherheitsmodul selbst führt keine Prüfung von X.509-Zertifikaten und -Zertifikatsketten durch. Die Prüfung erfolgt im wesentlichen durch das GW, das sich dazu aber der Kernroutine des Sicherheitsmoduls zur Signaturprüfung bedient.

3.2.6 Sicherheitsmodul als Speicher für symmetrische Schlüssel

Das GW soll gemäß [TR-03109-1] bzw. [PP 0073] eine Verschlüsselung seines Flash-Speichers besitzen und ferner einen Integritätsschutz für seine gespeicherten Daten (insbesondere für die gespeicherten Zertifikate) bieten. Die Krypto-Operationen hierzu finden komplett im GW statt, und das Sicherheitsmodul kann vom GW als Schlüsselspeicher für die hierzu erforderlichen

symmetrischen Schlüssel genutzt werden. Das Sicherheitsmodul bietet hierzu entsprechende Speicherbereiche für die Schlüssel sowie entsprechende Kommandos zum Zugriff auf diese Speicherbereiche an.

Die Speicherung der symmetrischen GW-Keys erfolgt in einem entsprechenden Datenfeld im DF.SMGW. Für die GW-Keys ist das Datenfeld EF.GWKeys vorgesehen.

Das Sicherheitsmodul stellt nur ausreichenden Speicherplatz für die symmetrischen GW-Keys zur Verfügung. Die konkrete Datenstruktur für die gespeicherten Inhalte (symmetrische Keys) wird von Seiten des GW bestimmt. Das GW übernimmt selbst das Management der gespeicherten Schlüssel sowie die Krypto-Operationen mit den im Sicherheitsmodul gespeicherten Schlüsseln.

Vorschlag:

AES-basierte Verfahren (da das GW an anderer Stelle bereits AES implementieren muss), z.B. AES-CBC und AES-CMAC; das GW benötigt hierzu 1 AES-Key für die Verschlüsselung und 1 AES-Key für die MAC-Sicherung.

Das Sicherheitsmodul stellt 2 Speicherbereiche für die beiden AES-Keys zur Verfügung, oder aber das Sicherheitsmodul stellt einen Speicherbereich für einen Masterkey zur Verfügung, aus dem das GW mit einer Schlüsselableitungsfunktion dann seine beiden AES-Keys ableitet (wobei diese Schlüsselableitungsfunktion allein Gegenstand des GW ist).

Im Sicherheitsmodul zu speichernde Objekte: 1-2 AES-Keys.

Speicher und Zugriff:

- Record-EF mit 2 Records ausreichender Größe für die beiden GW-Keys
- Lesezugriff über READ RECORD (siehe Kap. 3.4.4.3)
- Schreibzugriff über UPDATE RECORD (siehe Kap. 3.4.4.4)
- Auswahl des Record-EFs über ein vorhergehendes SELECT-Kommando (siehe Kap. 3.4.4.5) oder aber im READ RECORD bzw. UPDATE RECORD-Kommando via SFI-Referenz. (Zu beachten: SFIs sind auf den Wertebereich von 1 – 30 (dez.) beschränkt.)

Für die „Masterkey-Variante mit Schlüsselableitung im GW“ genügt ein Record-EF mit einem Record. Evtl. kann auch das Kommando APPEND RECORD angeboten werden, um weitere Records für weitere Schlüssel anzuhängen (siehe Kap. 3.4.4.5).

3.2.7 Sicherheitsmodul als Speicher und Nutzer von PINs

3.2.7.1 Generelles

Im Rahmen des PACE-Protokolls verwendet das GW eine System-PIN.

Die GW-System-PIN wird in einem sog. PIN-Objekt im DF.SMGW gespeichert. Das für die GW-System-PIN erforderliche PIN-Objekt wird im Initialisierungsfile für das Sicherheitsmodul bereits (als leeres Objekt) angelegt und mit spezifischen Werten für die PIN-Zusatzinformationen vorbelegt.

Die GW-System-PIN kann beim ersten Hochfahren des integrierten SMGW durch das GW automatisch generiert und im Sicherheitsmodul als Referenzwert hinterlegt werden (mittels Kommando CHANGE REFERENCE DATA). Alternativ kann die GW-System-PIN auch im

Rahmen der Vor-Personalisierung extern generiert, in das GW importiert und beim ersten Hochfahren des integrierten SMGW im Sicherheitsmodul als Referenzwert hinterlegt werden (wie zuvor über das Kommando CHANGE REFERENCE DATA).

3.2.7.2 PIN-Objekte

Für im Sicherheitsmodul gespeicherte und verwaltete PIN-Objekte gilt Folgendes:

- Die Referenzierung erfolgt über eine PIN-ID von 1 Byte Länge bzw. in Kommandos über die sog PIN Reference (siehe unten).
- Die PIN-Suche im Sicherheitsmodul erfolgt anhand der PIN Reference und ist Betriebssystem-spezifisch implementiert.
- Das Betriebssystem erkennt und lehnt die mehrfache Vergabe von PIN-IDs innerhalb desselben Ordners für die im Sicherheitsmodul gespeicherten PIN-Objekte ab.
- Gespeichert werden die PIN-Daten:
  - sog. PIN-Referenzwert
- Gespeichert werden mindestens folgende PIN-Zusatzinformationen (PIN-Attribute):
  - PIN-ID (zur Identifikation der PIN, relevant für die PIN-Suche)

    Für die eigentliche PIN-Nummer stehen nur die untersten 5 Bit in der PIN-ID zur Verfügung. Wird in einem Kommando ein PIN-Objekt referenziert, so wird im Sinne der [ISO 7816-4] nicht die PIN-ID, sondern die sog. PIN Reference übergeben. Die PIN Reference stimmt in ihren untersten 5 Bit mit den untersten 5 Bit der PIN-ID überein, während über das MSBit der PIN Reference die Steuerung der PIN-Suche (1 für lokale Suche, 0 für globale Suche) erfolgt.
  - PIN-LifeCycleStatus: PIN „initialisation“ / „operational state – activated“ (LCSI des PIN-Objektes)
  - PIN-Mindestlänge

Für die GW-System-PIN ist eine Mindestlänge von 10 Dezimalziffern vorgegeben. Siehe hierzu die Vorbelegung des PIN-Objektes für die GW-System-PIN im Initialisierungsfile in Kap. 3.1.2.

Aufgrund der vorgegebenen Mindestlänge der GW-System-PIN von 10 Dezimalziffern kann auf einen Fehlbedienungszähler für die PIN verzichtet werden. Das Sicherheitsmodul hat ausreichende Sicherheitsmechanismen zu implementieren derart, dass ein Brute Force-Angriff auf die GW-System-PIN bei der Verwendung der PIN im Rahmen der Ausführung des PACE-Protokolls bzw. bei der (indirekten) PIN-Prüfung bei einem Wechsel der PIN über das Kommando CHANGE REFERENCE DATA bzgl. der für das Sicherheitsmodul angenommenen Resistenz gegen hohes Angriffspotenzial nicht möglich ist.

Die Codierung der GW-System-PIN erfolgt als Character String (ASCII-Codierung) wie in [TR-03110-3], Kap. D.2.1.4 definiert.

3.2.7.3 PIN-LifeCycleStatus

Ein PIN-Objekt im Zustand „initialisation“ ist vor seiner Nutzung für Operationen (wie z.B. GENERAL AUTHENTICATE / Variante PACE) mit PIN-Daten zu füllen. Dies wird über das Kommando CHANGE REFERENCE DATA in der Variante Setzen einer PIN ermöglicht.

3.3 Zugriffsregeln im Sicherheitsmodul

3.3.1 Zugriffsregel-Mechanismus und Sicherheitszustände

Das Betriebssystem des Sicherheitsmoduls stellt einen Zugriffsregel-Mechanismus bereit, der den Zugriff auf im Sicherheitsmodul gespeicherte bzw. verarbeitete Daten und Objekte sowie auf die im Betriebssystem verfügbaren Kommandos kontrolliert. Der Zugriffsregel-Mechanismus berücksichtigt den Life Cycle-Status des Sicherheitsmoduls, den Life Cycle-Status (LCSI) der im Sicherheitsmodul verwalteten Ordner (DFs), Datenfelder (EFs), Key- und PIN-Objekte, das Security Environment (SE) sowie die vom Sicherheitsmodul verwalteten Sicherheitszustände, d.h. wertet diese Informationen jeweils aus und schaltet entsprechend der gesetzten Zugriffsregeln den Zugriff auf Daten, Objekte bzw. Kommandos des Sicherheitsmoduls frei.

Zugriffsregeln werden Betriebssystem-intern Hersteller-spezifisch codiert. Es erfolgen diesbzgl. keine weiteren technischen Vorgaben von Seiten der vorliegenden TR.

Das Sicherheitsmodul verwaltet folgende Sicherheitszustände:

- AUTH := Key-bezogener Sicherheitszustand, der über ein erfolgreiches EXTERNAL AUTHENTICATE gesetzt wird

  Genutzt wird dies im Rahmen der Authentisierung des GW-Administrators gegenüber dem Sicherheitsmodul.

- PACE := PIN-bezogener Sicherheitszustand, der über ein erfolgreiches GENERAL AUTHENTICATE / Variante PACE gesetzt wird

  Genutzt wird dies im Rahmen des PACE-Protokolls zwischen GW und Sicherheitsmodul.

  Hinweis: Die erfolgreiche Ausführung des PACE-Protokolls ist mit dem Aufbau eines sicheren Kanals zwischen GW und Sicherheitsmodul verbunden, der für nachfolgende Kommandos zur Verfügung steht (Secure Messaging). Siehe hierzu auch Kap. 3.5.

Ein Zurücksetzen der zuvor genannten Sicherheitszustände des Sicherheitsmoduls ist über einen Aufruf des Kommandos MANAGE CHANNEL möglich.

Für die Authentisierung des GW-Administrators gegenüber dem Sicherheitsmodul und damit zum Setzen des Sicherheitszustandes AUTH werden die Public Key-Objekte Key.GWA_AUT_1 und Key.GWA_AUT_2 (siehe Kap. 3.1.2, Tabelle 4) herangezogen und in den Zugriffsregeln für DFs, EFs, Key- und PIN-Objekte entsprechend hinterlegt.

Für die gegenseitige PACE-basierte Authentisierung zwischen GW und Sicherheitsmodul und damit zum Setzen des Sicherheitszustandes PACE wird das PIN-Objekt PIN.GW (siehe Kap. 3.1.2, Tabelle 5) herangezogen und in den Zugriffsregeln für DFs, EFs, Key- und PIN-Objekte entsprechend hinterlegt.

Das Sicherheitsmodul verwendet zur Differenzierung der Zugriffsregeln für die Ordner (DFs), Datenfelder (EFs), Key- und PIN-Objekte in den verschiedenen Phasen des Lebenszyklus-Modells

für Sicherheitsmodul und GW entsprechende Security Environments (SE). Folgende SEs sind den einzelnen Phasen des Lebenszyklus-Modells zugeordnet:

Phase des Lebenszyklus-Modells	SEID
Vor-Personalisierung + Integration	02
Installation + Vor-Ort-Inbetriebnahme	-
Personalisierung	01
Normalbetrieb	01

Tabelle 8: Security Environments (SE)

Für die Ausprägung der für das Sicherheitsmodul relevanten SEs mit den ihnen jeweils zugeordneten Zugriffsregeln siehe Kap. 3.3.3 mit seinen den einzelnen Phasen des Lebenszyklus-Modells des SMGW zugeordneten Unterkapiteln 3.3.3.1, 3.3.3.2 und 3.3.3.3.

Darüber hinaus hängt der Zugriff auf Ordner (DFs), Datenfelder (EFs), Key- und PIN-Objekte durch Kommandos des Sicherheitsmoduls von ihrem jeweiligen Life Cycle-Status (LCSI) ab. Siehe hierzu Kap. 3.3.2.

Zusammengenommen gelten die in den folgenden Kapiteln 3.3.2 und 3.3.3 genannten Zugriffsbedingungen für die im Sicherheitsmodul gespeicherten bzw. verarbeiteten Daten und Objekte sowie Kommandos in Abhängigkeit vom Life Cycle-Status des Sicherheitsmoduls, vom Life Cycle-Status (LCSI) der Ordner (DFs), Datenfelder (EFs), Key- und PIN-Objekte und vom jeweiligen SE.

Die Zugriffsregeln für das Sicherheitsmodul werden bereits im Rahmen der Produktion bzw. Initialisierung des Sicherheitsmoduls im Sicherheitsmodul hinterlegt. Die Zugriffsregeln sind Gegenstand der CC-Zertifizierung des Sicherheitsmoduls und nach Auslieferung des Sicherheitsmoduls als initialisiertes Modul nicht mehr veränderbar.

Im Sicherheitsmodul, d.h. im MF, im DF.SMGW und in ggf. weiteren Unterordnern im MF oder DF.SMGW, können im Rahmen der Produktion bzw. Initialisierung des Sicherheitsmoduls weitere Ordner (DFs), Datenfelder (EFs) sowie Key- und PIN-Objekte, die über die für das Initialisierungsfile in Kap. 3.1.2 vordefinierten Ordner, Datenfelder, Key- und PIN-Objekte hinausgehen, im Sicherheitsmodul angelegt werden. Hierzu kann das Initialisierungsfile aus Kap. 3.1.2 entsprechend um die zusätzlichen Ordner, Datenfelder, Key- und PIN-Objekte erweitert werden. Diese zusätzlichen Ordner, Datenfelder, Key- und PIN-Objekte und deren Inhalte liegen außerhalb der vorliegenden Spezifikation des Sicherheitsmoduls wie dieses für seinen Einsatz im SMGW bzw. Smart Meter-System vorgesehen ist. Es muss aber sichergestellt und im Rahmen der CC-Zertifizierung des Sicherheitsmoduls geprüft werden, dass diese zusätzlichen Ordner, Datenfelder, Key- und PIN-Objekte sowie die zugehörigen Management- und Krypto-Kommandos nicht die für das Smart Meter-Sicherheitsmodul und seinen Einsatz im SMGW bzw. im Smart Meter-System vorgesehenen Sicherheitsstrukturen verändern, umgehen oder außer Kraft setzen. Siehe hierzu auch Kap. 5.

Auch können in den nachfolgenden Phasen des Lebenszyklus-Modells unter Einhaltung der Zugriffsregeln wie in Kap. 3.3.3.1 und 3.3.3.3 angegeben weitere Ordner, Datenfelder, Key- und PIN-Objekte im Sicherheitsmodul angelegt werden.

3.3.2 Kommando-Verhalten in Abhängigkeit vom LCSI der Ordner, Datenfelder, Key- und PIN-Objekte

Betrachtet werden im folgenden die folgenden Objekttypen, die durch das Sicherheitsmodul und sein Betriebssystem bereitgestellt, verwaltet und verwendet werden:

MF / DF / EF / Key-Objekt (Key Pair-Objekt bzw. Public Key-Objekt) / PIN-Objekt.

Objekte der vorgenannten Objekttypen tragen jeweils einen LCSI. Für die vom Betriebssystem des Sicherheitsmoduls mindestens bereitzustellenden und zu verarbeitenden LCSI-Werte siehe Kap. 3.2.2, 3.2.4.2.1, 3.2.4.2.2 und 3.2.7.2.

Die folgenden Tabellen 9, 10, 11, 12, 13 und 14 geben für die verschiedenen Objekttypen jeweils an, ob bei einem Kommando-Zugriff auf ein Objekt des betreffenden Objekttyps in Abhängigkeit von seinem LCSI das betreffende Kommando prinzipiell ausgeführt werden kann (Notation: 'JA') oder das betreffende Kommando mit einer Fehlermeldung abbricht, da der gewünschte Objekt-Zugriff vom Betriebssystem des Sicherheitsmoduls prinzipiell abgelehnt wird (Notation: 'NEIN'). Die Angabe '---' in den Tabellen zeigt an, dass das betreffende Kommando für den in der jeweiligen Tabelle betrachteten Objekttyp nicht relevant ist.

Hierbei geht es nur um den *grundsätzlichen* Zugriff auf ein Objekt durch ein Kommando hinsichtlich der Auswertung des LCSI des Objektes. Etwaig weitere für ein Objekt bestehende Zugriffsbedingungen wie in Kap. 3.3.3.1 und 3.3.3.3 genannt werden in diesem Kapitel im folgenden nicht betrachtet.

Für einen Zugriff auf ein konkretes Objekt ergibt sich im Endeffekt eine UND-Verknüpfung, bestehend zum einen aus der betreffenden LCSI-abhängigen grundsätzlichen Zugriffsbedingung (wie im folgenden für die verschiedenen Objekttypen ausgeführt) und zum anderen aus der weiteren betreffenden Phasen-, Objekttyp- und Objekt-abhängigen spezifischen Zugriffsbedingung wie in Kap. 3.3.3.1 und 3.3.3.3, Tabellen 15, 16 und 17 angegeben.

Für die in den Tabellen 9, 10, 11, 12, 13 und 14 aufgeführten Kommandos siehe die Übersichtstabelle 18 in Kap. 3.4.1 sowie die Kommandobeschreibungen in Kap. 3.4 und 4.

a) MF

MF				
Kommando	**LCSI**			
	initialisation[1]	**operational state - activated**	**operational state - deactivated**[2]	**terminated**[3]
SELECT[4]	JA	JA	JA	JA
CREATE FILE	NEIN	JA	NEIN	NEIN
DELETE FILE	NEIN	NEIN	JA	JA
ACTIVATE FILE	NEIN	JA[5]	JA	NEIN
DEACTIVATE FILE	NEIN	NEIN	JA	NEIN
TERMINATE DF	NEIN	NEIN	JA	JA

MF				
Kommando	**LCSI**			
	initialisation[1]	**operational state - activated**	**operational state - deactivated**[2]	**terminated**[3]
CREATE KEY	NEIN	JA	NEIN	NEIN

Tabelle 9: Zugriff auf MF

Anmerkungen zu den nummerierten Tabelleneinträgen in Tabelle 9:

1 Im vorliegenden Fall für das MF nicht relevant, da der LCSI des MF im Initialisierungsfile mit „operational state – activated“ vorbelegt ist und ein Rückschritt auf LCSI „initialisation“ nicht möglich ist.

2 Im vorliegenden Fall für das MF nicht relevant, da der LCSI des MF im Initialisierungsfile mit „operational state – activated“ vorbelegt ist und für das MF mit diesem LCSI-Wert das Kommando DEACTIVATE FILE nicht möglich ist. Die Belegung in dieser Spalte erfolgt nur aus Gründen der Konsistenz zu den Festlegungen für DFs (siehe Tabelle 10).

3 Im vorliegenden Fall für das MF nicht relevant, da der LCSI des MF im Initialisierungsfile mit „operational state – activated“ vorbelegt ist und für das MF mit diesem LCSI-Wert das Kommando TERMINATE DF nicht möglich ist. Die Belegung in dieser Spalte erfolgt nur aus Gründen der Konsistenz zu den Festlegungen für DFs (siehe Tabelle 10).

4 Das Kommando ist (üblicherweise) nicht mit einer Auswertung von Zugriffsregeln verbunden.

5 Erfolgreiche Kommando-Ausführung ändert am LCSI des MF nichts.

b) DFs

DF				
Kommando	**LCSI**			
	initialisation	**operational state - activated**	**operational state - deactivated**	**terminated**
SELECT[1]	JA	JA	JA	JA
CREATE FILE	JA	JA	NEIN	NEIN
DELETE FILE	JA	JA	JA	JA
ACTIVATE FILE	JA	JA[2]	JA	NEIN
DEACTIVATE FILE	JA	JA	JA[3]	NEIN
TERMINATE DF	JA	JA	JA	JA[4]
CREATE KEY	JA	JA	NEIN	NEIN

Tabelle 10: Zugriff auf DFs

Anmerkungen zu den nummerierten Tabelleneinträgen in Tabelle 10:

1 Das Kommando ist (üblicherweise) nicht mit einer Auswertung von Zugriffsregeln verbunden.

2 Erfolgreiche Kommando-Ausführung ändert am LCSI des DF nichts.

3 Erfolgreiche Kommando-Ausführung ändert am LCSI des DF nichts.

4 Erfolgreiche Kommando-Ausführung ändert am LCSI des DF nichts.

c) EFs

EF				
Kommando	**LCSI**			
	initialisation	**operational state - activated**	**operational state - deactivated**	**terminated**
SELECT[1]	JA	JA	JA	JA
DELETE FILE	JA	JA	JA	JA
ACTIVATE FILE	JA	JA[2]	JA	NEIN
DEACTIVATE FILE	JA	JA	JA[3]	NEIN
TERMINATE EF	JA	JA	JA	JA[4]
READ BINARY	JA[5]	JA	NEIN	NEIN
UPDATE BINARY	JA	JA	NEIN	NEIN
READ RECORD	JA[6]	JA	NEIN	NEIN
UPDATE RECORD	JA	JA	NEIN	NEIN
APPEND RECORD (sofern implementiert)	JA	JA	NEIN	NEIN

Tabelle 11: Zugriff auf EFs

Anmerkungen zu den nummerierten Tabelleneinträgen in Tabelle 11:

1 Das Kommando ist (üblicherweise) nicht mit einer Auswertung von Zugriffsregeln verbunden.

2 Erfolgreiche Kommando-Ausführung ändert am LCSI des EF nichts.

3 Erfolgreiche Kommando-Ausführung ändert am LCSI des EF nichts.

4 Erfolgreiche Kommando-Ausführung ändert am LCSI des EF nichts.

5 Evtl. EF noch leer (Nutzdaten noch nicht vorhanden).

6 Evtl. EF noch leer (Nutzdaten noch nicht vorhanden).

d) Key-Objekte

d1) Key Pair-Objekte

Key Pair-Objekt (persistent)			
Kommando	**LCSI (Key-LifeCycleStatus)**		
	initialisation	**operational state - activated**	**operational state - deactivated**
DELETE KEY	JA	JA	JA
ACTIVATE KEY	NEIN[1]	JA[2]	JA
DEACTIVATE KEY	NEIN	JA	JA[3]
MSE SET[4]	JA	JA	JA
GENERATE ASYMMETRIC KEY PAIR / KeyGen	JA	NEIN	JA
GENERATE ASYMMETRIC KEY PAIR / Export Public Key	NEIN[5]	JA	JA[6]
PSO COMPUTE DIGITAL SIGNATURE	NEIN	JA	NEIN
GENERAL AUTHENTICATE / ECKA-EG[7]	NEIN	JA	NEIN
GENERAL AUTHENTICATE / ECKA-DH[8]	---	---	---
INTERNAL AUTHENTICATE	NEIN	JA	NEIN

Tabelle 12: Zugriff auf Key Pair-Objekte

Anmerkungen zu den nummerierten Tabelleneinträgen in Tabelle 12:

1 Key Pair-Objekt noch leer (Schlüsseldaten noch nicht vorhanden).

2 Erfolgreiche Kommando-Ausführung ändert am LCSI des Key Pair-Objektes nichts.

3 Erfolgreiche Kommando-Ausführung ändert am LCSI des Key Pair-Objektes nichts.

4 Das Kommando ist nicht mit einer Auswertung von Zugriffsregeln verbunden.

5 Da das Key Pair-Objekt noch leer ist (Schlüsseldaten noch nicht vorhanden), macht diese Kommando-Variante im LCSI „initialisation“ keinen Sinn.

6 Das Key Pair-Objekt ist noch mit alten Schlüsseldaten gefüllt, und es erfolgt eine Ausgabe dieser alten Schlüsseldaten.

7 Relevant ist hier nur GENERAL AUTHENTICATE in der Protokoll-Variante 1.1.

8 Nicht relevant, da nur ephemerale, also temporäre Schlüssel beteiligt sind.

Für temporäre Key Pair-Objekte wird im Sicherheitsmodul nur der Key-LifeCycleStatus „operational state – activated“ verwendet, siehe Kap. 3.2.4.2.1. Ein Zugriff auf temporäre Key Pair-Objekte findet über die Kommandos GENERATE ASYMMETRIC KEY PAIR in der Variante Schlüsselgenerierung mit Ausgabe des Public Key, GENERAL AUTHENTICATE / ECKA-EG (nur Protokoll-Variante 1.2) und GENERAL AUTHENTICATE / ECKA-DH (beide Protokoll-Varianten

2.1 und 2.2) statt und ist im Key-LifeCycleStatus „operational state – activated“ möglich. Weitere Kommandos, insbesondere Key Management-Kommandos wie z.B. DELETE KEY, DEACTIVATE KEY, usw. werden für temporäre Key Pair-Objekte nicht benötigt.

d2) Public Key-Objekte

Public Key-Objekt (persistent)				
Kommando		**LCSI (Key-LifeCycleStatus)**		
		initialisation	**operational state - activated**	**operational state - deactivated**
DELETE KEY		JA	JA	JA
ACTIVATE KEY		NEIN[1]	JA[2]	JA
DEACTIVATE KEY		NEIN	JA	JA[3]
MSE SET[4]		JA	JA	JA
PSO VERIFY CERTIFICATE				
	zu befüllendes Public Key-Objekt	JA	NEIN	JA
	Signaturprüfschlüssel	NEIN	JA	NEIN
PSO VERIFY DIGITAL SIGNATURE		NEIN	JA	NEIN
GENERAL AUTHENTICATE / ECKA-EG[5]		NEIN	JA	NEIN
GENERAL AUTHENTICATE / ECKA-DH[6]		---	---	---
EXTERNAL AUTHENTICATE		NEIN	JA	NEIN

Tabelle 13: Zugriff auf Public Key-Objekte

Anmerkungen zu den nummerierten Tabelleneinträgen in Tabelle 13:

1 Public Key-Objekt noch leer (Schlüsseldaten noch nicht vorhanden).

2 Erfolgreiche Kommando-Ausführung ändert am LCSI des Public Key-Objektes nichts.

3 Erfolgreiche Kommando-Ausführung ändert am LCSI des Public Key-Objektes nichts.

4 Das Kommando ist nicht mit einer Auswertung von Zugriffsregeln verbunden.

5 Relevant ist hier nur GENERAL AUTHENTICATE in der Protokoll-Variante 1.2.

6 Nicht relevant, da nur ephemerale, also temporäre Schlüssel beteiligt sind.

Temporäre Public Key-Objekte sind nur bei der Übergabe von Public Keys in einzelnen Krypto-Kommandos relevant und tragen keinen Key-LifeCycleStatus, siehe Kap. 3.2.4.2.2. Daher werden temporäre Public Key-Objekte hier nicht weiter betrachtet.

e) PIN-Objekte

PIN-Objekt		
Kommando	**LCSI (PIN-LifeCycleStatus)**	
	initialisation	**operational state - activated**
CHANGE REFERENCE DATA / Variante Setzen einer PIN	JA	NEIN
CHANGE REFERENCE DATA / Variante Wechseln einer PIN	NEIN	JA
GENERAL AUTHENTICATE / PACE	NEIN	JA

Tabelle 14: Zugriff auf PIN-Objekte

3.3.3 Phasen- bzw. SE-abhängige spezifische Zugriffsbedingungen

In den folgenden Unterkapiteln werden die Phasen- bzw. SE-abhängigen spezifischen Zugriffsbedingungen für die vom Betriebssystem des Sicherheitsmoduls verwalteten und verwendeten Objekttypen und Objekte betrachtet.

3.3.3.1 Vor-Personalisierung + Integration von Sicherheitsmodul und GW

Im Rahmen der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ werden bestimmte Datenfelder und Key- und PIN-Objekte, die mit der Initialisierung des Sicherheitsmoduls über das Initialisierungsfile mit seinen spezifischen Ordnern, Datenfeldern, Key- und PIN-Objekten (siehe Kap. 3.1.2) auf das Sicherheitsmodul aufgebracht wurden, mit Inhalt gefüllt.

Insbesondere wird in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ initial die GW-System-PIN generiert und im Sicherheitsmodul als Referenzwert hinterlegt (Kommando CHANGE REFERENCE DATA / Variante Setzen einer PIN).

Ferner werden in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ initial Public Keys importiert. Der Import von Public Keys wird mittels Import-Keys, die nur in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ zur Verfügung stehen sollen, realisiert. Die Import-Keys werden am Ende dieser Phase gelöscht. Siehe Kap. 3.1.2.

Relevant für die Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ ist das SE mit der SEID = 02.

Für die Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“, d.h. im SE mit SEID = 02, stehen prinzipiell alle in der vorliegenden TR in Kap. 3.4 und 4 für das Sicherheitsmodul spezifizierten Kommandos zur Verfügung. Jedoch bestehen über spezielle, dem SE mit SEID = 02 zugeordnete Zugriffsregeln Beschränkungen bzgl. der Ausführbarkeit der Kommandos des Sicherheitsmoduls. Diese Beschränkungen ergeben sich spezifisch für Ordner, Datenfelder sowie Key- und PIN-Objekte wie in folgender Tabelle 15 dargestellt.

Zur Notation in Tabelle 15:

Die Notation '- / PACE' in Tabelle 15 ist wie folgt zu verstehen: Die Unterscheidung der Zugriffsregeln hängt mit dem Lebenszyklus-Modell des SMGW zusammen und ergibt sich aus den möglichen Reihenfolgen der Teilphasen „Integration“ mit ihren Integrationsschritten und „Vor-Personalisierung 1“. Findet der Integrationsschritt zum Setzen der GW-System-PIN im Rahmen der Teilphase „Integration“ vor der Teilphase „Vor-Personalisierung 1“ statt, so ist die GW-System-PIN gesetzt und der PACE-Kanal zwischen GW und Sicherheitsmodul kann prinzipiell genutzt werden. Ist das PIN-Objekt für die GW-System-PIN nicht aktiviert, wird ohne PACE-Kanal zwischen GW und Sicherheitsmodul gearbeitet. Nach Aktivierung des PIN-Objektes für die GW-System-PIN kann mit dem PACE-Kanal zwischen GW und Sicherheitsmodul gearbeitet werden. Das Sicherheitsmodul muss jedoch PACE nicht erzwingen, wenn das PIN-Objekt für die GW-System-PIN aktiviert ist; soll der PACE-Kanal genutzt werden, so zeigt die das betreffende Kommando aufrufende Stelle dies über das entsprechende CLA-Byte im Kommando an. (Hinweis: In diesem Fall wird jedoch die Verwendung von PACE für die Teilphasen „Vor-Personalisierung 1“ und „Vor-Personalisierung 2“ empfohlen.)

Ferner ist die Notation '- / PACE' in Tabelle 15 dahingehend zu verstehen, dass für den Fall des gesetzten Sicherheitszustandes PACE das betreffende Kommando in dem mittels des PACE-Protokolls aufgebauten sicheren Kanal zwischen GW und Sicherheitsmodul, also mit Secure Messaging gemäß Kap. 3.5 ausgeführt wird.

MF/DFs/EFs/Key-Objekte/PIN-Objekte	**Zugriffsregel**
MF	• SELECT: ALWAYS • DELETE FILE: NEVER (MF nicht löschbar) • DEACTIVATE FILE: NEVER (MF nicht deaktivierbar) • TERMINATE DF: NEVER (MF nicht terminierbar) • CREATE FILE: NEVER • CREATE KEY: NEVER
EF.SecModTRInfo (Technisches Datenfeld mit Informationen zu der für das vorliegende Sicherheitsmodul relevanten Spezifikation)	• SELECT EF: ALWAYS • READ RECORD: ALWAYS • UPDATE RECORD: NEVER, ausgenommen das Sicherheitsmodul bietet die Funktionalität des Sicherheitsmodul-Firmware-Updates (in diesem Fall gelten die Zugriffsregeln für ein Update des Datenfeldes wie im Rahmen des Firmware-Updates vorgesehen) • APPEND RECORD: - / PACE (sofern das Kommando im Sicherheitsmodul implementiert ist) • DELETE FILE: NEVER • ACTIVATE FILE: NEVER • DEACTIVATE FILE: NEVER • TERMINATE EF: NEVER
EF.SecModAccess (Technisches Datenfeld mit Informationen zu der vom vorliegenden Sicherheitsmodul angebotenen PACE-Funktionalität)	• SELECT EF: ALWAYS • READ BINARY: ALWAYS • UPDATE BINARY: NEVER, ausgenommen das Sicherheitsmodul bietet die Funktionalität des Sicherheitsmodul-Firmware-Updates (in diesem Fall gelten die Zugriffsregeln für ein Update des Datenfeldes wie im Rahmen

MF/DFs/EFs/Key-Objekte/PIN-Objekte	Zugriffsregel
	des Firmware-Updates vorgesehen) • DELETE FILE: NEVER • ACTIVATE FILE: NEVER • DEACTIVATE FILE: NEVER • TERMINATE EF: NEVER
EF.SecModCrypto (Technisches Datenfeld mit Informationen zu der vom vorliegenden Sicherheitsmodul angebotenen Krypto-Funktionalität)	• SELECT EF: ALWAYS • READ BINARY: ALWAYS • UPDATE BINARY: NEVER, ausgenommen das Sicherheitsmodul bietet die Funktionalität des Sicherheitsmodul-Firmware-Updates (in diesem Fall gelten die Zugriffsregeln für ein Update des Datenfeldes wie im Rahmen des Firmware-Updates vorgesehen) • DELETE FILE: NEVER • ACTIVATE FILE: NEVER • DEACTIVATE FILE: NEVER • TERMINATE EF: NEVER
EF.SecModLifeCycle (Technisches Datenfeld mit Informationen zum Life Cycle-Status des vorliegenden Sicherheitsmoduls)	• SELECT EF: ALWAYS • READ RECORD: - / PACE • UPDATE RECORD: - / PACE • APPEND RECORD: - / PACE (sofern das Kommando im Sicherheitsmodul implementiert ist) • DELETE FILE: - / PACE • ACTIVATE FILE: - / PACE • DEACTIVATE FILE: - / PACE • TERMINATE EF: - / PACE
DF.SMGW	• SELECT DF: ALWAYS • CREATE FILE: NEVER • DELETE FILE: NEVER • ACTIVATE FILE: - / PACE • DEACTIVATE FILE: NEVER • TERMINATE DF: NEVER • CREATE KEY: NEVER
EF.SMPKIRoot_x (Datenfelder für SM-PKI-Root-Zertifikate)	• SELECT EF: ALWAYS • READ BINARY: - / PACE • UPDATE BINARY: - / PACE • DELETE FILE: NEVER • ACTIVATE FILE: - / PACE • DEACTIVATE FILE: - / PACE • TERMINATE EF: - / PACE
EF.GSCert_TLS, EF.GSCert_SIG, EF.GSCert_ENC (Datenfelder für Gütesiegel-Zertifikate)	• SELECT EF: ALWAYS • READ BINARY: - / PACE • UPDATE BINARY: - / PACE, sofern der LCSI des EFs auf

MF/DFs/EFs/Key-Objekte/PIN-Objekte	Zugriffsregel
	„initialisation“ steht, andernfalls NEVER • DELETE FILE: NEVER • ACTIVATE FILE: - / PACE • DEACTIVATE FILE: NEVER • TERMINATE EF: NEVER Das Schreiben der Gütesiegel-Zertifikate über das Kommando UPDATE BINARY soll nur in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ möglich sein. Hierzu wird nach dem Schreiben der Gütesiegel-Zertifikate in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ der LCSI der betreffenden EFs über das Kommando ACTIVATE FILE von „initialisation“ auf „operational state – activated“ umgesetzt. Die Ausführung des Kommandos UPDATE BINARY ist für diese EFs nur beim LCSI „initialisation“ möglich.
EF.GWKeys (Datenfeld für symmetrische GW-Keys)	• SELECT EF: ALWAYS • READ RECORD: - / PACE • UPDATE RECORD: - / PACE • APPEND RECORD: - / PACE (sofern das Kommando im Sicherheitsmodul implementiert ist) • DELETE FILE: - / PACE • ACTIVATE FILE: - / PACE • DEACTIVATE FILE: - / PACE • TERMINATE EF: - / PACE
PIN-Objekte (PIN.GW zur Speicherung der GW-System-PIN)	PIN.GW: • CHANGE REFERENCE DATA / Variante Setzen einer PIN: ALWAYS • CHANGE REFERENCE DATA / Variante Wechseln einer PIN: - / PACE • GENERAL AUTHENTICATE / PACE: ALWAYS Hinweis: Das Kommando CHANGE REFERENCE DATA in der Variante „Setzen einer PIN“ ist für das PIN-Objekt PIN.GW für die GW-System-PIN in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ genau einmal ausführbar. Dies wird über das Kommando-Verhalten des Kommandos CHANGE REFERENCE DATA in der Variante „Setzen einer PIN“ sowie über das PIN-Attribut PIN-LifeCycleStatus des PIN-Objektes erreicht, das im Initialisierungsfile mit dem Wert „initialisation“ vorbelegt ist.
Key Pair-Objekte	Speziell gelten für die für das Initialisierungsfile in Kap. 3.1.2 vordefinierten Key Pair-Objekte folgende Zugriffsregeln: Key.WAN_TLS_PRE, Key.WAN_SIG_PRE: • GENERATE ASYMMETRIC KEY PAIR (alle 3 Varianten): - / PACE • PSO COMPUTE DIGITAL SIGNATURE: - / PACE

MF/DFs/EFs/Key-Objekte/PIN-Objekte	Zugriffsregel
	Key.WAN_ENC_PRE: • GENERATE ASYMMETRIC KEY PAIR (alle 3 Varianten): - / PACE • INTERNAL AUTHENTICATE : - / PACE Key.WAN_TLS, Key.WAN_SIG, Key.WAN_ENC: Diese Key Pair-Objekte tragen das Key-Attribut Key-SEID mit Wert 01 und sind in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ nicht weiter von Interesse. Key.IMP_TRANS: • PSO COMPUTE DIGITAL SIGNATURE: - / PACE • DELETE KEY: - / PACE Key.IMP: • GENERATE ASYMMETRIC KEY PAIR (alle 3 Varianten): - / PACE • PSO COMPUTE DIGITAL SIGNATURE: - / PACE • DELETE KEY: - / PACE • DEACTIVATE KEY: - / PACE Key.EPH_x: Diese Key Pair-Objekte tragen das Key-Attribut Key-SEID mit Wert 01 und sind in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ nicht weiter von Interesse. Für alle zuvor genannten Key Pair-Objekte sind jeweils alle weiteren Kommandos zum Key Management sowie alle weiteren Krypto-Kommandos nicht zulässig (Zugriffsregel NEVER). Die Generierung der vorläufigen GW-Schlüsselpaare (Key.WAN_TLS_PRE, Key.WAN_SIG_PRE, Key.WAN_ENC_PRE) über das Kommando GENERATE ASYMMETRIC KEY PAIR ist in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ genau einmal ausführbar. Dies wird über das Kommando-Verhalten des Kommandos GENERATE ASYMMETRIC KEY PAIR, über das Key-Attribut Key-LifeCycleStatus der Key-Objekte, das im Initialisierungsfile jeweils mit dem Wert „initialisation“ vorbelegt ist, sowie die Nicht-Zulässigkeit des Kommandos DEACTIVATE KEY erreicht. Neue Key Pair-Objekte können in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ im Sicherheitsmodul nicht angelegt werden.
Public Key-Objekte	Speziell gelten für die für das Initialisierungsfile in Kap. 3.1.2 vordefinierten Public Key-Objekte folgende Zugriffsregeln: Key.SMPKIRoot_x: • PSO VERIFY DIGITAL SIGNATURE: - / PACE

MF/DFs/EFs/Key-Objekte/PIN-Objekte	Zugriffsregel
	Key.GWA_TLS_x, Key.GWA_SIG_x, Key.GWA_ENC_x, Key.GWA_AUT_x: • DEACTIVATE KEY: - / PACE Die Administrationsschlüssel des GW-Administrators werden in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ in das Sicherheitsmodul importiert, aber nicht in Krypto-Kommandos verwendet. Key.CA_x: • PSO VERIFY DIGITAL SIGNATURE: - / PACE Key.IMP_PUB_TRANS: • PSO VERIFY CERTIFICATE: - / PACE • DELETE KEY: - / PACE Key.IMP_PUB: • PSO VERIFY CERTIFICATE: - / PACE • DELETE KEY: - / PACE • DEACTIVATE KEY: - / PACE Für alle zuvor genannten Public Key-Objekte sind jeweils alle weiteren Kommandos zum Key Management sowie alle weiteren Krypto-Kommandos nicht zulässig (Zugriffsregel NEVER). Neue Public Key-Objekte können in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ im Sicherheitsmodul nicht angelegt werden.

Tabelle 15: Zugriffsregeln für MF/DFs/EFs/Key-Objekte/PIN-Objekte in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ (SEID = 02)

Zulässig sind in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ Hersteller-spezifische zusätzliche technische Sicherungsmechanismen auf Seiten des Sicherheitsmoduls, z.B. zur Freischaltung des Sicherheitsmoduls für seine Vor-Personalisierung und Integration (Umschalten auf das Security Environment mit SEID = 02), für das Anstoßen der Schlüsselgenerierung oder für die Absicherung des Datentransports der Vor-Personalisierungsdaten zum Sicherheitsmodul.

3.3.3.2 Installation + Vor-Ort-Inbetriebnahme des SMGW

In der Phase „Installation + Vor-Ort-Inbetriebnahme des SMGW“ erfolgen keine Administrations-tätigkeiten am Sicherheitsmodul, so dass diese Phase hier nicht weiter betrachtet wird.

3.3.3.3 Personalisierung und Normalbetrieb des SMGW

In den Phasen „Personalisierung des SMGW“ und „Normalbetrieb des SMGW“ setzt eine Verwendung des Sicherheitsmoduls in der Regel (d.h. bis auf wenige Ausnahmen) eine erfolgreiche PACE-Authentisierung zwischen GW und Sicherheitsmodul voraus, d.h. nur ein erfolgreich

zwischen GW und Sicherheitsmodul ausgeführtes PACE-Protokoll schaltet das Sicherheitsmodul zu seiner weiteren Nutzung in den Phasen „Personalisierung des SMGW“ und „Normalbetrieb des SMGW“ frei.

Administrationstätigkeiten des GW-Administrators am Sicherheitsmodul setzen einen TLS-Kanal zwischen GW-Administrator und GW sowie zusätzlich eine Authentisierung des GW-Administrators gegenüber dem Sicherheitsmodul voraus.

Relevant für die Phasen „Personalisierung des SMGW“ und „Normalbetrieb des SMGW“ ist das SE mit der SEID = 01.

Für die Phasen „Personalisierung des SMGW“ und „Normalbetrieb des SMGW“ stehen (im SE mit SEID = 01) die in der folgenden Tabelle 16 gelisteten Kommandos des Sicherheitsmoduls wie in der vorliegenden TR in Kap. 3.4 und 4 spezifiziert zur Verfügung.

Zur Notation in Tabelle 16:

In der folgenden Tabelle 16 bedeutet die Angabe 'X', dass für das jeweilige Kommando der betreffende Sicherheitszustand im Sicherheitsmodul gesetzt sein muss. Findet sich kein Eintrag wie 'X' oder anderes, so bestehen keine Nutzungsbeschränkungen für das jeweilige Kommando. Die Angabe '…' kennzeichnet, dass je nach Situation ein 'X' zu setzen ist oder nicht und die Zugriffsbedingung durch das Sicherheitsmodul entsprechend durchgesetzt wird, siehe dazu jeweils die zugehörigen Angaben in der Spalte „Bemerkung“. Mit der Angabe (X) wird gekennzeichnet, dass die aufrufende Stelle bei Aufruf des Kommandos entscheidet, ob der betreffende Sicherheitszustand ausgenutzt werden soll oder nicht; die Anzeige der aufrufenden Stelle erfolgt bzgl. des Sicherheitszustandes PACE über das CLA-Byte des betreffenden Kommandos. Die Angabe '---' schließlich bedeutet 'keine Zugriffsbedingung, da nicht relevant'.

Ferner ist die Notation 'X' in Tabelle 16 in der Spalte 'PACE' dahingehend zu verstehen, dass das betreffende Kommando in dem mittels des PACE-Protokolls aufgebauten sicheren Kanal zwischen GW und Sicherheitsmodul, also mit Secure Messaging gemäß Kap. 3.5 ausgeführt wird. Für die Bedeutung der Notation '(X)' in der Spalte 'PACE' siehe die zugehörigen Angaben in der Spalte 'Bemerkung'.

Kommando	**PACE**	**AUTH**	**Bemerkung**
Key Management			
CREATE KEY	X	X	
DELETE KEY	X	X	Weitere Zugriffsbeschränkungen je nach Key-Objekt, siehe Tabelle 17.
ACTIVATE KEY	X	X	Weitere Zugriffsbeschränkungen je nach Key-Objekt, siehe Tabelle 17.
DEACTIVATE KEY	X	X	Weitere Zugriffsbeschränkungen je nach Key-Objekt, siehe Tabelle 17.
Krypto-Kommandos (asymmetrische Kryptographie)			
GENERATE ASYMMETRIC KEY PAIR / KeyGen (Schlüsselgenerierung mit/ohne Export des Public Key)	X	X	Weitere Zugriffsbeschränkungen je nach Key-Objekt, siehe Tabelle 17.

Kommando	PACE	AUTH	Bemerkung
GENERATE ASYMMETRIC KEY PAIR / Export Public Key (nur Schlüsselexport, ohne Schlüsselgenerierung)	X		
PSO COMPUTE DIGITAL SIGNATURE	X		
PSO VERIFY DIGITAL SIGNATURE / Variante ohne Übergabe des Public Key im Kommando	X		
PSO VERIFY DIGITAL SIGNATURE / Variante mit Übergabe des Public Key im Kommando	(X)		Das Kommando kann auf Seiten des Sicherheitsmoduls prinzipiell mit und ohne Secure Messaging ausgeführt werden und verlangt damit nicht notwendig einen gesetzten Sicherheitszustand PACE. Die das Kommando aufrufende Stelle zeigt im CLA-Byte entsprechend an, ob das Kommando mit oder ohne Secure Messaging ausgeführt werden soll.
PSO VERIFY CERTIFICATE	X	X	
GENERAL AUTHENTICATE / ECKA (-EG, -DH)	X		
GENERAL AUTHENTICATE / PACE			
EXTERNAL AUTHENTICATE	X		
INTERNAL AUTHENTICATE	X		
Management des Security Environments (SE)			
MSE SET			
MSE RESTORE			
Zufallszahlengenerierung			
GET CHALLENGE	(X)		Das Kommando kann auf Seiten des Sicherheitsmoduls prinzipiell mit und ohne Secure Messaging ausgeführt werden und verlangt damit nicht notwendig einen gesetzten Sicherheitszustand PACE. Die das Kommando aufrufende Stelle zeigt im CLA-Byte entsprechend an, ob das Kommando mit oder ohne Secure Messaging ausgeführt werden soll.
PIN-Management			
CHANGE REFERENCE DATA / Variante Wechseln einer PIN	X	...	Weitere Zugriffsbeschränkungen je nach PIN; für die GW-System-PIN siehe Tabelle 17.
CHANGE REFERENCE DATA / Variante Setzen einer PIN	---	---	Diese Kommando-Variante ist in den Phasen „Personalisierung des SMGW“ und „Normalbetrieb des

Kommando	PACE	AUTH	Bemerkung
			SMGW“ nicht relevant. Siehe hierzu auch die diesbzgl. Angaben in Tabelle 17 zum PIN-Objekt für die GW-System-PIN.
Zugriff auf transparente EF			
READ BINARY	...	...	Zugriffsbeschränkungen je nach EF, siehe Tabelle 17.
UPDATE BINARY	...	...	Zugriffsbeschränkungen je nach EF, siehe Tabelle 17.
Zugriff auf Record-orientierte EF			
READ RECORD	...	...	Zugriffsbeschränkungen je nach EF, siehe Tabelle 17.
UPDATE RECORD	...	...	Zugriffsbeschränkungen je nach EF, siehe Tabelle 17.
APPEND RECORD (optional)	...	...	Zugriffsbeschränkungen je nach EF, siehe Tabelle 17.
Kartenmanagement / Management des Filesystems			
SELECT (MF/DF/EF)			
CREATE FILE	X	X	Weitere Zugriffsbeschränkungen je nach File möglich.
DELETE FILE	X	X	Weitere Zugriffsbeschränkungen je nach File möglich.
ACTIVATE FILE	X	X	Weitere Zugriffsbeschränkungen je nach File möglich.
DEACTIVATE FILE	X	X	Weitere Zugriffsbeschränkungen je nach File möglich.
TERMINATE DF	X	X	Weitere Zugriffsbeschränkungen je nach File möglich.
TERMINATE EF	X	X	Weitere Zugriffsbeschränkungen je nach File möglich.
Management des Life Cycle-Status			
TERMINATE CARD USAGE	X	X	
Management der Applikationsebene			
MANAGE CHANNEL			

Tabelle 16: Zugriffsregeln für Kommandos in den Phasen „Personalisierung des SMGW“ und „Normalbetrieb des SMGW“ (SEID = 01)

Ferner gelten für Ordner, Datenfelder sowie Key- und PIN-Objekte (für die Bezeichner siehe auch Kap. 3.1.2) folgende Festlegungen:

Zur Notation in Tabelle 17:

Die Notation 'PACE' und 'PACE + AUTH' in Tabelle 17 ist dahingehend zu verstehen, dass für den Fall des gesetzten Sicherheitszustandes PACE das betreffende Kommando in dem mittels des PACE-Protokolls aufgebauten sicheren Kanal zwischen GW und Sicherheitsmodul, also mit Secure Messaging gemäß Kap. 3.5 ausgeführt wird.

MF/DFs/EFs/Key-Objekte/PIN-Objekte	**Zugriffsregel**
MF	MF: • SELECT: ALWAYS • DELETE FILE: NEVER (MF nicht löschbar) • DEACTIVATE FILE: NEVER (MF nicht deaktivierbar) • TERMINATE DF: NEVER (MF nicht terminierbar) • CREATE FILE: PACE + AUTH • CREATE KEY: PACE + AUTH
	Über CREATE FILE im MF angelegte DFs und deren weitere über CREATE FILE angelegte Unterordner sind mit folgenden Zugriffsregeln belegt: • SELECT DF: ALWAYS • CREATE FILE: PACE + AUTH • DELETE FILE: PACE + AUTH • ACTIVATE FILE: PACE + AUTH • DEACTIVATE FILE: PACE + AUTH • TERMINATE DF: PACE + AUTH • CREATE KEY: PACE + AUTH Über CREATE FILE im MF und in weiteren ebenfalls über CREATE FILE angelegten Unterordnern des MF angelegte EFs sind mit folgenden Zugriffsregeln belegt: • SELECT EF: ALWAYS • READ BINARY / READ RECORD: PACE + AUTH • UPDATE BINARY / UPDATE RECORD: PACE + AUTH • APPEND RECORD: PACE + AUTH (sofern das Kommando im Sicherheitsmodul implementiert ist) • DELETE FILE: PACE + AUTH • ACTIVATE FILE: PACE + AUTH • DEACTIVATE FILE: PACE + AUTH • TERMINATE EF: PACE + AUTH Zu den Zugriffsregeln für über CREATE KEY angelegte Key-Objekte siehe untenstehende Angaben in den Tabellenzeilen zu Key Pair-Objekten und Public Key-Objekten.
EF.SecModTRInfo (Technisches Datenfeld mit Informationen zu der für das vorliegende Sicherheitsmodul relevanten Spezifikation)	• SELECT EF: ALWAYS • READ RECORD: ALWAYS • UPDATE RECORD: NEVER, ausgenommen das Sicherheitsmodul bietet die Funktionalität des Sicherheitsmodul-Firmware-Updates (in diesem Fall gelten die Zugriffsregeln für ein Update des Datenfeldes wie im Rahmen

MF/DFs/EFs/Key-Objekte/PIN-Objekte	Zugriffsregel
	des Firmware-Updates vorgesehen) • APPEND RECORD: PACE (sofern das Kommando im Sicherheitsmodul implementiert ist) • DELETE FILE: NEVER • ACTIVATE FILE: NEVER • DEACTIVATE FILE: NEVER • TERMINATE EF: NEVER
EF.SecModAccess (Technisches Datenfeld mit Informationen zu der vom vorliegenden Sicherheitsmodul angebotenen PACE-Funktionalität)	• SELECT EF: ALWAYS • READ BINARY: ALWAYS • UPDATE BINARY: NEVER, ausgenommen das Sicherheitsmodul bietet die Funktionalität des Sicherheitsmodul-Firmware-Updates (in diesem Fall gelten die Zugriffsregeln für ein Update des Datenfeldes wie im Rahmen des Firmware-Updates vorgesehen) • DELETE FILE: NEVER • ACTIVATE FILE: NEVER • DEACTIVATE FILE: NEVER • TERMINATE EF: NEVER
EF.SecModCrypto (Technisches Datenfeld mit Informationen zu der vom vorliegenden Sicherheitsmodul angebotenen Krypto-Funktionalität)	• SELECT EF: ALWAYS • READ BINARY: ALWAYS • UPDATE BINARY: NEVER, ausgenommen das Sicherheitsmodul bietet die Funktionalität des Sicherheitsmodul-Firmware-Updates (in diesem Fall gelten die Zugriffsregeln für ein Update des Datenfeldes wie im Rahmen des Firmware-Updates vorgesehen) • DELETE FILE: NEVER • ACTIVATE FILE: NEVER • DEACTIVATE FILE: NEVER • TERMINATE EF: NEVER
EF.SecModLifeCycle (Technisches Datenfeld mit Informationen zum Life Cycle-Status des vorliegenden Sicherheitsmoduls)	• SELECT EF: ALWAYS • READ RECORD: PACE • UPDATE RECORD: PACE • APPEND RECORD: PACE (sofern das Kommando im Sicherheitsmodul implementiert ist) • DELETE FILE: PACE • ACTIVATE FILE: PACE • DEACTIVATE FILE: PACE • TERMINATE EF: PACE
DF.SMGW	DF.SMGW: • SELECT DF: ALWAYS • CREATE FILE: PACE + AUTH • DELETE FILE: NEVER • ACTIVATE FILE: PACE + AUTH

MF/DFs/EFs/Key-Objekte/PIN-Objekte	Zugriffsregel
	• DEACTIVATE FILE: PACE + AUTH • TERMINATE DF: PACE + AUTH • CREATE KEY: PACE + AUTH
	Über CREATE FILE im DF.SMGW angelegte DFs und deren weitere über CREATE FILE angelegte Unterordner sind mit folgenden Zugriffsregeln belegt: • SELECT DF: ALWAYS • CREATE FILE: PACE + AUTH • DELETE FILE: PACE + AUTH • ACTIVATE FILE: PACE + AUTH • DEACTIVATE FILE: PACE + AUTH • TERMINATE DF: PACE + AUTH • CREATE KEY: PACE + AUTH Über CREATE FILE im DF.SMGW und in weiteren ebenfalls über CREATE FILE angelegten Unterordnern des DF.SMGW angelegte EFs sind mit folgenden Zugriffsregeln belegt: • SELECT EF: ALWAYS • READ BINARY / READ RECORD: PACE + AUTH • UPDATE BINARY / UPDATE RECORD: PACE + AUTH • APPEND RECORD: PACE + AUTH (sofern das Kommando im Sicherheitsmodul implementiert ist) • DELETE FILE: PACE + AUTH • ACTIVATE FILE: PACE + AUTH • DEACTIVATE FILE: PACE + AUTH • TERMINATE EF: PACE + AUTH Zu den Zugriffsregeln für über CREATE KEY angelegte Key-Objekte siehe untenstehende Angaben in den Tabellenzeilen zu Key Pair-Objekten und Public Key-Objekten.
EF.SMPKIRoot_x (Datenfelder für SM-PKI-Root-Zertifikate)	• SELECT EF: ALWAYS • READ BINARY: PACE • UPDATE BINARY: PACE + AUTH • DELETE FILE: PACE + AUTH • ACTIVATE FILE: PACE + AUTH • DEACTIVATE FILE: PACE + AUTH • TERMINATE EF: PACE + AUTH
EF.GSCert_TLS, EF.GSCert_SIG, EF.GSCert_ENC (Datenfelder für Gütesiegel-Zertifikate)	• SELECT EF: ALWAYS • READ BINARY: PACE • UPDATE BINARY: NEVER • DELETE FILE: NEVER • ACTIVATE FILE: NEVER • DEACTIVATE FILE: NEVER • TERMINATE EF: NEVER

MF/DFs/EFs/Key-Objekte/PIN-Objekte	Zugriffsregel
	Ein Überschreiben der Gütesiegel-Zertifikate über das Kommando UPDATE BINARY in den Phasen „Personalisierung des SMGW“ und „Normalbetrieb des SMGW“ soll nicht möglich sein. Hierzu wird nach dem Schreiben der Gütesiegel-Zertifikate in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ der LCSI der betreffenden EFs über das Kommando ACTIVATE FILE von „initialisation“ auf „operational state – activated“ umgesetzt. Die Ausführung des Kommandos UPDATE BINARY ist für diese EFs nur beim LCSI „initialisation“ möglich.
EF.GWKeys (Datenfeld für symmetrische GW-Keys)	• SELECT EF: ALWAYS • READ RECORD: PACE • UPDATE RECORD: PACE • APPEND RECORD: PACE (sofern das Kommando im Sicherheitsmodul implementiert ist) • DELETE FILE: PACE • ACTIVATE FILE: PACE • DEACTIVATE FILE: PACE • TERMINATE EF: PACE
PIN-Objekte (PIN.GW zur Speicherung der GW-System-PIN)	Generell: • PINs können nicht ausgelesen werden • Zugriff über PIN-Management-Kommandos bzw. Krypto-Kommandos wie in oben stehender Tabelle 16 angegeben, mit untenstehenden Ausnahmen für PIN.GW • zusätzlich zu den Angaben in Tabelle 16 bestehen je nach PIN-Management-Kommando ggf. weitere Zugriffsbeschränkungen in Abhängigkeit vom PIN-LifeCycleStatus des betreffenden PIN-Objektes: für Details siehe die Kommando-Spezifikation in Kap. 4.8
	PIN.GW: • CHANGE REFERENCE DATA / Variante Setzen einer PIN: NEVER • CHANGE REFERENCE DATA / Variante Wechseln einer PIN: PACE • GENERAL AUTHENTICATE / PACE: ALWAYS Hinweis: Das Kommando CHANGE REFERENCE DATA in der Variante „Setzen einer PIN“ ist für das PIN-Objekt PIN.GW für die GW-System-PIN in den Phasen „Personalisierung des SMGW“ und „Normalbetrieb des SMGW“ nicht mehr ausführbar. Dies wird über das Kommando-Verhalten des Kommandos CHANGE REFERENCE DATA in der Variante „Setzen einer PIN“ sowie über das PIN-Attribut PIN-LifeCycleStatus des PIN-Objektes PIN.GW erreicht, das im Initialisierungsfile mit dem Wert „initialisation“ vorbelegt ist und über den Aufruf des Kommandos CHANGE REFERENCE DATA in der Variante „Setzen einer PIN“ in der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ irreversibel auf den Wert „operational state – activated“ umgesetzt wird.

MF/DFs/EFs/Key-Objekte/PIN-Objekte	Zugriffsregel
Key Pair-Objekte	Generell: • Schlüsselpaare werden ausschließlich onboard generiert (persistente / temporäre Speicherung im Sicherheitsmodul) • Private Keys verbleiben grundsätzlich im Sicherheitsmodul und können nicht ausgelesen werden • zugehörige Public Keys können ausgelesen werden • zusätzlich zu den Angaben in Tabelle 16 bestehen je nach Key Management- bzw. Krypto-Kommando ggf. weitere Zugriffsbeschränkungen in Abhängigkeit vom Key Pair-Objekt und seinem Key-LifeCycleStatus: für Details siehe die folgenden Angaben und die jeweilige Kommando-Spezifikation in Kap. 3.4.5, 3.4.6, 4.4 und 4.5 sowie Kap. 3.2.4.2.3 und 3.3.2 d1)
	Speziell gelten für die für das Initialisierungsfile in Kap. 3.1.2 vordefinierten Key Pair-Objekte folgende Zugriffsregeln: Key.WAN_TLS_PRE, Key.WAN_SIG_PRE: • GENERATE ASYMMETRIC KEY PAIR / Export Public Key: PACE • GENERATE ASYMMETRIC KEY PAIR / KeyGen: NEVER • PSO COMPUTE DIGITAL SIGNATURE: PACE • DELETE KEY: NEVER • ACTIVATE KEY: NEVER • DEACTIVATE KEY: NEVER Key.WAN_ENC_PRE: • GENERATE ASYMMETRIC KEY PAIR / Export Public Key: PACE • GENERATE ASYMMETRIC KEY PAIR / KeyGen: NEVER • INTERNAL AUTHENTICATE : PACE • GENERAL AUTHENTICATE / ECKA-EG: PACE • DELETE KEY: NEVER • ACTIVATE KEY: NEVER • DEACTIVATE KEY: NEVER Key.WAN_TLS, Key.WAN_SIG: • GENERATE ASYMMETRIC KEY PAIR / Export Public Key: PACE • GENERATE ASYMMETRIC KEY PAIR / KeyGen: PACE + AUTH • PSO COMPUTE DIGITAL SIGNATURE: PACE • DELETE KEY: PACE + AUTH • ACTIVATE KEY: PACE + AUTH • DEACTIVATE KEY: PACE + AUTH Key.WAN_ENC: • GENERATE ASYMMETRIC KEY PAIR / Export Public Key:

MF/DFs/EFs/Key-Objekte/PIN-Objekte	Zugriffsregel
	PACE • GENERATE ASYMMETRIC KEY PAIR / KeyGen: PACE + AUTH • INTERNAL AUTHENTICATE: PACE • GENERAL AUTHENTICATE / ECKA-EG: PACE • DELETE KEY: PACE + AUTH • ACTIVATE KEY: PACE + AUTH • DEACTIVATE KEY: PACE + AUTH Key.IMP_TRANS, Key.IMP: Diese Key Pair-Objekte sind in den Phasen „Personalisierung des SMGW“ und „Normalbetrieb des SMGW“ nicht mehr im Sicherheitsmodul vorhanden. Key.EPH_x: • GENERATE ASYMMETRIC KEY PAIR / KeyGen mit Ausgabe des Public Key: PACE • GENERAL AUTHENTICATE / ECKA-DH: PACE Für alle zuvor genannten Key Pair-Objekte des Initialisierungsfiles wie in Kap. 3.1.2 definiert sind jeweils alle weiteren Kommandos zum Key Management sowie alle weiteren Krypto-Kommandos nicht zulässig (Zugriffsregel NEVER).
	Über das Kommando CREATE KEY (persistent) angelegte Key Pair-Objekte werden automatisch mit folgenden Zugriffsregeln belegt: Generell: • DELETE KEY: PACE + AUTH • ACTIVATE KEY: PACE + AUTH • DEACTIVATE KEY: PACE + AUTH Ferner: 1) für Key Pair-Objekte der AT-Anwendungsklasse: falls Key-CryptoAlg = id-ecka-eg: • GENERATE ASYMMETRIC KEY PAIR / Export Public Key: PACE • GENERATE ASYMMETRIC KEY PAIR / KeyGen: PACE + AUTH • GENERAL AUTHENTICATE / ECKA-EG: PACE falls Key-CryptoAlg = id-ecdsa-plain-signatures: • GENERATE ASYMMETRIC KEY PAIR / Export Public Key: PACE • GENERATE ASYMMETRIC KEY PAIR / KeyGen: PACE + AUTH • INTERNAL AUTHENTICATE: PACE falls Key-CryptoAlg = id-ecdsa-plain-SHA: nicht relevant 2) für Key Pair-Objekte der DST-Anwendungsklasse:

MF/DFs/EFs/Key-Objekte/PIN-Objekte	Zugriffsregel
	falls Key-CryptoAlg = id-ecdsa-plain-signatures: • GENERATE ASYMMETRIC KEY PAIR / Export Public Key: PACE • GENERATE ASYMMETRIC KEY PAIR / KeyGen: PACE + AUTH • PSO COMPUTE DIGITAL SIGNATURE: PACE falls Key-CryptoAlg = id-ecdsa-plain-SHA: nicht relevant Die zuvor verwendete Angabe 'nicht relevant' bei einzelnen Angaben zu Key-CryptoAlg bedeutet, dass es im Smart Meter-System derzeit nicht zwingend erforderlich ist, entsprechende Key-Objekte mit dem betreffenden Wert für das Key-Attribut Key-CryptoAlg anlegen zu können, da es keine entsprechenden Anwendungsfälle gibt und solche Key-Objekte daher später im Betrieb nicht genutzt werden würden. Für die Implementierung des Kommandos CREATE KEY siehe auch Kap. 4.4.1. Alle weiteren Kommandos zum Key Management sowie alle weiteren Krypto-Kommandos sind nicht zulässig (Zugriffsregel NEVER).
Public Key-Objekte	Generell: • Public Keys können importiert werden (temporäre / persistente Speicherung im Sicherheitsmodul) • zusätzlich zu den Angaben in Tabelle 16 bestehen je nach Key Management- bzw. Krypto-Kommando ggf. weitere Zugriffsbeschränkungen in Abhängigkeit vom Public Key-Objekt und seinem Key-LifeCycleStatus: für Details siehe die folgenden Angaben und die jeweilige Kommando-Spezifikation in Kap. 3.4.5, 3.4.6, 4.4 und 4.5 sowie Kap. 3.2.4.2.3 und 3.3.2 d2)
	Speziell gelten für die für das Initialisierungsfile in Kap. 3.1.2 vordefinierten Public Key-Objekte folgende Zugriffsregeln: Key.SMPKIRoot_x: • PSO VERIFY DIGITAL SIGNATURE / Variante ohne Übergabe des Public Key im Kommando: PACE • PSO VERIFY DIGITAL SIGNATURE / Variante mit Übergabe des Public Key im Kommando: --- (siehe Angabe in Tabelle 16) • DELETE KEY: PACE + AUTH • ACTIVATE KEY: PACE + AUTH • DEACTIVATE KEY: PACE + AUTH Key.GWA_TLS_x: • PSO VERIFY DIGITAL SIGNATURE / Variante ohne Übergabe des Public Key im Kommando: PACE • PSO VERIFY DIGITAL SIGNATURE / Variante mit Übergabe des Public Key im Kommando: --- (siehe Angabe in Tabelle 16) • DELETE KEY: PACE + AUTH • ACTIVATE KEY: PACE + AUTH

MF/DFs/EFs/Key-Objekte/PIN-Objekte	Zugriffsregel
	• DEACTIVATE KEY: PACE + AUTH Key.GWA_SIG_x: • PSO VERIFY DIGITAL SIGNATURE / Variante ohne Übergabe des Public Key im Kommando: PACE • PSO VERIFY DIGITAL SIGNATURE / Variante mit Übergabe des Public Key im Kommando: --- (siehe Angabe in Tabelle 16) • PSO VERIFY CERTIFICATE: PACE + AUTH • DELETE KEY: PACE + AUTH • ACTIVATE KEY: PACE + AUTH • DEACTIVATE KEY: PACE + AUTH Key.GWA_ENC_x: • GENERAL AUTHENTICATE / ECKA-EG: PACE • DELETE KEY: PACE + AUTH • ACTIVATE KEY: PACE + AUTH • DEACTIVATE KEY: PACE + AUTH Key.GWA_AUT_x: • EXTERNAL AUTHENTICATE: PACE • DELETE KEY: NEVER • ACTIVATE KEY: PACE + AUTH • DEACTIVATE KEY: PACE + AUTH Key.CA_x: • PSO VERIFY DIGITAL SIGNATURE / Variante ohne Übergabe des Public Key im Kommando: PACE • PSO VERIFY DIGITAL SIGNATURE / Variante mit Übergabe des Public Key im Kommando: --- (siehe Angabe in Tabelle 16) • DELETE KEY: PACE + AUTH • ACTIVATE KEY: PACE + AUTH • DEACTIVATE KEY: PACE + AUTH Key.IMP_PUB_TRANS, Key.IMP_PUB: Diese Public Key-Objekte sind in den Phasen „Personalisierung des SMGW“ und „Normalbetrieb des SMGW“ nicht mehr im Sicherheitsmodul vorhanden. Für alle zuvor genannten Public Key-Objekte des Initialisierungsfiles wie in Kap. 3.1.2 definiert sind jeweils alle weiteren Kommandos zum Key Management sowie alle weiteren Krypto-Kommandos nicht zulässig (Zugriffsregel NEVER).
	Über das Kommando CREATE KEY (persistent) angelegte Public Key-Objekte werden automatisch mit folgenden Zugriffsregeln belegt: Generell:

MF/DFs/EFs/Key-Objekte/PIN-Objekte	Zugriffsregel
	• DELETE KEY: PACE + AUTH • ACTIVATE KEY: PACE + AUTH • DEACTIVATE KEY: PACE + AUTH Ferner: 1) für Public Key-Objekte der AT-Anwendungsklasse: falls Key-CryptoAlg = id-ecka-eg: • GENERAL AUTHENTICATE / ECKA-EG: PACE falls Key-CryptoAlg = id-ecdsa-plain-signatures: nicht relevant falls Key-CryptoAlg = id-ecdsa-plain-SHA: nicht relevant 2) für Public Key-Objekte der DST-Anwendungsklasse: falls Key-CryptoAlg = id-ecdsa-plain-signatures: • PSO VERIFY DIGITAL SIGNATURE: PACE falls Key-CryptoAlg = id-ecdsa-plain-SHA: • PSO VERIFY CERTIFICATE: PACE + AUTH Die zuvor verwendete Angabe 'nicht relevant' bei einzelnen Angaben zu Key-CryptoAlg bedeutet, dass es im Smart Meter-System derzeit nicht zwingend erforderlich ist, entsprechende Key-Objekte mit dem betreffenden Wert für das Key-Attribut Key-CryptoAlg anlegen zu können, da es keine entsprechenden Anwendungsfälle gibt und solche Key-Objekte daher später im Betrieb nicht genutzt werden würden. Für die Implementierung des Kommandos CREATE KEY siehe auch Kap. 4.4.1. Alle weiteren Kommandos zum Key Management sowie alle weiteren Krypto-Kommandos sind nicht zulässig (Zugriffsregel NEVER).

Tabelle 17: Zugriffsregeln für MF/DFs/EFs/Key-Objekte/PIN-Objekte in den Phasen „Personalisierung des SMGW“ und „Normalbetrieb des SMGW“ (SEID = 02)

3.4 Kommandoset des Sicherheitsmoduls

Im vorliegenden Kapitel wird das vom Sicherheitsmodul bereitgestellte Set an Kommandos in Form einer Grobspezifikation beschrieben. Für die zugehörige Detail-Spezifikation der Kommandos sei auf Kap. 4 verwiesen.

3.4.1 Übersicht über das Kommandoset

Folgende Kommandos werden vom Sicherheitsmodul im Einzelnen für das GW angeboten:

Kommandos	Kapitel-Referenzen
Key Management	
CREATE KEY	3.4.5.1, 4.4.1
DELETE KEY	3.4.5.2, 4.4.2

Kommandos	Kapitel-Referenzen
ACTIVATE KEY	3.4.5.3, 4.4.3
DEACTIVATE KEY	3.4.5.4, 4.4.4
Krypto-Kommandos (asymmetrische Kryptographie)	
GENERATE ASYMMETRIC KEY PAIR / KeyGen (Schlüsselgenerierung mit/ohne Export des Public Key) und Export Public Key (Schlüsselexport ohne Schlüsselgenerierung)	3.4.6.2, 4.5.1
PSO COMPUTE DIGITAL SIGNATURE	3.4.6.3, 4.5.2
PSO VERIFY DIGITAL SIGNATURE	3.4.6.4, 4.5.3
PSO VERIFY CERTIFICATE	3.4.6.5, 4.5.4
GENERAL AUTHENTICATE / ECKA (-EG, -DH)	3.4.6.6, 4.5.5
GENERAL AUTHENTICATE / PACE	3.4.6.6, 4.5.5
EXTERNAL AUTHENTICATE	3.4.6.7, 4.5.6
INTERNAL AUTHENTICATE	3.4.6.8, 4.5.7
Management des Security Environments (SE)	
MSE SET	3.4.7.1, 4.6.1
MSE RESTORE	3.4.7.2, 4.6.2
Zufallszahlengenerierung	
GET CHALLENGE	3.4.8.1, 4.7.1
PIN-Management	
CHANGE REFERENCE DATA	3.4.9.2, 4.8.1
Zugriff auf transparente EF	
READ BINARY	3.4.4.1, 4.3.1
UPDATE BINARY	3.4.4.2, 4.3.2
Zugriff auf Record-orientierte EF	
READ RECORD	3.4.4.3, 4.3.3
UPDATE RECORD	3.4.4.4, 4.3.4
APPEND RECORD (optional)	3.4.4.5, 4.3.5
Kartenmanagement / Management des Filesystems	
SELECT (MF/DF/EF)	3.4.3.1, 4.2.1
CREATE FILE	3.4.3.2, 4.2.2
DELETE FILE	3.4.3.3, 4.2.3
ACTIVATE FILE	3.4.3.4, 4.2.4
DEACTIVATE FILE	3.4.3.5, 4.2.5
TERMINATE DF	3.4.3.6, 4.2.6

Kommandos	Kapitel-Referenzen
TERMINATE EF	3.4.3.7, 4.2.7
Management des Life Cycle-Status	
TERMINATE CARD USAGE	3.4.10.1, 4.9.1
Management der Applikationsebene	
MANAGE CHANNEL	3.4.11.1, 4.10.1

Tabelle 18: Kommandoset des Sicherheitsmoduls

3.4.2 Generelles zu den Kommandos des Sicherheitsmoduls

Das Sicherheitsmodul unterstützt für die Codierung von Längenfeldern (Lc, Le) in Kommando-APDUs das Extended Length-Format. Dies wird insbesondere erforderlich für das Importieren/Speichern und Exportieren/Auslesen von Zertifikaten sowie bei den Krypto-Kommandos je nach Krypto-Algorithmus bzw. Schlüssellänge. Ferner ist ggf. die Zufallszahlengenerierung (Kommando GET CHALLENGE) betroffen. Die Unterstützung des Extended Length-Formates durch das Sicherheitsmodul wird im ATR/ATS (siehe Kap. 3.6.2) in den Historical Bytes oder auf andere geeignete Weise (z.B. im EF.ATR/INFO wie in [ISO 7816-4] spezifiziert) angezeigt.

Command Chaining (siehe [ISO 7816-4]) wird nur für das Kommando GENERAL AUTHENTICATE / Variante PACE verwendet.

3.4.3 Kartenmanagement / Management des Filesystems

Im folgenden werden die vom Sicherheitsmodul angebotenen Kommandos für das Kartenmanagement bzw. das Management des Filesystems beschrieben.

3.4.3.1 Kommando SELECT

Kommando **SELECT** ([ISO 7816-4], Kap. 11.1.1)

- Auswahl des MF.
- Auswahl eines DF (hier z.B. DF.SMGW für die SMGW-Applikation).
- Auswahl eines EF (hier: für Read- und Update-Zugriff auf im Sicherheitsmodul gespeicherte symmetrische Schlüssel und Zertifikate).
- Die Selektion eines DF erhält alle Sicherheitszustände entlang des Pfads vom MF bis zum selektierten DF (einschließlich).

3.4.3.2 Kommando CREATE FILE

Kommando **CREATE FILE** ([ISO 7816-9], Kap. 6.1)

- Anlegen eines DF / EF.

- Das Kommando muss nicht zwingend in seiner vollen Bandbreite wie gemäß [ISO 7816-9] möglich implementiert werden; es ist eine „reduzierte“, auf die im SMGW benötigte Funktionalität beschränkte Version ausreichend. Jedoch ist die Anlage von Ordnern (DFs) und transparenten und Record-orientierten Datenfeldern (EFs) bereitzustellen.

3.4.3.3 Kommando DELETE FILE

Kommando **DELETE FILE** ([ISO 7816-9], Kap. 6.2)

- Irreversibles Löschen eines DF / EF.

  Hinweis: Das Löschen des MF wird nicht benötigt. Siehe auch Kap. 3.3.2 a).
- Referenzierung des Files über ein vorhergehendes SELECT-Kommando.

3.4.3.4 Kommando ACTIVATE FILE

Kommando **ACTIVATE FILE** ([ISO 7816-9], Kap. 6.4)

- Reversibles Aktivieren eines DF / EF (ggf. auch des MF).
- Beeinflussung des LCSI des DF / EF. Das Kommando setzt den LCSI auf „operational state – activated“ (sofern der aktuelle LCSI nicht mit „terminated“ belegt ist).
- Referenzierung des Files über ein vorhergehendes SELECT-Kommando.

3.4.3.5 Kommando DEACTIVATE FILE

Kommando **DEACTIVATE FILE** ([ISO 7816-9], Kap. 6.3)

- Reversibles Deaktivieren eines DF / EF.

  Hinweis: Das Deaktivieren des MF wird nicht benötigt. Siehe auch Kap. 3.3.2 a).
- Beeinflussung des LCSI des DF / EF. Das Kommando setzt den LCSI auf „operational state – deactivated“ (sofern der aktuelle LCSI nicht mit „terminated“ belegt ist).
- Referenzierung des Files über ein vorhergehendes SELECT-Kommando.

3.4.3.6 Kommando TERMINATE DF

Kommando **TERMINATE DF** ([ISO 7816-9], Kap. 6.5)

- Irreversibles Terminieren eines DF.

  Hinweis: Das Löschen des MF wird nicht benötigt. Siehe auch Kap. 3.3.2 a).
- Beeinflussung des Terminieren des MF. Das Kommando setzt den LCSI auf „terminated“.
- Referenzierung des Files über ein vorhergehendes SELECT-Kommando.

3.4.3.7 Kommando TERMINATE EF

Kommando **TERMINATE EF** ([ISO 7816-9], Kap. 6.6)

- Irreversibles Terminieren eines EF.
- Beeinflussung des LCSI des EF. Das Kommando setzt den LCSI auf „terminated“.
- Referenzierung des Files über ein vorhergehendes SELECT-Kommando.

3.4.4 Kommandos für den Zugriff auf Datenfelder

Im Filesystem des Sicherheitsmoduls werden insbesondere Datenfelder für die Speicherung von Daten (z.B. Nutzdaten wie Zertifikate oder technische Informationen) angelegt und verwaltet. Diese Datenfelder sind als transparente oder Record-orientierte EF ausgelegt und benötigen Lese- sowie Schreibzugriff.

Die in einem EF jeweils zu speichernde Datenmenge ist im Smart Meter-System relativ klein und liegt unter 32 KB. Aufgrund der Datengröße wird jedoch für die Lese- und Schreibkommandos das Extended Length-Format erforderlich. Command Chaining wird nicht verwendet.

Im folgenden werden die vom Sicherheitsmodul angebotenen Kommandos für den Zugriff auf Datenfelder beschrieben.

3.4.4.1 Kommando READ BINARY

Kommando **READ BINARY** ([ISO 7816-4], Kap. 11.2.3)

- Auslesen eines transparenten EF.
- Referenzierung des Files über ein vorhergehendes SELECT-Kommando oder via SFI-Referenz im Kommando.

3.4.4.2 Kommando UPDATE BINARY

Kommando **UPDATE BINARY** ([ISO 7816-4], Kap. 11.2.5)

- Schreiben / Update eines transparenten EF.
- Referenzierung des Files über ein vorhergehendes SELECT-Kommando oder via SFI-Referenz im Kommando.

3.4.4.3 Kommando READ RECORD

Kommando **READ RECORD** ([ISO 7816-4], Kap. 11.3.3)

- Auslesen eines Record-orientierten EF.
- Referenzierung des Files über ein vorhergehendes SELECT-Kommando oder via SFI-Referenz im Kommando.

3.4.4.4 Kommando UPDATE RECORD

Kommando **UPDATE RECORD** ([ISO 7816-4], Kap. 11.3.5)

- Schreiben / Update eines Record-orientierten EF.
- Referenzierung des Files über ein vorhergehendes SELECT-Kommando oder via SFI-Referenz im Kommando.

3.4.4.5 Kommando APPEND RECORD (optional)

Kommando **APPEND RECORD** ([ISO 7816-4], Kap. 11.3.6)

- Anlegen eines weiteren Records in einem Record-orientierten EF.
- Referenzierung des Files über ein vorhergehendes SELECT-Kommando oder via SFI-Referenz im Kommando.

3.4.5 Kommandos für das Key Management

Im folgenden werden die vom Sicherheitsmodul angebotenen Kommandos zum Key Management beschrieben. Diese Kommandos beziehen sich nur auf persistent im Sicherheitsmodul zu speichernde bzw. persistent im Sicherheitsmodul gespeicherte Key-Objekte.

3.4.5.1 Kommando CREATE KEY

Kommando **CREATE KEY**

- Anlegen eines (leeren) persistent gespeicherten Key-Objektes mit seiner internen Speicherstruktur wie Betriebssystem-spezifisch vorgesehen (inkl. Allokation von ausreichend viel Speicherplatz für das neue Key-Objekt).
- Übergabe von Key-ID/Key-Name im Kommando-Datenfeld.
- Übergabe weiterer Informationen zum Key-Objekt im Kommando-Datenfeld (wie z.B. die Information, ob ein Key Pair-Objekt oder Public Key-Objekt angelegt werden soll, für welchen Zweck das Key-Objekt vorgesehen ist usw.).
- Die Anlage des neuen Key-Objektes erfolgt im aktuell selektierten DF.

3.4.5.2 Kommando DELETE KEY

Kommando **DELETE KEY**

- Komplettes, physikalisches Löschen eines Key-Objektes (inkl. Löschen der Key-Daten und Key-Zusatzinformationen).
- Übergabe von Key Reference und Anwendungsklasse des Key-Objektes im Kommando-Datenfeld.
- Für Key Pair-Objekte erfolgt die Suche nach dem zu löschenden Key-Objekt lokal ausgehend vom aktuell selektierten DF bzw. global (je nach MSBit in der Key Reference).

Für Public Key-Objekte erfolgt stets eine lokale Suche ausgehend vom aktuell selektierten DF.

3.4.5.3 Kommando ACTIVATE KEY

Kommando **ACTIVATE KEY**

- Reversibles Aktivieren eines Keys-Objektes.
- Übergabe von Key Reference und Anwendungsklasse des Key-Objektes im Kommando-Datenfeld.
- Beeinflussung des Key-Attributs Key-LifeCycleStatus. Das Kommando setzt Key-LifeCycleStatus auf „operational state – activated“.
- Für Key Pair-Objekte erfolgt die Suche nach dem zu aktivierenden Key-Objekt lokal ausgehend vom aktuell selektierten DF bzw. global (je nach MSBit in der Key Reference). Für Public Key-Objekte erfolgt stets eine lokale Suche ausgehend vom aktuell selektierten DF.

3.4.5.4 Kommando DEACTIVATE KEY

Kommando **DEACTIVATE KEY**

- Reversibles Deaktivieren eines Key-Objektes.
- Übergabe von Key Reference und Anwendungsklasse des Key-Objektes im Kommando-Datenfeld.
- Beeinflussung des Key-Attributs Key-LifeCycleStatus. Das Kommando setzt Key-LifeCycleStatus auf „operational state – deactivated“.
- Für Key Pair-Objekte erfolgt die Suche nach dem zu deaktivierenden Key-Objekt lokal ausgehend vom aktuell selektierten DF bzw. global (je nach MSBit in der Key Reference). Für Public Key-Objekte erfolgt stets eine lokale Suche ausgehend vom aktuell selektierten DF.

3.4.5.5 Hinweise zum Key Management (informativ)

Weitere Kommandos für das Key Management sind denkbar und können für zukünftige Migrationsschritte im Smart Meter-System aufgenommen werden.

Das Auslesen eines Public Key wird allenfalls relevant bei onboard-generierten Key-Paaren. Auf das Kommando GET KEY zum Auslesen eines solchen Public Key kann verzichtet werden, da der öffentliche Part eines im Sicherheitsmodul generierten Key-Paares über das Kommando GENERATE ASYMMETRIC KEY PAIR ausgelesen werden kann.

Das Anlegen eines Key-Objektes erfolgt über das Kommando CREATE KEY. Im Falle eines Key Pair-Objektes wird dieses bei der Ausführung des Kommandos GENERATE ASYMMETRIC KEY PAIR mit den Key-Daten sowie den Key-Zusatzinformationen gefüllt. Im Falle eines Public Key-Objektes wird dieses bei der Ausführung des Kommandos PSO VERIFY CERTIFICATE mit den Key-Daten sowie den Key-Zusatzinformationen gefüllt.

Auf ein Kommando PUT KEY zum Importieren von Public Keys soll derzeit verzichtet werden, da entsprechende Standardisierungsarbeiten noch nicht abgeschlossen sind. Als Alternative erfolgt das Importieren von Public Keys derzeit über das etablierte Kommando PSO VERIFY CERTIFICATE.

Auf ein Kommando DELETE KEY zum Löschen von Key-Daten bei gleichzeitigem Erhalt des zugehörigen Key-Objektes soll derzeit verzichtet werden, da entsprechende Standardisierungsarbeiten noch nicht abgeschlossen sind. Über das Kommando DELETE KEY kann aber ein Löschen des gesamten Key-Objektes inkl. der gespeicherten Key-Daten und Key-Zusatzinformationen erfolgen.

Auf ein Kommando TERMINATE KEY zum Terminieren von Key-Objekten soll derzeit verzichtet werden, da entsprechende Standardisierungsarbeiten noch nicht abgeschlossen sind. Statt des Terminierens eines Key-Objektes genügt es im Rahmen des Smart Meter-Systems derzeit, das betreffende Key-Objekt mittels des Kommandos DEACTIVATE KEY zu deaktivieren und anschließend ggf. über das Kommando DELETE KEY komplett zu löschen bzw. über das Kommando GENERATE ASYMMETRIC KEY PAIR neu mit Daten zu befüllen (und damit die alten Daten zu überschreiben).

3.4.6 Kommandos für kryptographische Anwendungen und Protokolle

Im folgenden werden die vom Sicherheitsmodul angebotenen Kommandos für kryptographische Anwendungen und Protokolle beschrieben.

3.4.6.1 Generelles zu den Krypto-Kommandos

Die Berechnung von Hashwerten für die Signaturerzeugung und -prüfung im Rahmen der Kommandos PSO COMPUTE DIGITAL SIGNATURE und PSO VERIFY DIGITAL SIGNATURE findet komplett im GW statt. Entsprechendes gilt für das Kommando INTERNAL AUTHENTICATE. Das Sicherheitsmodul ist an dieser Stelle in die Berechnung von Hashwerten *nicht* involviert, so dass ein Kommando wie PSO HASH nicht erforderlich ist.

Für die Ausführung der Kommandos PSO VERIFY CERTIFICATE, GENERAL AUTHENTICATE (Variante PACE) und EXTERNAL AUTHENTICATE benötigt das Sicherheitsmodul für interne Berechnungen aber gleichwohl Hashverfahren (wie sie in [TR-03109-3] spezifiziert sind).

Für die Verwendung von Hashverfahren gilt: Die Bitlänge des Outputs der Hash-Funktion sollte nicht kleiner als die Bitlänge des Basispunktes der für das betreffende Krypto-Kommando und den betreffenden Schlüssel verwendeten Elliptischen Kurve gewählt werden. Für den Fall, dass bei der für ein Krypto-Kommando verwendeten Hash-Funktion die Bitlänge des Outputs der Hash-Funktion größer als die Bitlänge des Basispunktes der verwendeten Elliptischen Kurve (wie für den betreffenden Schlüssel relevant) ist, erfolgt eine Kürzung und Codierung des zu verarbeitenden Hashwertes gemäß [TR-03111], Kap. 4.1.2 und 4.2 als „truncated hash value“.

Hinweis: Das Sicherheitsmodul unterstützt durch Bereitstellung der Kernroutine zur Signaturprüfung (PSO VERIFY DIGITAL SIGNATURE) die Prüfung von X.509-Zertifikaten und -Zertifikatsketten. Die Prüfung von X.509-Zertifikaten und -Zertifikatsketten findet eigentlich im GW statt; das GW bedient sich hierzu der Signaturprüfungs-Routine des Sicherheitsmoduls und bereitet die Daten für die Signaturprüfung im Sicherheitsmodul entsprechend auf. Das Sicherheitsmodul selbst stellt keine Funktionalität zur kompletten Prüfung von X.509-Zertifikaten und -Zertifikatsketten bereit. (Üblicherweise prüfen Sicherheitsmodule bzw. Chipkarten allenfalls

CV-Zertifikate; für die Prüfung von X.509-Zertifikaten müssten proprietäre Routinen aufgesetzt werden.)

An der Kommando-Schnittstelle des Sicherheitsmoduls werden Elliptische Kurven sowie Krypto-Algorithmen und -Protokolle grundsätzlich über ihre zugehörigen OIDs referenziert (siehe Kap. 3.2.4.5). Ausnahme bildet lediglich das MSE-Kommando (SET-Variante mit AT-Template) zum Setzen der Elliptischen Kurve für das Kommando GENERAL AUTHENTICATE / Variante PACE; hier wird im MSE-Kommando die Elliptische Kurve über ihre ID wie in [TR-03110-3], Abschnitt A.2.1.1 angegeben referenziert.

Die Verwendung von OIDs hat gegenüber der Verwendung von Algorithmen-IDs den Vorteil, dass die für das Sicherheitsmodul gewünschte Interoperabilität unterstützt wird. Unbenommen hiervon ist die Betriebssystem-interne Referenzierung, Codierung, Speicherung und Verarbeitung der Informationen zu Elliptischen Kurven sowie Krypto-Algorithmen, z.B. in den Key-Zusatzinformationen der Key-Objekte. Anstelle von OIDs können Betriebssystem-intern die benötigten Informationen auch Hersteller-spezifisch in anderer Form referenziert, codiert, gespeichert und verarbeitet werden (z.B. Realisierung über ein geeignetes Mapping zwischen den OIDs und der Hersteller-spezifischen internen Codierung).

Die Krypto-Kommandos beinhalten jeweils:

- Schlüsselsuche anhand von Key Reference und ggf. Anwendungsklasse des Key-Objektes (sofern die Schlüsselsuche nicht zuvor schon in einem vorangehenden MSE-Kommando erfolgt ist; Hersteller-abhängige Implementierung des MSE- bzw. Krypto-Kommandos; siehe auch Kap. 3.4.7.1 und 4.6.1)
- Auswertung des Key-LifeCycleStatus
- Auswertung des Key-UsageCounter (sofern vorhanden)
- Falls das Kommando ein vorhergehendes MSE SET-Kommando erwartet:

  Die Konsistenzprüfung der Key-Zusatzinformationen aus dem Key-Objekt gegen die entsprechenden Informationen im SE, das zuvor über das MSE SET-Kommando gesetzt wurde, erfolgt Hersteller-spezifisch im MSE-Kommando oder im Krypto-Kommando (siehe auch Kap. 3.4.7.1 und 4.6.1).
- Zugriffsregelprüfung
- Ausführung der jeweiligen Krypto-Operation

3.4.6.2 Kommando GENERATE ASYMMETRIC KEY PAIR

Kommando **GENERATE ASYMMETRIC KEY PAIR** ([ISO 7816-8], Kap. 5.1)

- Verschiedene Kommando-Varianten:
  - Schlüsselgenerierung ohne Ausgabe des Public Key.
  - Schlüsselgenerierung mit gleichzeitiger Ausgabe des Public Key.
  - Ausgabe des Public Key eines zuvor onboard generierten Key-Paares.
- Über das Kommando GENERATE ASYMMETRIC KEY PAIR können persistente wie auch temporäre Key Pair-Objekte mit Schlüsseldaten gefüllt werden. Für persistente Key Pair-Objekte stehen alle drei Kommando-Varianten im Sicherheitsmodul zur Verfügung. Für

temporäre Key Pair-Objekte wird nur die Variante „Schlüsselgenerierung mit gleichzeitiger Ausgabe des Public Key“ benötigt.

- Entsprechend der Kommando-Spezifikation in [TR-03117] werden die für die Schlüsselgenerierung erforderlichen Informationen zum Key Pair-Objekt und zu den Schlüsselattributen im Datenfeld des Kommandos in CRT-Parametern übergeben, womit ein vorhergehendes MSE SET-Kommando nicht erforderlich ist.
- Für die beiden Kommando-Varianten mit Schlüsselgenerierung:
  - Diese beiden Kommando-Varianten setzen voraus, dass ein Key Pair-Objekt zur Ablage der Key-Daten und -Zusatzinformationen bereits vorhanden ist. Das Key Pair-Objekt für das zu generierende Key-Paar ist unter Angabe der Key-ID vor Aufruf des Kommandos GENERATE ASYMMETRIC KEY PAIR neu anzulegen (Kommando CREATE KEY), oder aber ein bereits vorhandenes Key Pair-Objekt wird (wieder-) verwendet und die dort ggf. gespeicherten Key-Daten und -Zusatzinformationen werden im Kommando GENERATE ASYMMETRIC KEY PAIR mit den neuen Daten überschrieben.
  - Im Kommando werden die Key Reference auf das zu befüllende Key Pair-Objekt sowie die Key-Paar-Zusatzinformationen (in Form von CRT-Parametern) übergeben. Das Kommando sucht das referenzierte Key Pair-Objekt, generiert das Key-Paar passend zu den übergebenen Key-Paar-Zusatzinformationen und befüllt das betreffende Key Pair-Objekt mit den Schlüsseldaten und Zusatzinformationen. Bei erfolgreicher Schlüsselgenerierung wird der Key-LifeCycleStatus defaultmäßig auf „operational state – activated“ gesetzt (siehe auch Kap. 3.2.4.2.3). Je nach Kommando-Variante erfolgt die Ausgabe des Public Key.
- Für die Kommando-Variante ohne Schlüsselgenerierung:
  - Diese Kommando-Variante setzt voraus, dass zuvor bereits ein Key-Paar (z.B. über eine der beiden anderen Kommando-Varianten) generiert und Sicherheitsmodul-intern gespeichert wurde.
  - Im Kommando wird die Key Reference auf das betreffende Key Pair-Objekt übergeben, dessen Public Key ausgegeben werden soll.

3.4.6.3 Kommando PSO COMPUTE DIGITAL SIGNATURE

Kommando **PSO COMPUTE DIGITAL SIGNATURE** ([ISO 7816-8], Kap. 5.4, [EN 14890-1], Kap. 7.4.1)

- Verwendung für die Erzeugung von Signaturen (z.B. im Rahmen der Inhaltsdatensignatur).
- Vor der Signaturerzeugung ist der zu verwendende Private Key über ein entsprechendes MSE-Kommando (SET-Variante mit DST-Template) zu setzen. Der zu verwendende Krypto-Algorithmus wird ebenfalls über das vorhergehende MSE SET-Kommando ausgewählt. Siehe MSE SET-Variante 1.1 in Kap. 4.6.1.
- In den Kommandodaten werden die zu signierenden Daten übergeben.

  (Hinweis: Das Hashing der Daten findet im GW statt.)
- Krypto-Algorithmen: ECDSA-Signaturerzeugung (mit Elliptischen Kurven und Schlüssellängen wie in [TR-03109-3] vorgegeben sowie ohne Hashing im Sicherheitsmodul), Variante „Signatur ohne Message Recovery“.

3.4.6.4 Kommando PSO VERIFY DIGITAL SIGNATURE

Kommando **PSO VERIFY DIGITAL SIGNATURE** ([ISO 7816-8], Kap. 5.7)

- Verwendung für die Prüfung von Signaturen (z.B. im Rahmen der Inhaltsdatensignatur und der Prüfung von Zertifikatsketten).
- 2 Kommando-Varianten:
  - 1. Variante: In den Kommandodaten werden die zu prüfende Signatur (inkl. aufbereitetem Hashwert) und der zu verwendende Public Key mit der zugehörigen Elliptischen Kurve übergeben.
  - 2. Variante: In den Kommandodaten wird die zu prüfende Signatur (inkl. aufbereitetem Hashwert) übergeben, wobei der zu verwendende Public Key zuvor über ein MSE-Kommando (SET-Variante mit DST-Template) gesetzt wird. Der zu verwendende Krypto-Algorithmus wird ebenfalls über das vorhergehende MSE SET-Kommando ausgewählt. Siehe MSE SET-Variante 1.2 in Kap. 4.6.1. In diesem Fall ist der Public Key zuvor in das Sicherheitsmodul zu importieren.

  (Hinweis: Das Hashing der Daten findet im GW statt.)
- Krypto-Algorithmen: ECDSA-Signaturprüfung (mit Elliptischen Kurven und Schlüssellängen wie in [TR-03109-3] vorgegeben sowie ohne Hashing im Sicherheitsmodul), Variante „Signatur ohne Message Recovery“.

3.4.6.5 Kommando PSO VERIFY CERTIFICATE

Kommando **PSO VERIFY CERTIFICATE** ([ISO 7816-8], Kap. 5.8)

- Verwendung für den Import von Public Keys (zur persistenten Speicherung).
- Für den zu speichernden Public Key muss vor Aufruf des Kommandos ein Public Key-Objekt vorhanden sein. Entweder wird ein solches Public Key-Objekt über das Kommando CREATE KEY zuvor neu angelegt, oder aber es existiert bereits ein Public Key-Objekt, das im Kommando mit dem zu importierenden Public Key gefüllt wird (Erst-Befüllung eines leeren Public Key-Objektes oder Überschreiben des alten Public Key im Public Key-Objekt).
- In den Kommandodaten wird ein Zertifikat (genauer: der zu verifizierende Body des Zertifikates und die zu verifizierende Signatur über den Zertifikatsbody) übergeben. Der Zertifikatsbody enthält dabei den zu importierenden Public Key mit Key-Zusatzinformationen sowie die Key Reference des zu befüllenden Public Key-Objektes.
- Verwendet werden nur Zertifikate des Typs „self-descriptive“, die nicht-selbstsigniert sind.
- Als Signaturprüfschlüssel für die Zertifikatssignatur zieht das Kommando einen Public Key heran, der bereits im Sicherheitsmodul vorhanden ist und über ein MSE-Kommando (SET-Variante mit DST-Template, Übergabe der Key Reference des zugehörigen Public Key-Objektes) gesetzt wurde. Der zu verwendende Krypto-Algorithmus, insbesondere der zu verwendende Hash-Algorithmus, wird ebenfalls über das vorhergehende MSE SET-Kommando ausgewählt. Siehe MSE SET-Variante 1.2 in Kap. 4.6.1. Im Fall einer positiven Signaturprüfung wird der Public Key aus dem Zertifikat (mit seinen

Key-Zusatzinformationen, also Key-Curve) im vorgesehenen Public Key-Objekt gespeichert.

- Bei erfolgreichem Import des Public Key wird das Key-Attribut Key-LifeCycleStatus des Public Key-Objektes defaultmäßig auf „operational state – activated“ gesetzt. Siehe auch Kap. 3.2.4.2.3.
- Krypto-Algorithmen: ECDSA-Signaturprüfung (mit Elliptischen Kurven und Schlüssellängen wie in [TR-03109-3] vorgegeben sowie mit Hashing im Sicherheitsmodul), Variante „Signatur ohne Message Recovery“.

3.4.6.6 Kommando GENERAL AUTHENTICATE

Kommando **GENERAL AUTHENTICATE** ([ISO 7816-4], Kap. 11.5.5, Anhang C.2 sowie insbesondere C.2.2)

a) ECKA:

- Realisierung der Kernroutinen für Key Agreement (Varianten ECKA-DH und ECKA-EG, jeweils mit Ausgabe des Shared Secret Value Z_{AB}, siehe [TR-03111], Kap. 4.3, insbesondere Kap. 4.3.1 Note).
- Hinweis: Die Key Derivation-Routine für die Ableitung der Session Keys aus dem Shared Secret Value ist nicht Bestandteil des im Sicherheitsmodul realisierten Key Agreement-Protokolls ECKA. Die Key Derivation-Routine wird im GW selbst implementiert. Grund hierfür ist die von Seiten des GW gewünschte Flexibilität: Es soll die Möglichkeit bestehen, zukünftig die Key Derivation-Mechanismen zu ändern, wobei zu berücksichtigen ist, dass ein Firmware-Update der GW-Software möglich ist, das Sicherheitsmodul jedoch diesbzgl. nicht notwendig migrierfähig ist.
- Hinweis: ECKA-DH (Anonymous Diffie-Hellman) wird im Rahmen des TLS-Handshakes verwendet, wobei die Funktionalität des Sicherheitsmoduls derart zu gestalten ist, dass für das SMGW Unterstützung sowohl für die Rolle des SMGW als TLS-Client als auch als TLS-Server bereitgestellt wird. ECKA-EG (ElGamal-Variante mit statischem Public Key) wird im Rahmen der Inhaltsdatenverschlüsselung eingesetzt.
- Vor dem Kommando-Aufruf ist ein entsprechendes MSE-Kommando (SET-Variante mit AT-Template) zur Auswahl der Protokoll-Variante abzusetzen. In diesem MSE-Kommando werden Protokoll-relevante Informationen übergeben, genauer die Protokoll-Variante referenziert und Schlüsselreferenzen gesetzt.

b) PACE:

- Realisierung des PACE-Protokolls (siehe [TR-03110-1], [TR-03110-2], [TR-03110-3]).
- Bei erfolgreicher Kommando-Ausführung wird ein entsprechender Sicherheitszustand im Sicherheitsmodul gesetzt („PACE“, siehe Kap. 3.3.1), der für die nachfolgenden Kommandos ausgewertet wird und den Zugriff auf Kommandos / Datenobjekte im Sicherheitsmodul reglementiert.
- Vor dem Kommando-Aufruf ist ein entsprechendes MSE-Kommando (SET-Variante mit AT-Template) zur Auswahl der PACE-Protokollparameter abzusetzen. In diesem MSE-Kommando werden weitere Protokoll-relevante Informationen übergeben sowie die PIN Reference gesetzt.

3.4.6.7 Kommando EXTERNAL AUTHENTICATE

Kommando **EXTERNAL AUTHENTICATE** ([ISO 7816-4], Kap. 11.5.4)

- Verwendung für die Authentisierung des GW-Administrators gegenüber dem Sicherheitsmodul.
- Bei erfolgreicher Kommando-Ausführung wird ein entsprechender Sicherheitszustand im Sicherheitsmodul gesetzt („AUTH“, siehe Kap. 3.3.1), der für die nachfolgenden Kommandos ausgewertet wird und den Zugriff auf Kommandos bzw. Objekte im Sicherheitsmodul reglementiert.
- Das Kommando setzt ein vorhergehendes GET CHALLENGE-Kommando voraus (Kommando-Variante mit Sicherheitsmodul-interner Verfügbarkeit der generierten Challenge, siehe Kap. 3.4.8.1).
- In den Kommandodaten wird das zu prüfende Authentisierungstoken von der das Kommando aufrufenden Stelle übergeben.
- Der zu verwendende Public Key muss vor Aufruf des Kommandos EXTERNAL AUTHENTICATE bereits im Sicherheitsmodul vorhanden sein (z.B. über einen Import mittels des Kommandos PSO VERIFY CERTIFICATE) und wird über ein dem Kommando EXTERNAL AUTHENTICATE vorhergehendes MSE-Kommando (SET-Variante mit AT-Template, Übergabe der Key Reference des zugehörigen Public Key-Objektes) gesetzt. Der zu verwendende Krypto-Algorithmus, insbesondere der zu verwendende Hash-Algorithmus, wird ebenfalls über das vorhergehende MSE SET-Kommando ausgewählt. Siehe MSE SET-Variante 2.3 in Kap. 4.6.1.

  Hinweis: Eine Übergabe des zu verwendenden Public Key im Kommando EXTERNAL AUTHENTICATE ist nach [ISO 7816-4] nicht vorgesehen.
- Krypto-Algorithmen: ECDSA-Signaturprüfung (mit Elliptischen Kurven und Schlüssellängen wie in [TR-03109-3] vorgegeben sowie mit Hashing im Sicherheitsmodul), Variante „Signatur ohne Message Recovery“.

3.4.6.8 Kommando INTERNAL AUTHENTICATE

Kommando **INTERNAL AUTHENTICATE** ([ISO 7816-4], Kap. 11.5.2)

- Verwendung für die interne Authentisierung. Genauer wird das Kommando im Smart Meter-System im Rahmen der Erstellung von Zertifikatsrequests zu ENC-Keys verwendet (Erstellung einer Signatur über ein Token mittels eines Private Key, der der Anwendungsklasse AT angehört).
- In den Kommandodaten wird das zu signierende Token von der das Kommando aufrufenden Stelle an das Sicherheitsmodul übergeben. Die Antwortdaten beinhalten die Signatur über das übergebene Token.

  (Hinweis: Das Hashing der Daten findet im GW statt.)
- Der zu verwendende Private Key ist zuvor im Sicherheitsmodul zu generieren (über das Kommando GENERATE ASYMMETRIC KEY PAIR) und wird über ein dem Kommando INTERNAL AUTHENTICATE vorhergehendes MSE-Kommando (SET-Variante mit AT-Template, Übergabe der Key Reference des zugehörigen Key Pair-Objektes) gesetzt. Der

zu verwendende Krypto-Algorithmus wird ebenfalls über das vorhergehende MSE SET-Kommando ausgewählt. Siehe MSE SET-Variante 2.4 in Kap. 4.6.1.

- Krypto-Algorithmen: ECDSA-Signaturerzeugung (mit Elliptischen Kurven und Schlüssellängen wie in [TR-03109-3] vorgegeben sowie ohne Hashing im Sicherheitsmodul), Variante „Signatur ohne Message Recovery“.

3.4.7 Kommandos zum Security Environment

Im folgenden werden die vom Sicherheitsmodul angebotenen Kommandos, die der Vorbereitung von Krypto-Kommandos bzw. dem Setzen des Security Environments (SE) für das folgende Krypto-Kommando dienen, beschrieben.

3.4.7.1 Kommando MSE SET

Kommando **MSE SET** ([ISO 7816-4], Kap. 11.5.11)

- Verwendung zum Setzen von Parametern des SE (Cryptographic Reference Template) wie z.B. den relevanten Informationen zu Key, Elliptischer Kurve, Krypto-Algorithmus.
- Das Kommando wird vor Aufruf der folgenden Kommandos benötigt:
  - PSO COMPUTE DIGITAL SIGNATURE
  - GENERAL AUTHENTICATE
  - PSO VERIFY DIGITAL SIGNATURE (sofern für die Signaturprüfung der Public Key nicht im Kommando übergeben wird)
  - PSO VERIFY CERTIFICATE
  - EXTERNAL AUTHENTICATE
  - INTERNAL AUTHENTICATE
- Varianten (im MSE-Kommando in den Parametern anzugeben):
  - DST-Template (digital signature) für Signaturerzeugung, Signaturverifikation und Verifikation von Zertifikaten
  - AT-Template (authentication) für Key Agreement (ECKA-DH, ECKA-EG), gegenseitige Authentisierung (PACE), externe Authentisierung und interne Authentisierung
- Im MSE-Kommando zu übergeben und damit im SE für das folgende Krypto-Kommando zu setzen: verschiedene Parameter, je nach Verwendungszweck bzw. nachfolgendem Krypto-Kommando.
- Sofern im nachfolgenden Krypto-Kommando keine Konsistenzprüfung der Key-Zusatzinformationen des betreffenden Key-Objektes gegen die entsprechenden Informationen im SE, das zuvor über das MSE SET-Kommando gesetzt wurde, erfolgt, ist diese Konsistenzprüfung im MSE SET-Kommando durchzuführen (Hersteller-abhängige Implementierung des MSE SET- bzw. Krypto-Kommandos). Siehe auch Kap. 3.4.6.1.

3.4.7.2 Kommando MSE RESTORE

Kommando **MSE RESTORE** ([ISO 7816-4], Kap. 11.5.11)

- Das Kommando wird zur Auswahl des SE bzw. zum Ändern der aktuell ausgewählten SEID benötigt.

3.4.8 Kommandos für die Generierung von Zufallszahlen

Im folgenden werden die vom Sicherheitsmodul angebotenen Kommandos für die Generierung von Zufallszahlen beschrieben.

3.4.8.1 Kommando GET CHALLENGE

Kommando **GET CHALLENGE** ([ISO 7816-4], Kap. 11.5.3)

- Verwendung für Zufallszahlen, die das GW benötigt bzw. ggf. auch Sicherheitsmodul-intern für weitere Operationen (hier: Kommando EXTERNAL AUTHENTICATE) zur Verfügung stehen.
- Länge der generierten Zufallszahlen: Hersteller-abhängig und in Abhängigkeit vom Ausgabeformat.
- Generierung von Zufallszahlen hoher Güte (gemäß der Anforderungen in [TR-03109-3]).

3.4.9 Kommandos für das Management von PINs

Im folgenden werden die vom Sicherheitsmodul angebotenen Kommandos für das PIN-Management beschrieben.

3.4.9.1 Generelles zum PIN-Management

Die Kommandos zum PIN-Management beinhalten jeweils:

- PIN-Suche anhand der PIN Reference
- Auswertung des PIN-LifeCycleStatus
- Abfrage, ob die vorgesehene Mindestlänge der PIN eingehalten ist
- Zugriffsregelprüfung
- Ausführung der jeweiligen PIN-Management-Operation

Die Kommandos zum PIN-Management sind nicht mit dem Setzen eines PIN-bezogenen Sicherheitszustands verbunden. Insbesondere ist für das Setzen eines PIN-bezogenen Sicherheitszustands bzgl. der GW-System-PIN eine erfolgreiche Ausführung des PACE-Protokolls (über das Kommando GENERAL AUTHENTICATE / Variante PACE) erforderlich.

3.4.9.2 Kommando CHANGE REFERENCE DATA

Kommando **CHANGE REFERENCE DATA** ([ISO 7816-4], Kap. 11.5.7)

- Verschiedene Varianten:
  - Setzen einer PIN
  - Wechseln einer PIN

3.4.10 Kommandos für das Life Cycle Management des Sicherheitsmoduls

Im folgenden werden die vom Sicherheitsmodul angebotenen Kommandos für die Außerbetriebnahme des Sicherheitsmoduls beschrieben.

3.4.10.1 Kommando TERMINATE CARD USAGE

Kommando **TERMINATE CARD USAGE** ([ISO 7816-9], Kap. 6.7)

- Verwendung für die Terminierung bzw. Außerbetriebnahme des Sicherheitsmoduls.
- Irreversibles Umsetzen des Life Cycle-Status des Sicherheitsmoduls auf „terminiert“.

3.4.11 Kommandos für das Management der Applikationsebene des Sicherheitsmoduls

Im folgenden werden die vom Sicherheitsmodul angebotenen Kommandos für das Zurücksetzen der Applikationsebene und insbesondere der Sicherheitszustände des Sicherheitsmoduls beschrieben.

3.4.11.1 Kommando MANAGE CHANNEL

Kommando **MANAGE CHANNEL** ([ISO 7816-4], Kap. 11.1.2)

- Verwendung für das vollständige Zurücksetzen der Applikationsebene des Sicherheitsmoduls.
- Das Zurücksetzen der Applikationsebene ist mit einer Re-Initialisierung des Sicherheitsmoduls mit Werten verbunden, wie sie bei einem Cold- / Warm-Reset des Sicherheitsmoduls verwendet werden. Insbesondere werden alle internen Sicherheitszustände zurückgesetzt, bestehende sichere Kanäle (wie z.B. ein PACE-Kanal) werden geschlossen, Session Keys werden gelöscht und das Masterfile (MF) ist implizit selektiert.

3.5 Secure Messaging

Die im Rahmen des PACE-Protokolls zwischen GW und Sicherheitsmodul ausgehandelten Session Keys werden nachfolgend für einen gesicherten Datentransfer zwischen GW und Sicherheitsmodul verwendet. Ausgehandelt wird ein Session Key K_{Enc} für die Verschlüsselung sowie ein Session Key K_{Mac} für die MAC-Sicherung der Kommando-Nachrichten und -Antworten.

In der Regel werden Kommando-Nachrichten und -Antworten MAC-gesichert *und* verschlüsselt. Eine Ausnahme bildet z.B. die Ausführung des PACE-Protokolls selbst, bei dem die zugehörigen Kommandos ungesichert ausgeführt werden (da die für Secure Messaging erforderlichen Session Keys erst im Rahmen des PACE-Protokolls ausgehandelt werden). Eine weitere Ausnahme gilt für das initiale Setzen der GW-System-PIN (über das Kommando CHANGE REFERENCE DATA / Variante Setzen einer PIN) im Rahmen der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“. Ebenso erfordert der lesende Zugriff auf die Technischen Datenfelder im Sicherheitsmodul keinen gesicherten Kanal. Ferner gilt für die Kommandos GET CHALLENGE und PSO VERIFY DIGITAL SIGNATURE in der Kommando-Variante „Übergabe des Public Key in der Kommando-Nachricht“, dass diese prinzipiell mit als auch ohne Secure Messaging ausgeführt werden können; die aufrufende Stelle zeigt im Kommando-Aufruf im CLA-Byte an, ob das Kommando mit oder ohne Secure Messaging ausgeführt werden soll. Für eine detaillierte Beschreibung der Zugriffsregeln und insbesondere die Erfordernis eines gesicherten Kanals zwischen GW und Sicherheitsmodul in den verschiedenen Phasen des Lebenszyklus-Modells siehe Kap. 3.3.

Hinweis zur MAC-Sicherung: In die MAC-Sicherung der Kommando-Antwort gehen die Status-Bytes des Antwort-Trailers ein. So kann das GW verlässlich feststellen, ob das jeweilige Kommando erfolgreich ausgeführt wurde, was z.B. insbesondere für die Kommandos zur Authentisierung, zur Verifikation von Signaturen, zum Management des Life Cycle-Status des Sicherheitsmoduls und zum Management von Key-/PIN-Objekten und DFs/EFs von Interesse ist.

Hinweis zur Verschlüsselung: Es wird grundsätzlich die Verschlüsselung der Kommandos gefordert. Vorteil dieses Konzeptes ist, dass auf Seiten des GW nicht im Einzelfall je nach übertragenen Daten geprüft und entschieden werden muss, ob eine Vertraulichkeitsanforderung besteht und somit Secure Messaging mit Verschlüsselung erforderlich ist oder nicht.

Secure Messaging erfolgt auf der Basis von AES, genauer: für Verschlüsselung AES in CBC-Mode, für MAC-Sicherung AES in CMAC-Mode, mit Send Sequence Counter nach den Vorgaben von [ISO 7816-4], [TR-03110-3], Appendix F bzw. [EN 14890-1], Kap. 9.

In [TR-03110-3], Appendix F sind insbesondere Vorgaben zum Verhalten des Betriebssystems bei Secure Messaging-Fehlern sowie beim Senden ungesicherter Kommandos an das Sicherheitsmodul bei zwischen GW und Sicherheitsmodul bestehendem PACE-Kanal enthalten, die für das Sicherheitsmodul zu berücksichtigen sind.

Hinweis: [TR-03110-3], Appendix F betrachtet explizit Secure Messaging nur für Kommandos mit geradem INS-Byte. In [ISO 7816-4] wird jedoch Secure Messaging auch für Kommandos mit ungeradem INS-Byte mit denselben Strukturen und DOs wie für Kommandos mit geradem INS-Byte spezifiziert, so dass die Ausführungen in [TR-03110-3], Appendix F auf Kommandos mit ungeradem INS-Byte direkt übertragbar sind. Betroffen ist in vorliegender Spezifikation des Sicherheitsmoduls insbesondere das Kommando GENERATE ASYMMETRIC KEY PAIR.

Es besteht folgende Nutzungsbegrenzung der AES-Session Keys für Secure Messaging:

K_{Enc}: keine Nutzungsbeschränkung für den Key, sofern das konkret vorliegende Sicherheitsmodul nicht eine solche Nutzungsbeschränkung vorgibt.

K_{Mac}: maximal 2^{32} Nutzungen des Keys (aufgrund der CMAC-Länge von 8 Byte), sofern das konkret vorliegende Sicherheitsmodul nicht eine niedrigere Maximalzahl an Nutzungen vorgibt.

Die Einhaltung der Nutzungsgrenze der AES-Session Keys für Secure Messaging liegt in der Verantwortung des GW. Optional kann das Sicherheitsmodul selbst einen Nutzungszähler mitführen und auswerten.

3.6 Weitere Funktionalitäten des Sicherheitsmoduls

3.6.1 Life Cycle-Status des Sicherheitsmoduls

Das Sicherheitsmodul bildet sein Lebenszyklus-Modell über einen geeigneten Life Cycle-Status ab. Das Sicherheitsmodul stellt geeignete Kommandos zum Weiterschalten seines Life Cycle-Status bereit.

Das Sicherheitsmodul verwendet mindestens folgende Werte für seinen Life Cycle-Status:

„nicht-initialisiert“ → „initialisiert“ → … → „terminiert“

Hierbei schließt der Status „initialisiert“ die Initialisierungsphase des Sicherheitsmoduls ab, und insbesondere ist nachfolgend das Sicherheitsmodul mit seiner SMGW-Applikation im Smart Meter-System nutzbar. Der Übergang „nicht-initialisiert“ → „initialisiert“ wird Hersteller-spezifisch realisiert, und es erfolgen keine technischen Vorgaben von Seiten der vorliegenden TR.

Hersteller-spezifisch können über die Werte „nicht-initialisiert“ und „initialisiert“ hinaus weitere Werte für den Life Cycle-Status des Sicherheitsmoduls und entsprechende Kommandos zum Weiterschalten des Life Cycle-Status im Sicherheitsmodul implementiert werden.

Der Übergang des Sicherheitsmoduls in den Status „terminiert“ wird über das Kommando TERMINATE CARD USAGE realisiert. Befindet sich das Sicherheitsmodul im Status „terminiert“, so sind alle Nutzungen, wie sie für das Sicherheitsmodul beabsichtigt sind, irreversibel gesperrt. Insbesondere können weder gespeicherte Daten ausgelesen oder neu geschrieben, Schlüssel generiert noch PINs oder Schlüssel benutzt werden.

Das Sicherheitsmodul stellt ein Technisches Datenfeld bereit, in dem das GW weitere Zwischen-Stati für das Sicherheitsmodul zwischen den Stati „initialisiert“ und „terminiert“ hinterlegen und verwalten kann (z.B. „Sicherheitsmodul vor-personalisiert“, „Sicherheitsmodul installiert“, „Sicherheitsmodul personalisiert“, „Sicherheitsmodul im Normalbetrieb“ oder ähnliches). Es erfolgt aber keine Auswertung der in diesem Technischen Datenfeld hinterlegten Status-Informationen durch das Sicherheitsmodul.

Hinweis: Das GW selbst bzw. das SMGW kann ebenso einen eigenen Life Cycle-Status mitführen. Die Abstimmung des Life Cycle-Status des GW auf den Life Cycle-Status des Sicherheitsmoduls wird durch das GW durchgeführt.

3.6.2 ATR/ATS

Das Sicherheitsmodul gibt beim Hochfahren bzw. nach einem Reset ATR/ATS-Informationen wie in [ISO 7816-3] bzw. [ISO 14443-4] beschrieben aus. Hierbei gilt genauer: Erfolgt die Implementierung des Sicherheitsmoduls auf physikalisch-technischer Ebene nach [ISO 7816-3] bzw. [ISO 14443-4], so liefert das Sicherheitsmodul entsprechend auch einen ATR/ATS gemäß der Vorgaben in [ISO 7816-3] bzw. [ISO 14443-4]. Bei anderweitiger Realisierung des Sicherheitsmoduls auf physikalisch-technischer Ebene erfolgt eine Ausgabe der in [ISO 7816-3] bzw. [ISO 14443-4] genannten ATR/ATS-Informationen auf geeignete vergleichbare Art und Weise, beispielsweise durch Implementierung einer weiteren geeigneten Protokollebene zur Übertragung der ATR/ATS-Informationen.

3.6.3 Verwendung logischer Kanäle

Die Implementierung logischer Kanäle ist sehr aufwendig, insbesondere auf Seiten des Sicherheitsmoduls sowohl aus funktionaler Sicht wie auch aus Sicherheitssicht. Auch sind logische Kanäle auf Seiten des GW entsprechend zu verwalten und anzusteuern. Es besteht derzeit aus Sicht des GW bzw. des Smart Meter-Systems keine Notwendigkeit, logische Kanäle verpflichtend zu fordern.

Bietet ein Sicherheitsmodul dennoch die Funktionalität logischer Kanäle an, so ist dies als Bestandteil des Evaluierungsgegenstands (TOE) aufzunehmen und damit Gegenstand der CC-Zertifizierung des Sicherheitsmoduls.

3.6.4 Firmware-Update des Sicherheitsmoduls

Die Möglichkeit eines Firmware-Updates des Sicherheitsmodul-Betriebssystems (d.h. Update des im Rahmen der Zertifizierung des Sicherheitsmoduls abgenommenen Betriebssystems) nach abgeschlossener Initialisierung des Sicherheitsmoduls und seiner Auslieferung als initialisiertes Modul wird nicht verpflichtend für das Sicherheitsmodul gefordert, kann aber optional vom Sicherheitsmodul angeboten werden. Bietet ein Sicherheitsmodul die Funktionalität eines Firmware-Updates seines Betriebssystems an, so ist dies inkl. seiner diesbzgl. Sicherungsmechanismen als Bestandteil des Evaluierungsgegenstands (TOE) aufzunehmen und damit Gegenstand der CC-Zertifizierung des Sicherheitsmoduls.

Hinweis: Das Einspielen von in der Zertifizierung des Sicherheitsmoduls abgenommenen Patches im Rahmen der Initialisierung des Sicherheitsmoduls fällt nicht unter das zuvor genannte Firmware-Update des Betriebssystems.

3.6.5 Explizit ausgeschlossene Funktionalitäten des Sicherheitsmoduls

Derzeit sind keine Anforderungen an den expliziten Ausschluss von Funktionalitäten des Sicherheitsmoduls bekannt.

4 Feinspezifikation des Sicherheitsmoduls

Das vorliegende Kapitel beinhaltet die Detail-Spezifikation der vom Sicherheitsmodul für das GW bereitzustellenden Kommandos und bricht die in Kap. 3 beschriebenen Anforderungen auf die (technische) Kommando-Ebene herunter.

4.1 Generelles zur Spezifikation der Kommandos

4.1.1 Kommando-Spezifikation auf logischer Ebene

Die für das Sicherheitsmodul spezifizierten Kommandos orientieren sich an [ISO 7816-4], [ISO 7816-8], [ISO 7816-9], [EN 14890-1], [EN 14890-2], [TR-03117] und [TR-03110-3] verbleiben damit auf der logischen Ebene. Es erfolgen von Seiten der [TR-03109] keine Vorgaben oder Anforderungen hinsichtlich der physikalisch-technischen Realisierung der Anbindung des Sicherheitsmoduls an das GW bzw. der vom Sicherheitsmodul für das GW bereitzustellenden Schnittstellen.

4.1.2 Technisches Format der Kommando-Spezifikation

Die Spezifikation der Kommandos des Sicherheitsmoduls in der vorliegenden Spezifikation erfolgt in ihrem ungesicherten Format. Falls die Kommandos mit Secure Messaging auszuführen sind, ist das CLA-Byte im APDU-Header jeweils entsprechend abzuändern. Siehe hierzu Kap. 3.5 und 4.11.

4.1.3 Notation

' … ' in Kommando-Beschreibungen kennzeichnet Angaben im HEX-Format.

4.1.4 Codierung

Für die Codierung von Daten im Umfeld von Krypto-Funktionalität (also ECC) ist [TR-03111] heranzuziehen. Insbesondere gilt dies für die Konvertierung von Daten. Als Encoding für die Punkte Elliptischer Kurven wird das Uncompressed Encoding gemäß [TR-03111] verwendet.

4.1.5 Warnings und Fehlermeldungen

Die Spezifikation der Warnings und Fehlermeldungen für die Antwort-APDUs der Kommandos erfolgt nur in groben Warn- und Fehlerkategorien, die für das GW sinnvoll und weiter verwertbar sind. Die in den nachstehenden Kapiteln der Kommando-Spezifikation spezifizierten Warnings und Fehlermeldungen sind um geeignete Warnings und Fehlermeldungen betreffend die generelle APDU-Struktur und Secure Messaging zu ergänzen. Über die hier spezifizierten Warnings und Fehlermeldungen hinaus sind weitere Hersteller-spezifische Warn- und Fehlermeldungen zulässig.

Fehlermeldungen der Kommandos bzgl. Extended Length-Format bzw. Command Chaining:

'67 00' Extended Length-Fehler

'68 83' Command Chaining-Fehler

4.1.6 Sonstiges

Die Kommandos realisieren geeignete Rollback- bzw. Rollforward-Mechanismen.

4.2 Kartenmanagement / Management des Filesystems

4.2.1 Kommando SELECT

Kommando **SELECT** ([ISO 7816-4], Kap. 11.1.1)

Implementierungsdetails:

- ---

a) Auswahl eines EF:

Variante: Selektieren eines EF mittels FID und ohne Ausgabe der FCP

Tabelle 19: SELECT Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'A4'	Instruction Byte gemäß [ISO 7816-4]
P1	'02'	Selektieren einer Datei (d.h. eines EF) mittels File-ID
P2	'0C'	„first occurence“, keine Antwortdaten, d.h. ohne Ausgabe der FCP-Daten
Lc	'02'	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XXYY'	FID (2 Byte)
Le	---	---

Tabelle 20: SELECT Antwort APDU im Erfolgsfall

Daten	**Inhalt**	**Beschreibung**
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Selektieren des EF ohne Ausgabe der FCP-Daten
'62 83'	FileDeactivated	Selektiertes File logisch/physikalisch deaktiviert
'62 85'	FileTerminated	Selektiertes File logisch/physikalisch terminiert

Tabelle 21: SELECT Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 82'	FileNotFound	Zu selektierendes File nicht gefunden

b) Auswahl eines DF:

Variante 1: Selektieren des MF (ohne Ausgabe der FCP)

Tabelle 22: SELECT Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'A4'	Instruction Byte gemäß [ISO 7816-4]
P1	'00'	Selektieren eines MF, DF oder EF
P2	'0C'	„first occurence“, keine Antwortdaten, d.h. ohne Ausgabe der FCP-Daten
Lc	'02'	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'3F 00'	FID für MF (2 Byte)
Le	---	---

Tabelle 23: SELECT Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Selektieren des MF ohne Ausgabe der FCP-Daten
'62 83'	FileDeactivated	Selektiertes File logisch/physikalisch deaktiviert
'62 85'	FileTerminated	Selektiertes File logisch/physikalisch terminiert

Tabelle 24: SELECT Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 82'	FileNotFound	Zu selektierendes File nicht gefunden

Variante 2: Selektieren eines DF mittels FID und ohne Ausgabe der FCP

Tabelle 25: SELECT Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'A4'	Instruction Byte gemäß [ISO 7816-4]
P1	'01'	Selektieren eines Ordners (hier: DF) mittels File-ID
P2	'0C'	„first occurence“, keine Antwortdaten, d.h. ohne Ausgabe der FCP-Daten
Lc	'02'	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XXYY'	FID (2 Byte)
Le	---	---

Tabelle 26: SELECT Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Selektieren des DF ohne Ausgabe der FCP-Daten
'62 83'	FileDeactivated	Selektiertes File logisch/physikalisch deaktiviert
'62 85'	FileTerminated	Selektiertes File logisch/physikalisch terminiert

Tabelle 27: SELECT Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 82'	FileNotFound	Zu selektierendes File nicht gefunden

Variante 3: Selektieren eines DF mittels AID und ohne Ausgabe der FCP

Tabelle 28: SELECT Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'A4'	Instruction Byte gemäß [ISO 7816-4]
P1	'04'	Selektieren eines Ordners (hier: DF) mittels Application-ID
P2	'0C'	„first occurence“, keine Antwortdaten, d.h. ohne Ausgabe der FCP-Daten
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette

	Inhalt	Beschreibung
Data	'XX …YY'	AID (bis zu 16 Oktette)
Le	---	---

Tabelle 29: SELECT Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Selektieren des DF ohne Ausgabe der FCP-Daten
'62 83'	FileDeactivated	Selektiertes File logisch/physikalisch deaktiviert
'62 85'	FileTerminated	Selektiertes File logisch/physikalisch terminiert

Tabelle 30: SELECT Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 82'	FileNotFound	Zu selektierendes File nicht gefunden

4.2.2 Kommando CREATE FILE

Kommando **CREATE FILE** ([ISO 7816-9], Kap. 6.1)

Implementierungsdetails:

- Hersteller-spezifische Implementierung. Insbesondere wird Hersteller-spezifisch festgelegt, mit welchem LCSI ein neues DF bzw. EF über das Kommando CREATE FILE angelegt wird.
- Über das Kommando CREATE FILE neu angelegte DFs und EFs werden dem SE mit SEID = 01 zugeordnet und sind nachfolgend nur in diesem SE verfügbar.

Tabelle 31: CREATE FILE Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'E0'	Instruction Byte gemäß [ISO 7816-4]
P1	'00'	---
P2	'00'	---
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX …YY'	Oktettstring mit den Spezifikationsparametern für das

	Inhalt	Beschreibung
		anzulegende File
Le	length	Anzahl der in den Antwortdaten erwarteten Oktette

Tabelle 32: CREATE FILE Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Anlegen eines Files

Tabelle 33: CREATE FILE Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'6A 8A'	DfNameExists	AID des neuen Objektes bereits verwendet
'6A 89'	DuplicatedObject	FID des neuen Objektes bereits verwendet
'6A 84'	OutOfMemory	Zu wenig Speicherplatz für neues Objekt
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.2.3 Kommando DELETE FILE

Kommando **DELETE FILE** ([ISO 7816-9], Kap. 6.2)

Implementierungsdetails:

- Das MF ist nicht löschbar.

Tabelle 34: DELETE FILE Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'E4'	Instruction Byte gemäß [ISO 7816-4]
P1	'00'	Delete current file
P2	'00'	Delete current file
Lc	---	---
Data	---	---
Le	---	---

Tabelle 35: DELETE FILE Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Löschen des durch das vorhergehende SELECT-Kommando ausgewählten Files

Tabelle 36: DELETE FILE Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 86'	NoCurrentFile	Kein File ausgewählt
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.2.4 Kommando ACTIVATE FILE

Kommando **ACTIVATE FILE** ([ISO 7816-9], Kap. 6.4)

Implementierungsdetails:

- ---

Tabelle 37: ACTIVATE FILE Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'44'	Instruction Byte gemäß [ISO 7816-4]
P1	'00'	Activate current file
P2	'00'	Activate current file
Lc	---	---
Data	---	---
Le	---	---

Tabelle 38: ACTIVATE FILE Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Aktivieren des durch das vorhergehende

		SELECT-Kommando ausgewählten Files

Tabelle 39: ACTIVATE FILE Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 86'	NoCurrentFile	Kein File ausgewählt
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 82'	ObjectTerminated	Objekt befindet sich im Zustand „terminated“
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.2.5 Kommando DEACTIVATE FILE

Kommando **DEACTIVATE FILE** ([ISO 7816-9], Kap. 6.3)

Implementierungsdetails:

- ---

Tabelle 40: DEACTIVATE FILE Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'04'	Instruction Byte gemäß [ISO 7816-4]
P1	'00'	Deactivate current file
P2	'00'	Deactivate current file
Lc	---	---
Data	---	---
Le	---	---

Tabelle 41: DEACTIVATE FILE Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Deaktivieren des durch das vorhergehende SELECT-Kommando ausgewählten Files

Tabelle 42: DEACTIVATE FILE Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 86'	NoCurrentFile	Kein File ausgewählt
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 82'	ObjectTerminated	Objekt befindet sich im Zustand „terminated“
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.2.6 Kommando TERMINATE DF

Kommando **TERMINATE DF** ([ISO 7816-9], Kap. 6.5)

Implementierungsdetails:

- ---

Tabelle 43: TERMINATE DF Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'E6'	Instruction Byte gemäß [ISO 7816-4]
P1	'00'	Terminate current file
P2	'00'	Terminate current file
Lc	---	---
Data	---	---
Le	---	---

Tabelle 44: TERMINATE DF Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Terminieren des durch das vorhergehende SELECT-Kommando ausgewählten DF

Tabelle 45: TERMINATE DF Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 86'	NoCurrentFile	Kein File ausgewählt

Trailer	Inhalt	Beschreibung
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.2.7 Kommando TERMINATE EF

Kommando **TERMINATE EF** ([ISO 7816-9], Kap. 6.6)

Implementierungsdetails:

- ---

Tabelle 46: TERMINATE EF Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'E8'	Instruction Byte gemäß [ISO 7816-4]
P1	'00'	Terminate current file
P2	'00'	Terminate current file
Lc	---	---
Data	---	---
Le	---	---

Tabelle 47: TERMINATE EF Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Terminieren des durch das vorhergehende SELECT-Kommando ausgewählten EF

Tabelle 48: TERMINATE EF Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 86'	NoCurrentFile	Kein File ausgewählt
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.3 Kommandos für den Zugriff auf Datenfelder

Für die Anwendungsfälle im Smart Meter-System genügt bei den Lese- und Schreibkommandos im Hinblick auf die zu verarbeitenden Datengrößen jeweils ein gerades INS-Byte.

4.3.1 Kommando READ BINARY

Kommando **READ BINARY** ([ISO 7816-4], Kap. 11.2.3)

Implementierungsdetails:

- Im Hinblick auf die Größe der gespeicherten Daten ist Extended Length erforderlich.

Variante 1: ohne SFI

Tabelle 49: READ BINARY Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'B0'	Instruction Byte gemäß [ISO 7816-4], hier „sequence of data elements in response field“
P1	'XX'	(Offset-P2)/256, wobei Offset Element aus ['0000', '7FFF']
P2	'XX'	Offset mod 256
Lc	---	---
Data	---	---
Le	length	Anzahl der in den Antwortdaten erwarteten Oktette

Tabelle 50: READ BINARY Antwort APDU im Erfolgsfall

Daten	**Inhalt**	**Beschreibung**
'XX …YY'	Daten	Ausgelesene Daten
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Auslesen der Daten

Tabelle 51: READ BINARY Antwort APDU im Fehlerfall

Trailer	**Inhalt**	**Beschreibung**
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 86'	NoCurrentEF	Kein EF ausgewählt
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 81'	WrongFileType	Ausgewähltes EF nicht transparent
'6B 00'	OffsetTooBig	Parameter „Offset“ in Kommando-APDU zu groß

Variante 2: mit SFI

Tabelle 52: READ BINARY Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'B0'	Instruction Byte gemäß [ISO 7816-4], hier „sequence of data elements in response field“
P1	'XX'	'80'+SFI
P2	'XX'	Offset (im Intervall ['00', 'FF'])
Lc	---	---
Data	---	---
Le	length	Anzahl der in den Antwortdaten erwarteten Oktette

Tabelle 53: READ BINARY Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
'XX …YY'	Daten	Ausgelesene Daten
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Auslesen der Daten

Tabelle 54: READ BINARY Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 82'	FileNotFound	Per SFI adressiertes EF nicht gefunden
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 81'	WrongFileType	Ausgewähltes EF nicht transparent
'6B 00'	OffsetTooBig	Parameter „Offset“ in Kommando-APDU zu groß

4.3.2 Kommando UPDATE BINARY

Kommando **UPDATE BINARY** ([ISO 7816-4], Kap. 11.2.5)

Implementierungsdetails:

- Im Hinblick auf die Größe der zu speichernden Daten ist Extended Length erforderlich.

Variante 1: ohne SFI

Tabelle 55: UPDATE BINARY Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'D6'	Instruction Byte gemäß [ISO 7816-4], hier „sequence of data elements in command field“
P1	'XX'	(Offset-P2)/256, wobei Offset Element aus ['0000', '7FFF']
P2	'XX'	Offset mod 256
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX …YY'	Zu schreibende Daten (neue Daten)
Le	---	---

Tabelle 56: UPDATE BINARY Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Schreiben der Daten

Tabelle 57: UPDATE BINARY Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 86'	NoCurrentEF	Kein EF ausgewählt
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 81'	WrongFileType	Ausgewähltes EF nicht transparent
'6B 00'	OffsetTooBig	Parameter „Offset“ in Kommando-APDU zu groß
'6A 87'	DataTooBig	Parameter „neue Daten“ ragt über Dateiende hinaus
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

Variante 2: mit SFI

Tabelle 58: UPDATE BINARY Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'D6'	Instruction Byte gemäß [ISO 7816-4], hier „sequence of data elements in command field“
P1	'XX'	'80'+SFI

	Inhalt	Beschreibung
P2	'XX'	Offset (im Intervall ['00', 'FF'])
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX …YY'	Zu schreibende Daten (neue Daten)
Le	---	---

Tabelle 59: UPDATE BINARY Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Schreiben der Daten

Tabelle 60: UPDATE BINARY Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 82'	FileNotFound	Per SFI adressiertes EF nicht gefunden
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 81'	WrongFileType	Ausgewähltes EF nicht transparent
'6B 00'	OffsetTooBig	Parameter „Offset“ in Kommando-APDU zu groß
'6A 87'	DataTooBig	Parameter „neue Daten“ ragt über Dateiende hinaus
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.3.3 Kommando READ RECORD

Kommando **READ RECORD** ([ISO 7816-4], Kap. 11.3.3)

Implementierungsdetails:

- ---

Variante 1: ohne SFI

Tabelle 61: READ RECORD Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'B2'	Instruction Byte gemäß [ISO 7816-4]
P1	'XX'	Record number

	Inhalt	Beschreibung
P2	'04'	Read Record der Nummer wie in P1 angegeben
Lc	---	---
Data	---	---
Le	length	Anzahl der in den Antwortdaten erwarteten Oktette

Tabelle 62: READ RECORD Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
'XX …YY'	Daten	Ausgelesene Daten
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Auslesen der Daten

Tabelle 63: READ RECORD Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 86'	NoCurrentEF	Kein EF ausgewählt
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 81'	WrongFileType	Ausgewähltes EF nicht strukturiert
'6A 83'	RecordNotFound	Record-Nummer existiert nicht

Variante 2: mit SFI

Tabelle 64: READ RECORD Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'B2'	Instruction Byte gemäß [ISO 7816-4]
P1	'XX'	Record number
P2	'XX'	(SFI <<3)+'04' (Read Record der Nummer wie in P1 angegeben)
Lc	---	---
Data	---	---
Le	length	Anzahl der in den Antwortdaten erwarteten Oktette

Tabelle 65: READ RECORD Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
'XX …YY'	Daten	Ausgelesene Daten

Trailer	Inhalt	Beschreibung
'90 00'	NoError	Erfolgreiches Auslesen der Daten

Tabelle 66: READ RECORD Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 82'	FileNotFound	Per SFI adressiertes EF nicht gefunden
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 81'	WrongFileType	Ausgewähltes EF nicht strukturiert
'6A 83'	RecordNotFound	Record-Nummer existiert nicht

4.3.4 Kommando UPDATE RECORD

Kommando **UPDATE RECORD** ([ISO 7816-4], Kap. 11.3.5)

Implementierungsdetails:

- ---

Variante 1: ohne SFI

Tabelle 67: UPDATE RECORD Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'DC'	Instruction Byte gemäß [ISO 7816-4]
P1	'XX'	Record number
P2	'04'	Update Record der Nummer wie in P1 angegeben
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX …YY'	Zu schreibende Daten (neue Daten)
Le	---	---

Tabelle 68: UPDATE RECORD Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Schreiben der Daten

Tabelle 69: UPDATE RECORD Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 86'	NoCurrentEF	Kein EF ausgewählt
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 81'	WrongFileType	Ausgewähltes EF nicht strukturiert
'6A 83'	RecordNotFound	Record-Nummer existiert nicht
'6A 87'	WrongRecordLength	Parameter „neue Daten“ passt nicht zur zulässigen Record-Länge
'6A 84'	OutOfMemory	Speicherplatz des EF reicht nicht aus
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

Variante 2: mit SFI

Tabelle 70: UPDATE RECORD Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'DC'	Instruction Byte gemäß [ISO 7816-4]
P1	'XX'	Record number
P2	'XX'	(SFI <<3)+'04' (Update Record der Nummer wie in P1 angegeben)
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX …YY'	Zu schreibende Daten (neue Daten)
Le	---	---

Tabelle 71: UPDATE RECORD Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Schreiben der Daten

Tabelle 72: UPDATE RECORD Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 82'	FileNotFound	Per SFI adressiertes EF nicht gefunden
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt

Trailer	Inhalt	Beschreibung
'69 81'	WrongFileType	Ausgewähltes EF nicht strukturiert
'6A 83'	RecordNotFound	Record-Nummer existiert nicht
'6A 87'	WrongRecordLength	Parameter „neue Daten“ passt nicht zur zulässigen Record-Länge
'6A 84'	OutOfMemory	Speicherplatz des EF reicht nicht aus
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.3.5 Kommando APPEND RECORD (optional)

Kommando **APPEND RECORD** ([ISO 7816-4], Kap. 11.3.6)

Implementierungsdetails:

- Zu schreibende Record-Daten werden im Kommando-Datenfeld übergeben.
- Command Chaining ist nicht erforderlich.

Variante 1: ohne SFI

Tabelle 73: APPEND RECORD Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'E2'	Instruction Byte gemäß [ISO 7816-4]
P1	'00'	---
P2	'00'	---
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX …YY'	Zu schreibende Daten (Record-Daten)
Le	---	---

Tabelle 74: APPEND RECORD Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Hinzufügen eines Records (inklusive Schreiben der übergebenen Record-Daten)

Tabelle 75: APPEND RECORD Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 86'	NoCurrentEF	Kein EF ausgewählt
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 81'	WrongFileType	Ausgewähltes EF nicht strukturiert
'6A 84'	FullEF	EF lässt keine weiteren Records zu
'6A 87'	WrongRecordLength	Parameter „neue Daten“ passt nicht zur zulässigen Record-Länge
'6A 84'	OutOfMemory	Speicherplatz des EF reicht nicht aus

Variante 2: mit SFI

Tabelle 76: APPEND RECORD Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'E2'	Instruction Byte gemäß [ISO 7816-4]
P1	'00	---
P2	'XX'	SFI (mit SFI <<3)
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX …YY'	Zu schreibende Daten (Record-Daten)
Le	---	---

Tabelle 77: APPEND RECORD Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Hinzufügen eines Records (inklusive Schreiben der übergebenen Record-Daten)

Tabelle 78: APPEND RECORD Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 82'	FileNotFound	Per SFI adressiertes EF nicht gefunden
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 81'	WrongFileType	Ausgewähltes EF nicht strukturiert

Trailer	Inhalt	Beschreibung
'6A 84'	FullEF	EF lässt keine weiteren Records zu
'6A 87'	WrongRecordLength	Parameter „neue Daten“ passt nicht zur zulässigen Record-Länge
'6A 84'	OutOfMemory	Speicherplatz des EF reicht nicht aus

4.4 Kommandos für das Key Management

4.4.1 Kommando CREATE KEY

Kommando **CREATE KEY**

Implementierungsdetails:

- Übergabe von Informationen zum anzulegenden Key-Objekt im Kommando-Datenfeld, im Detail:
  - Key-ID / Key-Name

    Die Key-ID wird hier immer mit MSBit = 0 übergeben, da keine Steuerung der Schlüsselsuche im Kommando CREATE KEY erfolgt. Die Anlage des neuen Key-Objektes erfolgt automatisch im aktuell selektierten DF.
  - Key-Type (Key Pair-Objekt / Public Key-Objekt)
  - optional: Key-UsageCounterInit (Initialwert eines binär codierten Bedienungszählers für das Key-Objekt)
  - Key-Usage (DST / AT)
  - Key-CryptoAlg (OID für ECDSA ohne Hashing / ECDSA mit Hashing / ECKA-EG / ECKA-DH)
- Hinweis: Das Key-Attribut Key-Curve wird erst im Rahmen der Schlüsselgenerierung gefüllt.
- Das Key-Attribut Key-Storage wird implizit im Kommando mit dem Wert „persistente Speicherung“ belegt.
- Sofern das Key-Attribut Key-UsageCounterInit im Kommando übergeben wird, wird das Key-Attribut Key-UsageCounter implizit mit dem Wert des Key-Attributs Key-UsageCounterInit belegt. Falls keine Übergabe des Key-Attributs Key-UsageCounterInit in der Kommando-Nachricht erfolgt, wird das Key-Objekt ohne Bedienungszähler angelegt, so dass das Key-Objekt bzw. der zukünftig darin enthaltene Schlüssel später unbegrenzt verwendet werden kann.
- Für Public Key-Objekte erkennt das Betriebssystem die doppelte Vergabe von Key-Names im Pfad eines Public Key-Objektes vom aktuell selektierten DF bis zum MF (auch über die Anwendungsklassen von Public Key-Objekten hinweg) und lehnt eine solche doppelte Vergabe bei der Ausführung des Kommandos CREATE KEY ab.

  Für Key Pair-Objekte erkennt und lehnt das Betriebssystem i.a. die mehrfache Vergabe von

Key-IDs innerhalb desselben Ordners für im Sicherheitsmodul gespeicherte Key Pair-Objekte ab. Hersteller-spezifisch können ggf. aber auch doppelt vergebene Key-IDs für Key Pair-Objekte im selben Ordner zulässig sein, sofern sich die Key Pair-Objekte in ihrer Anwendungsklasse (AT / DST im Key-Attribut Key-Usage) unterscheiden.

- Über das Kommando CREATE KEY können nur Key-Objekte für SEID = 01 angelegt werden. Das Kommando CREATE KEY trägt für das Key-Attribut Key-SEID automatisch den im Betriebssystem für SEID = 01 vorgesehenen Wert ein.
- Für das Key-Attribut Key-CryptoAlg können im Kommando CREATE KEY mehrere Werte übergeben werden.
- Im Kommando-Datenfeld zu übergeben: CP Template (DO mit Tag '62') mit Security Parameter Template (DO mit Tag 'AD'), das wiederum ein SE Template (DO mit Tag '7B') enthält.
- Key-Attribut Key-LifeCycleStatus:
  - Ein mittels des Kommandos CREATE KEY angelegtes Key Pair-Objekt trägt defaultmäßig den Key-LifeCycleStatus „initialisation“. Mittels des Kommandos GENERATE ASYMMETRIC KEY PAIR wird nachfolgend ein solches Key Pair-Objekt mit Schlüsseldaten gefüllt und dabei der Key-LifeCycleStatus auf „operational state – activated“ umgesetzt.
  - Ein mittels des Kommandos CREATE KEY angelegtes Public Key-Objekt trägt defaultmäßig den Key-LifeCycleStatus „initialisation“. Mittels des Kommandos PSO VERIFY CERTIFICATE wird nachfolgend ein solches Public Key-Objekt mit Schlüsseldaten gefüllt und dabei der Key-LifeCycleStatus auf „operational state – activated“ umgesetzt.
  - Siehe auch Kap. 3.2.4.2.3.
- Für die mittels des Kommandos CREATE KEY angelegten Key Pair- bzw. Public Key-Objekte werden defaultmäßig einheitliche Zugriffsregeln bzgl. des Zugriffs über Key Management-Kommandos und Krypto-Kommandos gesetzt wie sie in Tabelle 17 in den Tabellenzeilen für Key Pair- bzw. Public Key-Objekte angegeben sind.
- Hinsichtlich der in Tabelle 17 bei Key-Objekten als 'nicht relevant' gekennzeichneten Werte für das Key-Attribut Key-CryptoAlg sind die beiden folgenden Implementierungsvarianten des Kommandos CREATE KEY zulässig:

  Entweder: Mit dem Kommando CREATE KEY können Key-Objekte mit beliebigen Werten für das Key-Attribut Key-CryptoAlg angelegt werden, wobei dann bei den Key-Objekten, bei denen der Wert für das Key-Attribut Key-CryptoAlg mit 'nicht relevant' gekennzeichnet ist, die Zugriffsregeln für alle Krypto-Kommandos auf NEVER stehen.

  Oder: Key-Objekte, bei denen der Wert für das Key-Attribut Key-CryptoAlg mit 'nicht relevant' gekennzeichnet ist, können mittels des Kommandos CREATE KEY nicht angelegt werden und das Kommando CREATE KEY wird abgelehnt.

Tabelle 79: CREATE KEY Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'E0'	Instruction Byte gemäß [ISO 7816-4]

	Inhalt	Beschreibung	
P1	'81'	Mode, hier Übergabe von Informationen im Datenfeld	
P2	'00'	---	
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette	
Data	'XX ...YY'	'62-L_{62}-	CP Template
		AD-L_{AD}-	SP Template
		83-L_{83}-KID/KeyName ||	Key-ID/Key-Name
		96-L_{96}-KeyUsageCounterInit ||	Key-UsageCounter Init (optional)
		A1-00 bzw. A2-00 ||	Security attribute extension A1=Private Key (d.h. Key Pair-Objekt) A2=Public Key
		7B-L_{7B}-	SE DO
		A4-L_{A4}-(80-L_{80}-CryptoID) (|| A4-L_{A4}-(80-L_{80}-CryptoID))	A4=AT-CRT ggf. zweites AT-Template
		bzw.	bzw.
		B6-L_{B6}-(80-L_{80}-CryptoID) (|| B6-L_{B6}-(80-L_{80}-CryptoID))'	B6=DST-CRT ggf. zweites DST-Template, wobei CryptoID=OID des Krypto-Algorithmus (nur Wert, ohne Tag '06')
Le	---	---	

Tabelle 80: CREATE KEY Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Anlegen eines Key-Objektes

Tabelle 81: CREATE KEY Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“

Trailer	Inhalt	Beschreibung
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'6A 89'	DuplicatedObject	Key-ID/Key-Name des neuen Key-Objektes bereits verwendet
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'6A 84'	OutOfMemory	Zu wenig Speicherplatz für neues Key-Objekt
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.4.2 Kommando DELETE KEY

Kommando **DELETE KEY**

Implementierungsdetails:

- Übergabe von Key Reference und Anwendungsklasse des zu löschenden Key-Objektes im Kommando-Datenfeld.

Tabelle 82: DELETE KEY Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'E4'	Instruction Byte gemäß [ISO 7816-4]
P1	'21'	Mode, hier 1 Byte lange Key-ID bzw. 4-8 Oktett-langer Key-Name im Datenfeld
P2	'00'	---
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX ...YY'	a) Für Public Key-Objekt: 'A4-L_{A4}-(83-L_{83}-Key Reference)' Public Key für AT 'B6-L_{B6}-(83-L_{83}-Key Reference)' Public Key für DST b) Für Key Pair-Objekt: 'A4-L_{A4}-(84-01-Key Reference)' Key Pair für AT 'B6-L_{B6}-(84-01-Key Reference)' Key Pair für DST
Le	---	---

Tabelle 83: DELETE KEY Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Löschen des Key-Objektes

Tabelle 84: DELETE KEY Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 88'	KeyNotFound	Key-Objekt nicht gefunden
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.4.3 Kommando ACTIVATE KEY

Kommando **ACTIVATE KEY**

Implementierungsdetails:

- Übergabe von Key Reference und Anwendungsklasse des zu aktivierenden Key-Objektes im Kommando-Datenfeld.

Tabelle 85: ACTIVATE KEY Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'44'	Instruction Byte gemäß [ISO 7816-4]
P1	'21'	Mode, hier 1 Byte lange Key-ID bzw. 4-8 Byte langer Key-Name im Datenfeld
P2	'00'	---
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX ...YY'	a) Für Public Key-Objekt: 'A4-L_{A4}-(83-L_{83}-Key Reference)' Public Key für AT 'B6-L_{B6}-(83-L_{83}-Key Reference)' Public Key für DST b) Für Key Pair-Objekt: 'A4-L_{A4}-(84-01-Key Reference)' Key Pair für AT 'B6-L_{B6}-(84-01-Key Reference)' Key Pair für DST
Le	---	---

Tabelle 86: ACTIVATE KEY Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**

'90 00'	NoError	Erfolgreiche Aktivierung des Key-Objektes

Tabelle 87: ACTIVATE KEY Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 88'	KeyNotFound	Key-Objekt nicht gefunden
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.4.4 Kommando DEACTIVATE KEY

Kommando **DEACTIVATE KEY**

Implementierungsdetails:

- Übergabe von Key Reference und Anwendungsklasse des zu deaktivierenden Key-Objektes im Kommando-Datenfeld.

Tabelle 88: DEACTIVATE KEY Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'04'	Instruction Byte gemäß [ISO 7816-4]
P1	'21'	Mode, hier 1 Byte lange Key-ID bzw. 4-8 Byte langer Key-Name im Datenfeld
P2	'00'	---
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX ...YY'	a) Für Public Key-Objekt: 'A4-L_{A4}-(83-L_{83}-Key Reference)' Public Key für AT 'B6-L_{B6}-(83-L_{83}-Key Reference)' Public Key für DST b) Für Key Pair-Objekt: 'A4-L_{A4}-(84-01-Key Reference)' Key Pair für AT 'B6-L_{B6}-(84-01-Key Reference)' Key Pair für DST
Le	---	---

Tabelle 89: DEACTIVATE KEY Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---

Trailer	Inhalt	Beschreibung
'90 00'	NoError	Erfolgreiche Deaktivierung des Key-Objektes

Tabelle 90: DEACTIVATE KEY Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 88'	KeyNotFound	Key-Objekt nicht gefunden
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.5 Kommandos für kryptographische Anwendungen und Protokolle

4.5.1 Kommando GENERATE ASYMMETRIC KEY PAIR

Kommando **GENERATE ASYMMETRIC KEY PAIR** ([ISO 7816-8], Kap. 5.1)

Implementierungsdetails:

- Das Kommando wird zur onboard-Generierung von ECC-Schlüsselpaaren verwendet, aber auch zum Auslesen des Public Key eines zuvor onboard generierten Key-Paares.
- Die folgenden verschiedenen Varianten des Kommandos werden unterstützt:
  - Schlüsselgenerierung ohne Ausgabe des Public Key.
  - Schlüsselgenerierung mit Ausgabe des Public Key.
  - Ausgabe eines Public Key eines zuvor über das Kommando generierten Schlüsselpaares.
- Eine Schlüsselgenerierung für ein bereits vorhandenes leeres Key Pair-Objekt (d.h. ein Erst-Befüllen eines Key Pair-Objektes) oder für ein bereits bestehendes mit einem alten Key-Paar gefülltes Key Pair-Objekt (d.h. ein Neu-Befüllen eines bereits zuvor gefüllten Key Pair-Objektes) ist bei persistenten Key Pair-Objekten nur dann möglich, wenn das Key-Attribut Key-LifeCycleStatus des Key Pair-Objektes auf „initialisation“ bzw. „operational state – deactivated“ steht.

  Für temporäre Key Pair-Objekte ist unabhängig vom Wert des Key-Attributs Key-LifeCycleStatus des Key Pair-Objektes stets eine Schlüsselgenerierung möglich.

  Siehe auch Kap. 3.2.4.2.3.
- Falls das zu befüllende Key Pair-Objekt die Key-Attribute Key-UsageCounterInit und Key-UsageCounter trägt, so setzen die beiden Kommando-Varianten mit Schlüsselgenerierung implizit den Bedienungszähler im Key-Attribut Key-UsageCounter auf den im Key-Attribut Key-UsageCounterInit für den Bedienungszähler hinterlegten Initialwert.

- Eine Ausgabe des Public Key eines zuvor über das Kommando generierten Schlüsselpaares (Kommando-Variante ohne Schlüsselgenerierung) kann für die Werte „operational state – activated“ und „operational state – deactivated“ im Key-Attribut Key-LifeCycleStatus des Key Pair-Objektes erfolgen. Siehe auch Kap. 3.2.4.2.3.
- Bei den Kommando-Varianten mit Ausgabe des Public Key:
  - Ausgabe der Schlüsseldaten (Kurvenpunkt) zusammen mit der zugehörigen Elliptischen Kurve (als OID).
  - Hinweis: Für das Smart Meter-System wird eine Ausgabe der OID der Elliptischen Kurve anstelle einer Ausgabe ihrer Domain Parameter als ausreichend erachtet. Zukünftig könnten die Ausgabedaten des Kommandos aber auch mittels Verwendung einer sog. Extended Header List parametrisiert werden.
- Für das Kommando wird ein ungerades INS-Byte ('47') spezifiziert. Hintergrund für INS = '47' ist, dass in diesem Fall eine ASN.1-Codierung im Daten-/Responsefeld des Kommandos „erzwungen“ wird (und eine solche Codierung des Daten-/Responsefeldes bei der Abarbeitung des Kommandos syntaktisch geprüft wird). Für Auswirkungen auf die Codierung der Datenobjekte bzgl. Secure Messaging siehe Kap. 3.5 und 4.11.

Variante 1: Schlüsselgenerierung ohne Ausgabe des Public Key

Implementierungsdetails:

- ---

Tabelle 91: GENERATE ASYMMETRIC KEY PAIR Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'47'	Instruction Byte gemäß [ISO 7816-4], hier „sequence of data objects in data/response field“
P1	'86'	Schlüsselgenerierung und keine Ausgabe des Public Key
P2	'00'	---
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX ...YY'	a) Für ein Key-Paar, das für Signaturzwecke verwendet werden soll (Anwendungsklasse DST): 'B6-L_{B6}-(84-01-Key Reference) || 7F49-L_{7F49}-KeyParameter' b) Für ein Key-Paar, das für Zwecke der Authentisierung verwendet werden soll (Anwendungsklasse AT): 'A4-L_{A4}-(84-01-Key Reference) || 7F49-L_{7F49}-KeyParameter' Das CRT-Template (DST-Template bzw. AT-Template) enthält die Referenz auf das Key Pair-Objekt (Tag '84'), in dem das zu generierende Key-Paar gespeichert werden soll. KeyParameter = '06-L_{06}-OID der Ellipt. Kurve'
Le	---	---

Tabelle 92: GENERATE ASYMMETRIC KEY PAIR Antwort APDU im Erfolgsfall

Daten	**Inhalt**	**Beschreibung**
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Schlüsselgenerierung (ohne Ausgabe des Public Key)

Tabelle 93: GENERATE ASYMMETRIC KEY PAIR Antwort APDU im Fehlerfall

Trailer	**Inhalt**	**Beschreibung**
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 88'	KeyNotFound	Key Pair-Objekt nicht gefunden
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 82'	KeyActivated	Key Pair-Objekt befindet sich im Zustand „operational state - activated“
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

Variante 2: Schlüsselgenerierung mit Ausgabe des Public Key

Implementierungsdetails:

- ---

Tabelle 94: GENERATE ASYMMETRIC KEY PAIR Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'47'	Instruction Byte gemäß [ISO 7816-4], hier „sequence of data objects in data/response field“
P1	'82'	Schlüsselgenerierung und Ausgabe des Public Key
P2	'00'	---
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX ...YY'	a) Für ein Key-Paar, das für Signaturzwecke verwendet werden soll (Anwendungsklasse DST): 'B6-L_{B6}-(84-01-Key Reference) || 7F49-L_{7F49}-KeyParameter' b) Für ein Key-Paar, das für Zwecke der Authentisierung verwendet werden soll (Anwendungsklasse AT): 'A4-L_{A4}-(84-01-Key Reference) || 7F49-L_{7F49}-KeyParameter' Das CRT-Template (DST-Template bzw. AT-Template) enthält die Referenz auf das Key Pair-Objekt (Tag '84'), in

	Inhalt	Beschreibung
		dem das zu generierende Key-Paar gespeichert werden soll. KeyParameter = '06-L_{06}-OID der Ellipt. Kurve' Für temporäre Key Pair-Objekte wird nur das AT-Template benötigt.
Le	length	Anzahl der in den Antwortdaten erwarteten Oktette

Tabelle 95: GENERATE ASYMMETRIC KEY PAIR Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
'XX ...YY'	'7F49-L_{7F49}- (06-L_{06}-OIDCurve || 86-L_{86}-PubKeyData)'	Generierter Public Key mit der zugehörigen Elliptischen Kurve (als OID), ausgegeben als constructed public key data object OID der Ellipt. Kurve Public Key-Daten Grundsätzlich erfolgt die Ausgabe eines Public Key in Form seiner Schlüsseldaten zusammen mit der OID der zugehörigen Elliptischen Kurve.
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Schlüsselgenerierung (mit Ausgabe des Public Key)

Tabelle 96: GENERATE ASYMMETRIC KEY PAIR Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 88'	KeyNotFound	Key Pair-Objekt nicht gefunden
'64 00'	KeyInvalid	Auszulesende Public Key-Daten fehlen
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 82'	KeyActivated	Key Pair-Objekt befindet sich im Zustand „operational state - activated“
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

Variante 3: Ausgabe des Public Key

Implementierungsdetails:

- ---

Tabelle 97: GENERATE ASYMMETRIC KEY PAIR Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'47'	Instruction Byte gemäß [ISO 7816-4], hier „sequence of data objects in data/response field“
P1	'83'	Keine Schlüsselgenerierung, aber Ausgabe des Public Key eines zuvor generierten Key-Paares
P2	'00'	---
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX …YY'	'B6-L_{B6}-CRTTemplate' wobei CRTTemplate = Template mit Referenz auf das Key Pair-Objekt (Tag '84'), das dem auszugebenden Public Key zugeordnet ist
Le	length	Anzahl der in den Antwortdaten erwarteten Oktette

Tabelle 98: GENERATE ASYMMETRIC KEY PAIR Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
'XX ...YY'	'7F49-L_{7F49}- (06-L_{06}-OIDCurve || 86-L_{86}-PubKeyData)'	Generierter Public Key mit der zugehörigen Elliptischen Kurve (als OID), ausgegeben als constructed public key data object OID der Ellipt. Kurve Public Key-Daten Grundsätzlich erfolgt die Ausgabe eines Public Key in Form seiner Schlüsseldaten zusammen mit der OID der zugehörigen Elliptischen Kurve.
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Ausgabe des Public Key

Tabelle 99: GENERATE ASYMMETRIC KEY PAIR Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 88'	KeyNotFound	Key Pair-Objekt nicht gefunden
'64 00'	KeyInvalid	Auszulesende Public Key-Daten fehlen
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 82'	KeyInitialisation	Key Pair-Objekt befindet sich im Zustand „initialisation“

Trailer	Inhalt	Beschreibung
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld

4.5.2 Kommando PSO COMPUTE DIGITAL SIGNATURE

Kommando **PSO COMPUTE DIGITAL SIGNATURE** ([ISO 7816-8], Kap. 5.4, EN 14890-1, Kap. 7.4.1)

Implementierungsdetails:

- Zu implementieren ist PSO COMPUTE DIGITAL SIGNATURE in der Variante „Signatur ohne Message Recovery“ (ohne Hashing im Sicherheitsmodul).
- Signatur-Algorithmus: ECDSA mit den Elliptischen Kurven wie in [TR-03109-3] definiert.
- Das Kommando signiert die in der Kommando-Nachricht im Datenfeld als Parameter übergebenen Daten („Data to be signed“) unter Nutzung eines Private Key.
- Der zu verwendende Private Key wird vor Aufruf des Kommandos durch ein entsprechendes MSE-Kommando (SET-Variante mit DST-Template, Übergabe der Key Reference des zugehörigen Key Pair-Objektes) ausgewählt (die zu verwendende Elliptische Kurve ist damit implizit bekannt). Der zu verwendende Krypto-Algorithmus wird ebenfalls über das vorhergehende MSE SET-Kommando ausgewählt. Siehe MSE SET-Variante 1.1 in Kap. 4.6.1.
- Parameter „Data to be signed“:
  - Der Parameter enthält die zu signierenden Daten (als Oktett-String in einer zum Signaturschlüssel bzw. zur Elliptischen Kurve passenden Länge).
  - Hierbei wird eine passende Aufbereitung des Signatur-Inputs durch die das Kommando aufrufende Stelle vorausgesetzt.

    Insbesondere wird für den Fall, dass bei der für den Signatur-Input in der Kommando-Nachricht verwendeten Hash-Funktion die Bitlänge des Outputs der Hash-Funktion größer als die Bitlänge des Basispunktes der für den Signaturschlüssel verwendeten Elliptischen Kurve ist, eine Kürzung und Codierung des Signatur-Inputs gemäß [TR-03111], Kap. 4.1.2 und 4.2 als „truncated hash value“ erwartet.

    Ungeachtet dessen steht es für die Implementierung des Kommandos frei, bei inkorrekter Aufbereitung des Signatur-Inputs durch die das Kommando aufrufende Stelle entweder die Ausführung des Kommandos mit einer entsprechenden Fehlermeldung abzubrechen oder aber selbst eine Kürzung der in der Kommando-Nachricht übergebenen Daten gemäß [TR-03111], Kap. 4.1.2 und 4.2 auf die passende Länge („truncated hash value“) durchzuführen und mit der Ausführung des Kommandos fortzufahren.

    Hinweis: Die Bitlänge des Outputs der Hash-Funktion sollte nicht kleiner als die Bitlänge des Basispunktes der verwendeten Elliptischen Kurve gewählt werden.

Tabelle 100: PSO COMPUTE DIGITAL SIGNATURE Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'2A'	Instruction Byte gemäß [ISO 7816-4], hier PSO Kommando
P1	'9E'	Operation Generierung Digitale Signatur
P2	'9A'	Format der Kommando-Daten, hier die zu signierenden Daten im „plain data“-Format (d.h. Oktett-String bel. Inhalts, aber passender Länge)
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX ...YY'	Data to be signed (zu signierende Daten als Oktett-String)
Le	length	Anzahl der in den Antwortdaten erwarteten Oktette

Tabelle 101: PSO COMPUTE DIGITAL SIGNATURE Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
'XX ...YY'	Signature	Digitale Signatur über die zur Signatur übergebenen Daten (Klartextsignatur ohne weitere TLV-Codierung)
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Generierung einer Digitalen Signatur

Tabelle 102: PSO COMPUTE DIGITAL SIGNATURE Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 85'	NoKeyReference	Key-Objekt nicht ausgewählt
'6A 88'	KeyNotFound	Key Pair-Objekt nicht gefunden
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'6A 81'	UnsupportedFunction	Schlüssel unterstützt Algorithmus nicht
'64 00'	KeyInvalid	Key-Daten nicht vorhanden
'69 82'	ObjectDeactivated	Key-Objekt befindet sich im Zustand „operational state - deactivated“

4.5.3 Kommando PSO VERIFY DIGITAL SIGNATURE

Kommando **PSO VERIFY DIGITAL SIGNATURE** ([ISO 7816-8], Kap. 5.7)

Implementierungsdetails:

- Zu implementieren ist PSO VERIFY DIGITAL SIGNATURE in der Variante „Signatur ohne Message Recovery“ (ohne Hashing im Sicherheitsmodul).

- Signatur-Algorithmus: ECDSA mit den Elliptischen Kurven wie in [TR-03109-3] definiert.
- Das Kommando überprüft die in der Kommando-Nachricht im Datenfeld als Parameter übergebene ECDSA-Signatur unter Nutzung eines Public Key.
- 2 Kommando-Varianten:
  - Variante 1: Der zu verwendende Public Key (PubKey = Kurvenpunkt), die zu verwendende Elliptische Kurve (OID = OID der Kurve), der in die Signatur-Berechnung eingegangene (ggf. aufbereitete) Hashwert (Hash = Oktett-String beliebigen Inhalts, aber passender Länge) und die zu prüfende Signatur (Signatur = 'R||S' = Oktett-String beliebigen Inhalts, aber passender Länge) werden im Datenfeld der Kommando-Nachricht in Form eines Signatur-Templates übergeben.
  - Variante 2: Der zu verwendende Public Key wird zuvor über ein MSE-Kommando (SET-Variante mit DST-Template, Übergabe der Key Reference des zugehörigen Public Key-Objektes) gesetzt (die zu verwendende Elliptische Kurve ist damit implizit bekannt). Der zu verwendende Krypto-Algorithmus wird ebenfalls über das vorhergehende MSE SET-Kommando ausgewählt. Siehe MSE SET-Variante 1.2 in Kap. 4.6.1. Der in die Signatur-Berechnung eingegangene (ggf. aufbereitete) Hashwert (Hash = Oktett-String beliebigen Inhalts, aber passender Länge) und die zu prüfende Signatur (Signatur = 'R||S' = Oktett-String beliebigen Inhalts, aber passender Länge) werden im Datenfeld der Kommando-Nachricht in Form eines Signatur-Templates übergeben.

Variante 1: mit Übergabe des Public Key

Tabelle 103: PSO VERIFY DIGITAL SIGNATURE Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'2A'	Instruction Byte gemäß [ISO 7816-4], hier PSO Kommando
P1	'00'	Operation Prüfung Digitale Signatur
P2	'A8'	Format der Kommando-Daten, hier Template einer Digitalen Signatur
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX …YY'	Signature Template gemäß [ISO 7816-8], Tab. 6: '06-L_{06}-OID || 90-L_{90}-Hash || 9C-L_{9C}-PubKey || 9E-L_{9E}-Signature' wobei OID = OID der Ellipt. Kurve, Hash = Oktett-String bel. Inhalts, aber passender Länge, PubKey = Public Key-Daten, Signature = 'R || S' (Oktett-String, R und S von gleicher Oktett-Länge) Die Codierung der Public Key-Daten im Objekt '9C' erfolgt mit Uncompressed Encoding, siehe Kap. 4.1.4.

	Inhalt	Beschreibung
Le	---	---

Tabelle 104: PSO VERIFY DIGITAL SIGNATURE Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Verifikation einer Digitalen Signatur
'63 00'	VerificationError	Signaturprüfung fehlgeschlagen

Tabelle 105: PSO VERIFY DIGITAL SIGNATURE Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 81'	UnsupportedFunction	Schlüssel unterstützt Algorithmus nicht
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt

Variante 2: ohne Übergabe des Public Key

Tabelle 106: PSO VERIFY DIGITAL SIGNATURE Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'2A'	Instruction Byte gemäß [ISO 7816-4], hier PSO Kommando
P1	'00'	Operation Prüfung Digitale Signatur
P2	'A8'	Format der Kommando-Daten, hier Template einer Digitalen Signatur
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX …YY'	Signature Template gemäß [ISO 7816-8], Tab. 6: '90-L_{90}-Hash || 9E-L_{9E}-Signature' wobei Hash = Oktett-String bel. Inhalts, aber passender Länge, Signature = 'R || S' (Oktett-String, R und S von gleicher Oktett-Länge)
Le	---	---

Tabelle 107: PSO VERIFY DIGITAL SIGNATURE Antwort APDU im Erfolgsfall

Daten	**Inhalt**	**Beschreibung**
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Verifikation einer Digitalen Signatur
'63 00'	VerificationError	Signaturprüfung fehlgeschlagen

Tabelle 108: PSO VERIFY DIGITAL SIGNATURE Antwort APDU im Fehlerfall

Trailer	**Inhalt**	**Beschreibung**
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 85'	NoKeyReference	Key-Objekt nicht ausgewählt
'6A 88'	KeyNotFound	Key-Objekt nicht gefunden
'6A 81'	UnsupportedFunction	Schlüssel unterstützt Algorithmus nicht
'64 00'	KeyInvalid	Key-Daten nicht vorhanden
'69 82'	ObjectDeactivated	Key-Objekt befindet sich im Zustand „operational state - deactivated“
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt

4.5.4 Kommando PSO VERIFY CERTIFICATE

Kommando **PSO VERIFY CERTIFICATE** ([ISO 7816-8], Kap. 5.8)

Implementierungsdetails:

- Inhalte und Struktur der Zertifikate:
  - Krypto-Algorithmen: ECDSA-Signaturen (mit Elliptischen Kurven und Schlüssellängen wie in [TR-03109-3] vorgegeben), wobei „Signatur ohne Message Recovery“ (inklusive Hashing im Sicherheitsmodul).
  - Im Zertifikat enthalten: Referenz auf das Public Key-Objekt, in dem der zu importierende Public Key abgelegt werden soll, sowie die Schlüsseldaten des Public Key inklusive der OID der zugehörigen Elliptischen Kurve.
  - Weitere Details der Zertifikatsinhalte und -struktur: Siehe Beschreibung im Kommando-Datenfeld.
- Der zu verwendende Public Key wird zuvor über ein MSE-Kommando (SET-Variante mit DST-Template, Übergabe der Key Reference des zugehörigen Public Key-Objektes) gesetzt (die zu verwendende Elliptische Kurve ist damit implizit bekannt). Der zu verwendende Krypto-Algorithmus, insbesondere der zu verwendende Hash-Algorithmus, wird ebenfalls über das genannte MSE SET-Kommando ausgewählt. Siehe MSE SET-Variante 1.2 in Kap. 4.6.1.

- Über das Kommando PSO VERIFY CERTIFICATE kann ein Public Key-Objekt nur im Zustand „initialisation“ oder „operational state – deactivated“ mit (neuen) Schlüsseldaten gefüllt werden. Ein Befüllen eines Public Key-Objektes im Zustand „operational state – activated“ ist nicht möglich. Siehe auch Kap. 3.2.4.2.3.
- Im Falle, dass bei der für das Zertifikat in der Kommando-Nachricht verwendeten Hash-Funktion die Bitlänge des Outputs der Hash-Funktion größer als die Bitlänge des Basispunktes der für den Signaturschlüssel verwendeten Elliptischen Kurve ist, wird der im Kommando PSO VERIFY CERTIFICATE berechnete Hash-Wert gemäß [TR-03111], Kap. 4.1.2 und 4.2 auf „truncated hash value“ gekürzt und codiert. Eine entsprechende Aufbereitung wird für den in der Signatur des Zertifikates steckenden Hash-Wert auf Seiten der externen Welt erwartet.

  Hinweis: Die Bitlänge des Outputs der Hash-Funktion sollte nicht kleiner als die Bitlänge des Basispunktes der für die Signatur des Zertifikats verwendeten Elliptischen Kurve gewählt werden.

Tabelle 109: PSO VERIFY CERTIFICATE Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'2A'	Instruction Byte gemäß [ISO 7816-4], hier PSO Kommando
P1	'00'	Operation Prüfung Digitale Signatur
P2	'BE'	Format der Kommando-Daten, hier Certificate
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX …YY'	Certificate: '7F4E-L_{7F4E}-CertificateBody || 5F37-L_{5F37}-Signature' wobei CertificateBody = '5F29-L_{5F29}-CPI 7F49-L_{7F49}- 06-L_{06}-OIDCurve 86-L_{86}-PubKeyWert 5F20-L_{5F20}-KeyName' zu verifizierender Body des Zertifikates, wobei CPI = Certificate Profile Identifier (hier: '71'), OIDCurve = OID der Ellipt. Kurve, PubKeyWert = Schlüsseldaten des Public Key, KeyName = Key-Name des zu füllenden Public Key-Objektes Signature = 'R || S' zu verifizierende Signatur über CertificateBody inklusive Tag und Length (entsprechend [TR-03110-3])
Le	---	---

Tabelle 110: PSO VERIFY CERTIFICATE Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Verifikation des Zertifikates und erfolgreicher Import des Public Keys
'63 00'	VerificationError	Verifikation des Zertifikats fehlgeschlagen

Tabelle 111: PSO VERIFY CERTIFICATE Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 85'	NoKeyReference	Key-Objekt nicht ausgewählt
'6A 88'	KeyNotFound	Key-Objekt nicht gefunden
'6A 81'	UnsupportedFunction	Schlüssel unterstützt Algorithmus nicht
'64 00'	KeyInvalid	Key-Daten nicht vorhanden
'69 82'	ObjectDeactivated	Key-Objekt (Prüfschlüssel) befindet sich im Zustand „operational state – deactivated“
'69 82'	ObjectActivated	Key-Objekt (zu befüllendes Objekt) befindet sich im Zustand „operational state – activated“
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt

4.5.5 Kommando GENERAL AUTHENTICATE

Kommando **GENERAL AUTHENTICATE** ([ISO 7816-4], Kap. 11.5.5, Anhang C.2 sowie insbesondere C.2.2)

Über das Kommando GENERAL AUTHENTICATE werden zum einen verschiedene Protokoll-Varianten von ECKA und zum anderen das PACE-Protokoll Sicherheitsmodul-seitig realisiert.

a) ECKA:

Für die Funktionalität des Key Agreement sind folgende Protokoll-Varianten von ECKA zu realisieren (siehe auch [TR-03111], Kap. 4.3):

Protokoll-Variante 1.1: ECKA-EG mit SMGW/Sicherheitsmodul als Recipient

- Die externe Welt benötigt den Static Public Key GW_PUB_STAT des GW (aus dem zugehörigen im GW gespeicherten Zertifikat, ggf. alternativ Auslesen von GW_PUB_STAT aus dem Sicherheitsmodul mittels GENERATE ASYMMETRIC KEY PAIR).

- Die im folgenden zu verwendende Elliptische Kurve ist über den Public Key GW_PUB_STAT bestimmt. Im Sicherheitsmodul liegt die Information zur Elliptischen Kurve bereits über das zugehörige Key Pair-Objekt (GW_PRV_STAT, GW_PUB_STAT) vor.
- Die externe Welt generiert das Ephemeral Key Pair (EXT_PRV_EPH, EXT_PUB_EPH).
- Das Sicherheitsmodul nutzt für das Key Agreement
  - Static Private Key GW_PRV_STAT des GW
  - Ephemeral Public Key EXT_PUB_EPH der externen Welt (Übergabe im Kommando an das Sicherheitsmodul)

  und berechnet hieraus Z_{AB}.

  Ausgabe des Sicherheitsmoduls insgesamt: Z_{AB}
- Hinweis: Die externe Welt nutzt für das Key Agreement
  - Static Public Key GW_PUB_STAT des GW
  - Ephemeral Private Key EXT_PRV_EPH der externen Welt

  und berechnet hieraus ihrerseits Z_{AB}.

Protokoll-Variante 1.2: ECKA-EG mit SMGW/Sicherheitsmodul als Initiator

- Die externe Welt besitzt das Static Key Pair (EXT_PRV_STAT, EXT_PUB_STAT).
- Die im folgenden zu verwendende Elliptische Kurve ist über das Key Pair-Objekt (EXT_PRV_STAT, EXT_PUB_STAT) bestimmt. Dem Sicherheitsmodul ist die zu verwendende Elliptische Kurve mitzuteilen: Bei Übergabe von EXT_PUB_STAT im Kommando wird die Elliptische Kurve ebenso im Kommando mitgeteilt; liegt EXT_PUB_STAT bereits im Sicherheitsmodul vor, wird die Information zur Elliptischen Kurve dem Public Key-Objekt entnommen.
- Das Sicherheitsmodul generiert das Ephemeral Key Pair (GW_PRV_EPH, GW_PUB_EPH) und gibt GW_PUB_EPH aus.
- Das Sicherheitsmodul nutzt für das Key Agreement
  - Ephemeral Private Key GW_PRV_EPH des GW
  - Static Public Key EXT_PUB_STAT der externen Welt (Übergabe im Kommando, sofern nicht schon im Sicherheitsmodul vorhanden)

  und berechnet hieraus Z_{AB}.

  Ausgabe des Sicherheitsmoduls insgesamt: GW_PUB_EPH, Z_{AB}
- Hinweis: Die externe Welt nutzt für das Key Agreement
  - Ephemeral Public Key GW_PUB_EPH des GW
  - Static Private Key EXT_PRV_STAT der externen Welt

  und berechnet hieraus ihrerseits Z_{AB}.

Protokoll-Variante 2.1: ECKA-DH mit SMGW/Sicherheitsmodul als Initiator oder Recipient (Unterstützung des SMGW in der Rolle des TLS-Clients)

- Die im folgenden zu verwendende Elliptische Kurve wird vom GW oder der externen Welt vorgegeben. Dem Sicherheitsmodul ist die zu verwendende Elliptische Kurve mitzuteilen.
- Die externe Welt generiert das Ephemeral Key Pair (EXT_PRV_EPH, EXT_PUB_EPH) und übergibt EXT_PUB_EPH im Rahmen der Server Key Exchange Message des TLS Handshakes an das GW.
- Das Sicherheitsmodul generiert das Ephemeral Key Pair (GW_PRV_EPH, GW_PUB_EPH) und führt die Berechnung des Shared Secret Value Z_{AB} (als Teil des TLS Key Agreements auf Seiten des GW) durch, wobei es hierfür folgendes Schlüsselmaterial verwendet:
  - Ephemeral Private Key GW_PRV_EPH des GW
  - Ephemeral Public Key EXT_PUB_EPH der externen Welt (Übergabe vom GW im Kommando an das Sicherheitsmodul)

  Das Sicherheitsmodul gibt GW_PUB_EPH und Z_{AB} an das GW zur weiteren Verwendung im TLS Handshake aus. Die Realisierung erfolgt über das Kommando GENERAL AUTHENTICATE / ECKA-DH in der Protokoll-Variante 2.1.
- Alternativ zu vorigem Punkt kann auch wie folgt verfahren werden:

  Das Sicherheitsmodul generiert das Ephemeral Key Pair (GW_PRV_EPH, GW_PUB_EPH) und gibt GW_PUB_EPH an das GW zur weiteren Verwendung im TLS Handshake aus. Die Realisierung erfolgt über das Kommando GENERATE ASYMMETRIC KEY PAIR (Variante 2).

  Das Sicherheitsmodul führt die Berechnung des Shared Secret Value Z_{AB} (als Teil des TLS Key Agreements auf Seiten des GW) durch, wobei es hierfür folgendes Schlüsselmaterial verwendet:
  - Ephemeral Private Key GW_PRV_EPH des GW
  - Ephemeral Public Key EXT_PUB_EPH der externen Welt (Übergabe vom GW im Kommando an das Sicherheitsmodul)

  Das Sicherheitsmodul gibt Z_{AB} an das GW zur weiteren Verwendung im TLS Handshake aus. Die Realisierung erfolgt über das Kommando GENERAL AUTHENTICATE / ECKA-DH in der Protokoll-Variante 2.2.
- Hinweis: Die externe Welt nutzt für das Key Agreement
  - Ephemeral Public Key GW_PUB_EPH des GW
  - Ephemeral Private Key EXT_PRV_EPH der externen Welt

  und berechnet hieraus ihrerseits Z_{AB}.

Protokoll-Variante 2.2: ECKA-DH mit SMGW/Sicherheitsmodul als Initiator oder Recipient (Unterstützung des SMGW in der Rolle des TLS-Servers)

- Die im folgenden zu verwendende Elliptische Kurve wird vom GW oder der externen Welt vorgegeben. Dem Sicherheitsmodul ist die zu verwendende Elliptische Kurve mitzuteilen.
- Das Sicherheitsmodul generiert das Ephemeral Key Pair (GW_PRV_EPH, GW_PUB_EPH) und gibt GW_PUB_EPH an das GW zur weiteren Verwendung im TLS Handshake aus. Die

Realisierung erfolgt über das Kommando GENERATE ASYMMETRIC KEY PAIR (Variante 2).

- Das GW erzeugt eine Signatur über spezifische TLS Daten (insbesondere GW_PUB_EPH) im Rahmen der Server Key Exchange Message des TLS Handshakes und nutzt für die Erzeugung dieser Signatur das Sicherheitsmodul. Die Realisierung erfolgt über das Kommando PSO COMPUTE DIGITAL SIGNATURE.
- Die externe Welt generiert das Ephemeral Key Pair (EXT_PRV_EPH, EXT_PUB_EPH) und übergibt EXT_PUB_EPH an das GW.
- Das Sicherheitsmodul führt die Berechnung des Shared Secret Value Z_{AB} (als Teil des TLS Key Agreements auf Seiten des GW) durch und nutzt hierfür folgendes Schlüsselmaterial:
  - Ephemeral Private Key GW_PRV_EPH des GW
  - Ephemeral Public Key EXT_PUB_EPH der externen Welt (Übergabe vom GW im Kommando an das Sicherheitsmodul)

  Das Sicherheitsmodul gibt Z_{AB} an das GW zur weiteren Verwendung im TLS Handshake aus. Die Realisierung erfolgt über das Kommando GENERAL AUTHENTICATE / ECKA-DH in der Protokoll-Variante 2.2.
- Hinweis: Die externe Welt nutzt für das Key Agreement
  - Ephemeral Public Key GW_PUB_EPH des GW
  - Ephemeral Private Key EXT_PRV_EPH der externen Welt

  und berechnet hieraus ihrerseits Z_{AB}.

Implementierungsdetails:

- Realisierung der Kernroutinen für Key Agreement (Varianten ECKA-DH und ECKA-EG, jeweils mit Ausgabe des Shared Secret Value Z_{AB}, siehe [TR-03111], Kap. 4.3, insbesondere Kap. 4.3.1 Note).
- Hinweis: Die Key Derivation-Routine für die Ableitung der Session Keys aus dem Shared Secret Value ist nicht Bestandteil des im Sicherheitsmodul realisierten Key Agreement-Protokolls ECKA. Die Realisierung der Key Derivation-Routine ist im GW selbst zu implementieren.
- Die relevante Protokoll-Variante (hier: ECKA-Protokoll-Variante) wird vor Aufruf des Kommandos GENERAL AUTHENTICATE durch ein entsprechendes MSE-Kommando (SET-Variante mit AT-Template) über die ihr zugehörige OID ausgewählt. Wird ein persistentes oder temporäres Key-Paar des Sicherheitsmoduls benötigt, so wird dieses ebenfalls im MSE-Kommando (über die Key Reference des zugehörigen Key Pair-Objektes) gesetzt. Ferner wird, falls erforderlich, über das MSE-Kommando auch ein Public Key der externen Welt (über die Key Reference des zugehörigen Public Key-Objektes) gesetzt. Siehe MSE SET-Variante 2.2 in Kap. 4.6.1.
- Die zu verwendende Elliptische Kurve ist je nach Protokoll-Variante implizit aus dem beteiligten Schlüsselmaterial bekannt oder wird explizit im GENERAL AUTHENTICATE-Kommando mitgeteilt.
- Über das Kommando GENERAL AUTHENTICATE wird ein aus mehreren Schritten bestehendes Protokoll abgearbeitet, wobei die Protokoll-spezifischen Datenobjekte im GENERAL AUTHENTICATE-Kommando ausgetauscht werden. Die einzelnen

nacheinander auszuführenden Protokollschritte hängen von der jeweiligen Protokoll-Variante ab, siehe hierzu auch die oben stehenden Ausführungen zu den verschiedenen Protokoll-Varianten.

Für die beiden Protokoll-Varianten ECKA-EG 1.1 und 1.2 sowie ECKA-DH 2.1 und 2.2 lassen sich die erforderlichen Protokollschritte geeignet zusammenfassen, so dass für jede Protokoll-Variante jeweils nur ein Kommando GENERAL AUTHENTICATE zur Durchführung des Protokolls benötigt wird. Im Kommando GENERAL AUTHENTICATE wird dabei je nach Protokoll-Variante ein Public Key übergeben bzw. auf einen bereits im Sicherheitsmodul vorhandenen Public Key zugegriffen, sowie prinzipiell ein Schlüsselpaar generiert und das Shared Secret Value berechnet.

Hinweis: Die für ECKA-DH in der Protokoll-Variante 2.2 bzw. für ECKA-DH in der Protokoll-Variante 2.1 für den alternativen Weg erforderliche vorhergehende Generierung des ephemeralen Schlüsselmaterials des SMGW bzw. seines Sicherheitsmoduls erfolgt mittels des Kommandos GENERATE ASYMMETRIC KEY PAIR in der Variante 2 mit Schlüsselgenerierung und Ausgabe des Public Key.

- Vom Kommando GENERAL AUTHENTICATE generiertes und/oder verwendetes ephemerales Schlüsselmaterial wird nach seiner Benutzung zum Ende der Kommandoausführung gelöscht.
- Für eine einheitliche Gestaltung der im Kommando GENERAL AUTHENTICATE übergebenen Datenobjekte bzw. der in der Response zurückgegebenen Datenobjekte wird für die vier Kommando-Varianten eine gemeinsame Struktur der Datenobjekte definiert. Die übergebenen Daten-Objekte werden (wie von [ISO 7816-4] vorgesehen) mit fortlaufenden Tags versehen, wobei je nach Kommando-Variante ggf. Datenobjekte fehlen.
- Protokoll-spezifische Datenobjekte für die verschiedenen Protokoll-Varianten: Siehe nachfolgende Kommandospezifikation.
- Command Chaining ist nicht erforderlich. Ggf. ist je nach Größe der übergebenen Datenobjekte aber eine Unterstützung des Extended Length-Formates erforderlich.

Tabelle 112: GENERAL AUTHENTICATE / ECKA Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'86'	Instruction Byte gemäß [ISO 7816-4]
P1	'00'	Keys und Protokoll implizit bekannt (über vorhergehendes MSE SET-Kommando)
P2	'00'	Keys und Protokoll implizit bekannt (über vorhergehendes MSE SET-Kommando)
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX ...YY'	'7C-L_{7C}-Protokoll-spezifische Datenobjekte', genauer: 'A0-L_{A0}- 06-L_{06}-OIDCurve || (OID der Elliptischen Kurve) 86-L_{86}-PublicKey' (Public Key-Daten) (je nach Protokoll-Variante fehlen evtl. DOs)

	Inhalt	Beschreibung
		Die Codierung der Public Key-Daten im Objekt '86' erfolgt jeweils mit Uncompressed Encoding, siehe Kap. 4.1.4. Für die verschiedenen ECKA-Protokoll-Varianten sieht dies im Detail wie folgt aus: Protokoll-Variante 1.1: ECKA-EG mit SMGW/Sicherheitsmodul als Recipient 'A0-L_{A0}- 86-L_{86}-EphPubKey' (hier: EXT_PUB_EPH) (hier: OID der Elliptischen Kurve implizit bekannt) Protokoll-Variante 1.2: ECKA-EG mit SMGW/Sicherheitsmodul als Initiator sofern Key-Übergabe im Kommando: 'A0-L_{A0}- 06-L_{06}-OIDCurve || (OID der Elliptischen Kurve) 86-L_{86}-StatPubKey' (hier: EXT_PUB_STAT) wenn EXT_PUB_STAT schon im Sicherheitsmodul vorliegt: --- (leeres '7C'-Template) Protokoll-Variante 2.1: ECKA-DH mit SMGW/Sicherheitsmodul als Initiator oder Recipient (TLS-Client) 'A0-L_{A0}- 06-L_{06}-OIDCurve || (OID der Elliptischen Kurve) 86-L_{86}-EphPubKey' (hier: EXT_PUB_EPH) Protokoll-Variante 2.2: ECKA-DH mit SMGW/Sicherheitsmodul als Initiator oder Recipient (TLS-Server) 'A0-L_{A0}- 86-L_{86}-EphPubKey' (hier: EXT_PUB_EPH) (hier: OID der Elliptischen Kurve implizit bekannt) Hinweise: 1) Protokoll-Variante 1.2 stimmt aus algorithmischer Sicht sowie aus Sicht der Datenein- und Datenausgabe mit Protokoll-Variante 2.1 überein. Eine Unterscheidung besteht aber hinsichtlich der OID für das Key Agreement-Protokoll und der Verwendung von persistentem bzw. temporärem Schlüsselmaterial auf Seiten des Sicherheitsmoduls. 2) Protokoll-Variante 2.2 stimmt aus algorithmischer Sicht

	Inhalt	Beschreibung
		sowie aus Sicht der Datenein- und Datenausgabe mit Protokoll-Variante 1.1 überein. Eine Unterscheidung besteht aber hinsichtlich der OID für das Key Agreement-Protokoll und der Verwendung von persistentem bzw. temporärem Schlüsselmaterial auf Seiten des Sicherheitsmoduls.
Le	length	Anzahl der in den Antwortdaten erwarteten Oktette

Tabelle 113: GENERAL AUTHENTICATE / ECKA Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
'XX ...YY'	'XX ...YY'	'7C-L_{7C}-Protokoll-spezifische Datenobjekte', genauer: '81-L_{81}-PublicKey' || '82-L_{82}-Z_{AB}' wobei • PublicKey (nur Schlüsseldaten) je nach Protokoll-Variante/-Schritt einen Ephemeral Public Key beinhaltet oder das DO fehlt • Z_{AB} je nach Protokoll-Variante/-Schritt das Shared Secret Value beinhaltet oder das DO fehlt Die Codierung der Public Key-Daten im Objekt '81' erfolgt jeweils mit Uncompressed Encoding, siehe Kap. 4.1.4. Für die verschiedenen ECKA-Protokoll-Varianten sieht dies im Detail wie folgt aus: Protokoll-Variante 1.1: ECKA-EG mit SMGW/Sicherheitsmodul als Recipient '82-L_{82}-Z_{AB}' Protokoll-Variante 1.2: ECKA-EG mit SMGW/Sicherheitsmodul als Initiator '81-L_{81}-Ephemeral Public Key' (hier: GW_PUB_EPH) || '82-L_{82}-Z_{AB}' Protokoll-Variante 2.1: ECKA-DH mit SMGW/Sicherheitsmodul als Initiator oder Recipient (TLS-Client) '81-L_{81}-Ephemeral Public Key' (hier: GW_PUB_EPH) || '82-L_{82}-Z_{AB}' Protokoll-Variante 2.2: ECKA-DH mit SMGW/Sicherheitsmodul als Initiator oder Recipient

		(TLS-Server) '82-L_{82}-Z_{AB}'
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Durchführung der Operation
'63 00'	ProtocolError	Protokoll (Schlüsselaushandlung) fehlgeschlagen

Tabelle 114: GENERAL AUTHENTICATE / ECKA Antwort APDU im Fehlerfall

Trailer	**Inhalt**	**Beschreibung**
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 85'	NoKeyReference	Key-Objekt nicht ausgewählt
'6A 88'	KeyNotFound	Key-Objekt nicht gefunden
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'6A 81'	UnsupportedFunction	Schlüssel unterstützt Algorithmus nicht
'64 00'	KeyInvalid	Key-Daten nicht vorhanden
'69 82'	ObjectDeactivated	Key-Objekt befindet sich im Zustand „operational state - deactivated“
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld

b) PACE:

Implementierungsdetails:

- Implementiert wird das in [TR-03110-3] spezifizierte PACE-Protokoll, siehe [TR-03110-3], A.3, genauer A.3.2 (nur Generic Mapping wie in A.3.4.1, nur AES), A.3.3.
- Das relevante Protokoll (hier: PACE-Protokoll mit seinen Protokollparametern) wird vor Aufruf des Kommandos durch ein entsprechendes MSE-Kommando (SET-Variante mit AT-Template) über die zugehörige OID ausgewählt. Ferner werden im MSE-Kommando die relevante PIN Reference und die ID der Elliptischen Kurve (gemäß [TR-03110-3], Abschnitt A.2.1.1) gesetzt.
- Über GENERAL AUTHENTICATE wird ein aus mehreren Schritten bestehendes Protokoll abgearbeitet, wobei die Protokoll-spezifischen Datenobjekte in einer Kette von GENERAL AUTHENTICATE-Kommandos ausgetauscht werden:
  - 1. Schritt: Generierung der Nonce, Schlüsselableitung aus der (gemeinsamen) PIN und Verschlüsselung der Nonce → Ausgabe: verschlüsselte Nonce
  - 2. Schritt: Mapping der Nonce, d.h. Generierung der Ephemeral Domain Parameter → Ausgabe: Mapping Data (Ephemeral Domain Parameter)
  - 3. Schritt: Perform Key Agreement, d.h. Generierung des Ephemeral Key Pairs des Sicherheitsmoduls, Berechnung von K_A, Ableitung der Session Keys K_{ENC} und K_{MAC} → Ausgabe: Ephemeral Public Key
  - 4. Schritt: Mutual Authentication, d.h. Generierung des Authentication Token des Sicherheitsmoduls und Prüfung des Authentication Token des GW → Ausgabe: Authentication Token des Sicherheitsmoduls

- Protokoll-spezifische Datenobjekte: Siehe [TR-03110-3].
- Command Chaining wird erforderlich, da mehrere Kommandos GENERAL AUTHENTICATE in Folge zu senden sind. Ggf. ist je nach Größe der übergebenen Datenobjekte eine Unterstützung des Extended Length-Formates erforderlich.

Tabelle 115: GENERAL AUTHENTICATE / PACE Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'/'10'	CLA Byte gemäß [ISO 7816-4], mit Command Chaining
INS	'86'	Instruction Byte gemäß [ISO 7816-4]
P1	'00'	Protokoll implizit bekannt (über vorhergehendes MSE SET-Kommando)
P2	'00'	Protokoll implizit bekannt (über vorhergehendes MSE SET-Kommando)
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX ...YY'	'7C-L_{7C}-Protokoll-spezifische Datenobjekte', genauer: PACE-Protokoll 1. Protokollschritt: --- 2. Protokollschritt: '81-L_{81}-Mapping Data' 3. Protokollschritt: '83-L_{83}-Ephemeral Public Key' (hier: Public Key des GW) 4. Protokollschritt: '85-L_{85}-Authentication Token' (hier: Token des GW)
Le	length	Anzahl der in den Antwortdaten erwarteten Oktette

Tabelle 116: GENERAL AUTHENTICATE / PACE Antwort APDU im Erfolgsfall

Daten	**Inhalt**	**Beschreibung**
'XX ...YY'	'XX ...YY'	'7C-L_{7C}-Protokoll-spezifische Datenobjekte', genauer: PACE-Protokoll 1. Protokollschritt: '80-L_{80}-Encrypted Nonce' (hier: Nonce des Sicherheitsmoduls) 2. Protokollschritt: '82-L_{82}-Mapping Data' 3. Protokollschritt: '84-L_{84}-Ephemeral Public Key' (hier: Public Key des Sicherheitsmoduls) 4. Protokollschritt: '86-L_{86}-Authentication Token' (hier: Token des Sicherheitsmoduls)
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Durchführung der Operation
'63 00'	ProtocolError	Protokoll (Authentisierung) fehlgeschlagen

Tabelle 117: GENERAL AUTHENTICATE / PACE Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 85'	NoPINReference	PIN-Objekt nicht ausgewählt
'6A 88'	PINNotFound	PIN-Objekt nicht gefunden
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'64 00'	PINInvalid	PIN-Daten nicht vorhanden
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.5.6 Kommando EXTERNAL AUTHENTICATE

Kommando **EXTERNAL AUTHENTICATE** ([ISO 7816-4], Kap. 11.5.4)

Implementierungsdetails:

- Das Kommando überprüft das von der das Kommando aufrufenden Stelle in der Kommando-Nachricht im Datenfeld als Parameter übergebene Authentisierungstoken (als Oktettstring beliebigen Inhalts, aber passender Länge) unter Nutzung eines Public Key (Signaturprüfung).
- Das Kommando greift auf die im vorhergehenden GET CHALLENGE-Kommando im Sicherheitsmodul generierte und dort verfügbare Zufallszahl (Challenge) zu.
- Der zu verwendende Public Key wird zuvor über ein MSE-Kommando (SET-Variante mit AT-Template, Übergabe der Key Reference des zugehörigen Public Key-Objektes) gesetzt (die zu verwendende Elliptische Kurve ist damit implizit bekannt). Der zu verwendende Krypto-Algorithmus, insbesondere der zu verwendende Hash-Algorithmus, wird ebenfalls über das genannte MSE SET-Kommando ausgewählt. Siehe MSE SET-Variante 2.3 in Kap. 4.6.1.
- Es erfolgt keine Aushandlung von Session Keys.
- Im Kommando implementierte Mechanismen und Datenstrukturen orientieren sich am Kommando EXTERNAL AUTHENTICATE bzw. an der Terminal Authentication in [TR-03110-3] (Anpassung der dortigen Spezifikation an Belange des Smart Meter-Systems). Für die Zufallszahl (Challenge) wird eine Länge von 16 Byte verlangt.
- Authentisierungstoken:

  sig := SignatureGeneration (Auth-Private Key; Hash(Challenge))

  → Übergabe des Tokens im Kommando-Datenfeld
  → Kommando berechnet hash := Hash(Challenge)
  → Kommando führt SignatureVerification (Auth-Public Key; sig, hash) durch

  Im Falle, dass bei der für das Authentisierungstoken in der Kommando-Nachricht verwendeten Hash-Funktion die Bitlänge des Outputs der Hash-Funktion größer als die Bitlänge des Basispunktes der für den Signaturschlüssel verwendeten Elliptischen Kurve ist,

wird der Wert hash gemäß [TR-03111], Kap. 4.1.2 und 4.2 auf „truncated hash value“ gekürzt und codiert. Eine entsprechende Aufbereitung wird für Hash(Challenge) in sig auf Seiten der externen Welt erwartet.

Hinweis: Die Bitlänge des Outputs der Hash-Funktion sollte nicht kleiner als die Bitlänge des Basispunktes der verwendeten Elliptischen Kurve gewählt werden.

Tabelle 118: EXTERNAL AUTHENTICATE Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'82'	Instruction Byte gemäß [ISO 7816-4]
P1	'00'	„algorithm / no information given“ (hier: Krypto-Algorithmus implizit bekannt)
P2	'00'	„reference data / no information given“ (hier: Public Key implizit bekannt)
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX …YY'	Authentisierungstoken (als Oktett-String)
Le	---	---

Tabelle 119: EXTERNAL AUTHENTICATE Antwort APDU im Erfolgsfall

Daten	**Inhalt**	**Beschreibung**
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Authentisierung
'63 00'	AuthenticationError	Authentisierung fehlgeschlagen

Tabelle 120: EXTERNAL AUTHENTICATE Antwort APDU im Fehlerfall

Trailer	**Inhalt**	**Beschreibung**
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 85'	NoKeyReference	Key-Objekt nicht ausgewählt
'6A 88'	KeyNotFound	Key-Objekt nicht gefunden
'6A 81'	UnsupportedFunction	Schlüssel unterstützt Algorithmus nicht
'64 00'	KeyInvalid	Key-Daten nicht vorhanden
'69 82'	ObjectDeactivated	Key-Objekt befindet sich im Zustand „operational state - deactivated“
'69 85'	NoRandom	Keine Zufallszahl vorhanden
'69 85'	WrongRandomLength	Falsche Länge der Zufallszahl
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

Trailer	Inhalt	Beschreibung
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt

4.5.7 Kommando INTERNAL AUTHENTICATE

Kommando **INTERNAL AUTHENTICATE** ([ISO 7816-4], Kap. 11.5.2)

Implementierungsdetails:

- Das Kommando erzeugt über das von der das Kommando aufrufenden Stelle in der Kommando-Nachricht im Datenfeld als Parameter übergebene Token (als Oktettstring beliebigen Inhalts, aber passender Länge) unter Nutzung eines Private Key eine Signatur (Signaturerzeugung) und gibt diese in den Antwortdaten aus.
- Für das in der Kommando-Nachricht als Parameter übergebene Token wird eine passende Aufbereitung durch die das Kommando aufrufende Stelle vorausgesetzt.

  Insbesondere wird für den Fall, dass bei der für das Token in der Kommando-Nachricht verwendeten Hash-Funktion die Bitlänge des Outputs der Hash-Funktion größer als die Bitlänge des Basispunktes der für den Private Key verwendeten Elliptischen Kurve ist, eine Kürzung und Codierung des Tokens gemäß [TR-03111], Kap. 4.1.2 und 4.2 als „truncated hash value“ erwartet.

  Ungeachtet dessen steht es für die Implementierung des Kommandos frei, bei inkorrekter Aufbereitung des Tokens durch die das Kommando aufrufende Stelle entweder die Ausführung des Kommandos mit einer entsprechenden Fehlermeldung abzubrechen oder aber selbst eine Kürzung der in der Kommando-Nachricht übergebenen Daten gemäß [TR-03111], Kap. 4.1.2 und 4.2 auf die passende Länge („truncated hash value“) durchzuführen und mit der Ausführung des Kommandos fortzufahren.

  Hinweis: Die Bitlänge des Outputs der Hash-Funktion sollte nicht kleiner als die Bitlänge des Basispunktes der verwendeten Elliptischen Kurve gewählt werden.
- Der zu verwendende Private Key wird zuvor über ein MSE-Kommando (SET-Variante mit AT-Template, Übergabe der Key Reference des zugehörigen Key Pair-Objektes) gesetzt (die zu verwendende Elliptische Kurve ist damit implizit bekannt). Der zu verwendende Krypto-Algorithmus wird ebenfalls über das genannte MSE SET-Kommando gesetzt. Siehe MSE SET-Variante 2.4 in Kap. 4.6.1.
- Es erfolgt keine Aushandlung von Session Keys.

Tabelle 121: INTERNAL AUTHENTICATE Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'88'	Instruction Byte gemäß [ISO 7816-4]
P1	'00'	„algorithm / no information given“ (hier: Krypto-Algorithmus implizit bekannt)
P2	'00'	„reference data / no information given“ (hier: Private Key

	Inhalt	Beschreibung
		implizit bekannt)
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX …YY'	Token (als Oktett-String)
Le	length	Anzahl der in den Antwortdaten enthaltenen Oktette

Tabelle 122: INTERNAL AUTHENTICATE Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
'XX …YY'	Signature ('R || S')	Authentisierende Daten
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Authentisierung

Tabelle 123: INTERNAL AUTHENTICATE Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 85'	NoKeyReference	Key-Objekt nicht ausgewählt
'6A 88'	KeyNotFound	Key-Objekt nicht gefunden
'6A 81'	UnsupportedFunction	Schlüssel unterstützt Algorithmus nicht
'64 00'	KeyInvalid	Key-Daten nicht vorhanden
'69 82'	ObjectDeactivated	Key-Objekt befindet sich im Zustand „operational state - deactivated“
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt

4.6 Kommandos zum Security Environment

4.6.1 Kommando MSE SET

Kommando **MSE SET** ([ISO 7816-4], Kap. 11.5.11)

Implementierungsdetails:

- Das Kommando wird zum Setzen von Parametern des SE (Cryptographic Reference Template) wie z.B. den relevanten Informationen zu Key, Kurve, Krypto-Algorithmus für ein nachfolgendes Krypto-Kommando verwendet.
- Das Kommando wird vor Aufruf der folgenden Kommandos benötigt:
  - PSO COMPUTE DIGITAL SIGNATURE
  - GENERAL AUTHENTICATE

  - PSO VERIFY DIGITAL SIGNATURE (sofern für die Signaturprüfung der Public Key nicht im Kommando übergeben wird)
  - PSO VERIFY CERTIFICATE
  - EXTERNAL AUTHENTICATE
  - INTERNAL AUTHENTICATE
- Varianten (im MSE-Kommando in Parametern anzugeben):
  - DST-Template (digital signature) für Signaturerzeugung, Signaturverifikation und Verifikation von Zertifikaten
  - AT-Template (authentication) für Key Agreement (ECKA-DH, ECKA-EG), gegenseitige Authentisierung (PACE), externe Authentisierung und interne Authentisierung
- Im MSE-Kommando zu übergeben und damit im SE zu setzen: verschiedene Parameter, je nach Verwendungszweck bzw. nachfolgendem Kommando.
- Hersteller-spezifisch kann bei der Ausführung des MSE-Kommandos nur ein Setzen der Key-Referenzen und weiteren Krypto-Informationen erfolgen, ohne dass die Schlüsselsuche oder sonstige Konsistenzprüfungen ausgeführt werden, wobei in diesem Fall diese Aktionen dann erst im Rahmen des nachfolgenden Krypto-Kommandos „nachgeholt“ werden. Alternativ können Hersteller-spezifisch aber auch die Schlüsselsuche und sonstige Konsistenzprüfungen bereits in die Abarbeitung des MSE-Kommandos integriert sein.

Variante 1.1: Kommando MSE SET mit CRT DST (digital signature) für Signaturerzeugung

Implementierungsdetails:

- Im Kommando-Datenfeld wird die Key Reference des Private Key (genauer: die Key Reference des zugehörigen Key Pair-Objektes) sowie die ID für den Krypto-Algorithmus übergeben. Für den Krypto-Algorithmus ist hierbei nur id-ecdsa-plain-signatures (siehe Tabelle 7) relevant.

Tabelle 124: MSE SET (DST) Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'22'	Instruction Byte gemäß [ISO 7816-4]
P1	'41'	Operation mode „digital signature generation“ (Setzen eines Schlüssels zur Erzeugung einer digitalen Signatur)
P2	'B6'	CRT Tag „digital signature“
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX ...YY'	'80-L_{80}-CryptoID' (nur Wert, ohne Tag '06', für den Krypto-Algorithmus) ‖ '84-01-Key Reference' (für das Key Pair-Objekt)
Le	---	---

Tabelle 125: MSE SET (DST) Antwort APDU im Erfolgsfall

Daten	**Inhalt**	**Beschreibung**
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Operation

Tabelle 126: MSE SET (DST) Antwort APDU im Fehlerfall

Trailer	**Inhalt**	**Beschreibung**
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 88'	KeyNotFound	Key-Objekt nicht gefunden
'6A 81'	UnsupportedFunction	Schlüssel unterstützt Algorithmus nicht
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 82'	ObjectDeactivated	Key-Objekt befindet sich im Zustand „operational state - deactivated“
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

Variante 1.2: Kommando MSE SET mit CRT DST (digital signature) für Signaturverifikation und Verifikation von Zertifikaten

Implementierungsdetails:

- Im Kommando-Datenfeld wird die Key Reference des Public Key (genauer: die Key Reference des zugehörigen Public Key-Objektes) sowie die ID für den Krypto-Algorithmus übergeben. Für den Krypto-Algorithmus sind hierbei nur id-ecdsa-plain-signatures, id-ecdsa-plain-SHA256, id-ecdsa-plain-SHA384 bzw. id-ecdsa-plain-SHA512 (siehe Tabelle 7) relevant.

Tabelle 127: MSE SET (DST) Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'22'	Instruction Byte gemäß [ISO 7816-4]
P1	'81'	Operation mode „digital signature verification“ (Setzen eines Schlüssels zur Verifikation einer digitalen Signatur)
P2	'B6'	CRT Tag „digital signature“
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX ...YY'	'80-L_{80}-CryptoID' (nur Wert, ohne Tag '06', für den Krypto-Algorithmus) || '83-L_{83}-Key Reference' (für das Public Key-Objekt)
Le	---	---

Tabelle 128: MSE SET (DST) Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Operation

Tabelle 129: MSE SET (DST) Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 88'	KeyNotFound	Key-Objekt nicht gefunden
'6A 81'	UnsupportedFunction	Schlüssel unterstützt Algorithmus nicht
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 82'	ObjectDeactivated	Key-Objekt befindet sich im Zustand „operational state - deactivated“
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

Variante 2.1: Kommando MSE SET mit CRT AT (authentication) für Key Agreement

Implementierungsdetails:

- Im Kommando-Datenfeld wird die OID für das Key Agreement-Protokoll und ggf. weitere Informationen (wie Key Reference des Key Pair-Objektes, Key Reference des Public Key-Objektes) übergeben.

Tabelle 130: MSE SET (AT) Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'22'	Instruction Byte gemäß [ISO 7816-4]
P1	'C1'	Operation mode „key agreement“ (Setzen eines Schlüssels für interne / externe Authentisierung)
P2	'A4'	CRT Tag „authentication“
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX ...YY'	'80-L_{80}-OID Key Agreement-Protokoll' (nur Wert ohne Tag '06') || '83-L_{83}-Key Reference' (für das Public Key-Objekt) || '84-01-Key Reference' (für das Key Pair-Objekt) (je nach Protokoll-Variante fehlen evtl. DOs) Für die verschiedenen ECKA-Protokoll-Varianten sieht dies

	Inhalt	Beschreibung
		im Detail wie folgt aus (vgl. für die Schlüsselbezeichner die Ausführungen in Kap. 4.5.5 a)): Protokoll-Variante 1.1: ECKA-EG mit SMGW/Sicherheitsmodul als Recipient '80-L_{80}-OID Key Agreement-Protokoll' ‖ '84-01-Key Reference' (hier: Key Reference von GW_PRV_STAT) Protokoll-Variante 1.2: ECKA-EG mit SMGW/Sicherheitsmodul als Initiator bei Übergabe von EXT_PUB_STAT im Kommando GENERAL AUTHENTICATE: '80-L_{80}-OID Key Agreement-Protokoll' wenn EXT_PUB_STAT schon im Sicherheitsmodul vorliegt: '80-L_{80}-OID Key Agreement-Protokoll' ‖ '83-L_{83}-Key Reference' (hier: Key Reference von EXT_PUB_STAT) Protokoll-Variante 2.1: ECKA-DH mit SMGW/Sicherheitsmodul als Initiator oder Recipient (TLS-Client) '80-L_{80}-OID Key Agreement-Protokoll' Protokoll-Variante 2.2: ECKA-DH mit SMGW/Sicherheitsmodul als Initiator oder Recipient (TLS-Server) '80-L_{80}-OID Key Agreement-Protokoll' ‖ '84-01-Key Reference' (hier: Key Reference von GW_PRV_EPH)
Le	---	---

Tabelle 131: MSE SET (AT) Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Operation

Tabelle 132: MSE SET (AT) Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 88'	KeyNotFound	Key-Objekt nicht gefunden
'6A 81'	UnsupportedFunction	Schlüssel unterstützt Algorithmus nicht
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 82'	ObjectDeactivated	Key-Objekt befindet sich im Zustand „operational state - deactivated“
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

Variante 2.2: Kommando MSE SET mit CRT AT (authentication) für PACE

Implementierungsdetails:

- Im Kommando-Datenfeld wird die OID des PACE-Protokolls, eine Referenz auf die zu verwendende PIN sowie die ID der zu verwendenden Elliptischen Kurve (gemäß [TR-03110-3], Abschnitt A.2.1.1) übergeben.

Tabelle 133: MSE SET (AT) Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'22'	Instruction Byte gemäß [ISO 7816-4]
P1	'C1'	Operation mode „mutual authentication“ (Setzen eines Schlüssels für interne / externe Authentisierung)
P2	'A4'	CRT Tag „authentication“
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX ...YY'	'80-L_{80}-OID PACE-Protokoll' (nur Wert ohne Tag '06') || '83-01-PIN Reference' für das PIN-Objekt || '84-L_{84}-ID der Ellipt. Kurve' (notwendig, da mehrere Ellipt. Kurven für das PACE-Protokoll möglich sind; ID gemäß [TR-03110-3], Abschnitt A.2.1.1)
Le	---	---

Tabelle 134: MSE SET (AT) Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Operation

Tabelle 135: MSE SET (AT) Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 88'	PasswordNotFound	PIN-Objekt nicht gefunden
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

Variante 2.3: Kommando MSE SET mit CRT AT (authentication) für externe Authentisierung

Implementierungsdetails:

- Im Kommando-Datenfeld wird die Key Reference des Public Key (genauer: die Key Reference des zugehörigen Public Key-Objektes) sowie die ID für den Krypto-Algorithmus übergeben. Für den Krypto-Algorithmus sind hierbei nur id-ecdsa-plain-SHA256, id-ecdsa-plain-SHA384 bzw. id-ecdsa-plain-SHA512 (siehe Tabelle 7) relevant.

Tabelle 136: MSE SET (AT) Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'22'	Instruction Byte gemäß [ISO 7816-4]
P1	'81'	Operation mode „external authentication“ (Setzen eines Schlüssels für externe Authentisierung)
P2	'A4'	CRT Tag „authentication“
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX ...YY'	'80-L_{80}-CryptoID' (nur Wert, ohne Tag '06', für den Krypto-Algorithmus) || '83-L_{83}-Key Reference' (für das Public Key-Objekt)
Le	---	---

Tabelle 137: MSE SET (AT) Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Operation

Tabelle 138: MSE SET (AT) Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“

Trailer	Inhalt	Beschreibung
'6A 88'	KeyNotFound	Key-Objekt nicht gefunden
'6A 81'	UnsupportedFunction	Schlüssel unterstützt Algorithmus nicht
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 82'	ObjectDeactivated	Key-Objekt befindet sich im Zustand „operational state - deactivated“
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

Variante 2.4: Kommando MSE SET mit CRT AT (authentication) für interne Authentisierung

Implementierungsdetails:

- Im Kommando-Datenfeld wird die Key Reference des Private Key (genauer: die Key Reference des zugehörigen Key Pair-Objektes) sowie die ID für den Krypto-Algorithmus übergeben. Für den Krypto-Algorithmus ist hierbei nur id-ecdsa-plain-signatures (siehe Tabelle 7) relevant.

Tabelle 139: MSE SET (AT) Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'22'	Instruction Byte gemäß [ISO 7816-4]
P1	'41'	Operation mode „internal authentication“ (Setzen eines Schlüssels für interne Authentisierung)
P2	'A4'	CRT Tag „authentication“
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX ...YY'	'80-L_{80}-CryptoID' (nur Wert, ohne Tag '06', für den Krypto-Algorithmus) || '84-01-Key Reference' (für das Key Pair-Objekt)
Le	---	---

Tabelle 140: MSE SET (AT) Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Operation

Tabelle 141: MSE SET (AT) Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 88'	KeyNotFound	Key-Objekt nicht gefunden
'6A 81'	UnsupportedFunction	Schlüssel unterstützt Algorithmus nicht
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 82'	ObjectDeactivated	Key-Objekt befindet sich im Zustand „operational state - deactivated“
'6A 80'	IncorrectParameters	Inkorrekte Parameter im Kommando-Datenfeld
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.6.2 Kommando MSE RESTORE

Kommando **MSE RESTORE** ([ISO 7816-4], Kap. 11.5.11)

Implementierungsdetails:

- Übergabe der SEID für das zu setzende SE.
- Das Betriebssystem des Sicherheitsmoduls muss nur die SEs mit SEID = 01 und SEID = 02 unterstützen, und im Kommando MSE RESTORE referenzierte SEs mit anderen SEIDs als 01 und 02 können mit einer entsprechenden Fehlermeldung abgelehnt werden. Weitere SEs als die mit SEID = 01 und SEID = 02 können aber Hersteller-spezifisch unterstützt werden.
- Defaultmäßig befindet sich das Sicherheitsmodul nach dem Hochfahren im SE mit SEID = 01, wobei dieses SE für die Phasen „Personalisierung des SMGW“ und „Normalbetrieb des SMGW“ benötigt wird. Zur Durchführung der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ wird eine Umschaltung auf das SE mit SEID = 02 erforderlich, was mittels des Kommandos MSE RESTORE realisiert wird.

  Hersteller-spezifisch kann das Kommando MSE RESTORE hinsichtlich der Umschaltung vom SE mit SEID = 01 auf das SE mit SEID = 02 mit dem Ziel einer Trennung der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ von den nachfolgenden Phasen „Personalisierung des SMGW“ und „Normalbetrieb des SMGW“ abgesichert werden. Dies kann beispielsweise wie folgt erfolgen:

  Hersteller-spezifisch wird eine technische Trennung der Phase „Vor-Personalisierung + Integration von Sicherheitsmodul und GW“ von den nachfolgenden Phasen „Personalisierung des SMGW“ und „Normalbetrieb des SMGW“ mit einem entsprechenden irreversiblen Phasenübergang realisiert.

  Im Rahmen dieses Phasenübergangs wird das Kommando MSE RESTORE irreversibel deaktiviert. Das Kommando wird nachfolgend in den Phasen „Personalisierung des SMGW“ und „Normalbetrieb des SMGW“ abgelehnt (z.B. mit Fehlermeldung '6A 86').

  Alternativ kann mit dem Phasenübergang die weitere Verfügbarkeit des SE mit SEID = 02 unterbunden werden, so dass das Kommando MSE RESTORE für ein Umschalten auf das SE mit SEID = 02 in den Phasen „Personalisierung des SMGW“ und „Normalbetrieb des

SMGW“ abgelehnt wird (z.B. mit Fehlermeldung '6A 88'). Das Einstellen des SEs mit SEID = 01 über das Kommando MSE RESTORE ist in diesen beiden Phasen aber ggf. weiterhin zugelassen.

Tabelle 142: MSE RESTORE Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'22'	Instruction Byte gemäß [ISO 7816-4]
P1	'F3'	Operation mode „Auswahl einer SEID“
P2	'XX'	SEID
Lc	---	---
Data	---	---
Le	---	---

Tabelle 143: MSE RESTORE Antwort APDU im Erfolgsfall

Daten	**Inhalt**	**Beschreibung**
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Operation

Tabelle 144: MSE RESTORE Antwort APDU im Fehlerfall

Trailer	**Inhalt**	**Beschreibung**
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“

4.7 Kommandos für die Generierung von Zufallszahlen

4.7.1 Kommando GET CHALLENGE

Kommando **GET CHALLENGE** ([ISO 7816-4], Kap. 11.5.3)

Implementierungsdetails:

- Im Kommando wird im Parameter P1 eine Information (Algorithmus-ID) mitgegeben, ob die generierte Zufallszahl nur nach extern (hier: also an das GW) ausgegeben wird oder nachfolgend auch Sicherheitsmodul-intern zur weiteren Verwendung in nachfolgenden Kommandos (hier: Kommando EXTERNAL AUTHENTICATE) zur Verfügung steht.
- Eine generierte Zufallszahl, die explizit Sicherheitsmodul-intern zur weiteren Verfügung steht, ist für das Kommando EXTERNAL AUTHENTICATE reserviert.

- Kommando-Variante mit weiterer Sicherheitsmodul-interner Verfügbarkeit der generierten Zufallszahl:

  Hinsichtlich der Verfügbarkeitsdauer der Zufallszahl gelten folgende Anforderungen:

  - Nach einer Nutzung der Zufallszahl durch ein Kommando des Sicherheitsmoduls (hier: Kommando EXTERNAL AUTHENTICATE) ist die Zufallszahl „verbraucht“ und steht nachfolgend Sicherheitsmodul-intern nicht mehr zur weiteren Verwendung zur Verfügung.
  - Ein weiterer Aufruf des Kommandos GET CHALLENGE überschreibt die zuvor mittels GET CHALLENGE generierte Zufallszahl mit der neuen Zufallszahl.
  - Nach einem Reset des Sicherheitsmoduls steht die Zufallszahl Sicherheitsmodul-intern nicht zur weiteren Verwendung zur Verfügung.
  - Nach Detektion von Angriffen auf die HW/SW des Sicherheitsmoduls steht die Zufallszahl Sicherheitsmodul-intern nicht zur weiteren Verwendung zur Verfügung.
  - Ein Aufruf der Kommandos PSO COMPUTE DIGITAL SIGNATURE, PSO VERIFY DIGITAL SIGNATURE, MSE SET, SELECT DF (mit Ausnahme SELECT MF) erhält Sicherheitsmodul-intern die Zufallszahl.

    Hinweis: Hintergrund dieser Anforderung ist, dass die mit GET CHALLENGE generierte Zufallszahl in einem nachfolgenden EXTERNAL AUTHENTICATE zur Authentisierung des GW-Administrators verwendet werden soll. Hierzu ist die Zufallszahl innerhalb des TLS-Kanals zwischen GW-Administrator und GW in einem signierten Datenpaket an den GW-Administrator zu übermitteln. Zurückkommt im TLS-Kanal zwischen GW-Administrator und GW in einem ebenfalls signierten Datenpaket das Authentisierungstoken des GW-Administrators. Zwischen dem Aufruf der Kommandos GET CHALLENGE und EXTERNAL AUTHENTICATE sind somit weitere Krypto-Kommandos zur Signaturerzeugung und -verifikation (inkl. Schlüsselsuche und -setzen) zu benutzen.
  - Ein Schließen bzw. Neuaufsetzen des PACE-Kanals zwischen GW und Sicherheitsmodul bedingt, dass die Zufallszahl Sicherheitsmodul-intern nicht mehr zur weiteren Verwendung zur Verfügung steht.

Tabelle 145: GET CHALLENGE Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'84'	Instruction Byte gemäß [ISO 7816-4], hier „sequence of data elements in response field“
P1	'00'/'01'	'00': generierte Zufallszahl wird intern gespeichert '01': generierte Zufallszahl wird intern nicht gespeichert
P2	'00'	---
Lc	---	---
Data	---	---
Le	length	Länge der in den Antwortdaten erwarteten Zufallszahl (Hersteller-abhängig und in Abhängigkeit vom

	Inhalt	Beschreibung
		Ausgabeformat)

Tabelle 146: GET CHALLENGE Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
'XX ...YY'	Challenge	Zufallszahl der im Kommando gewünschten Länge (Zufallszahl wird nach extern ausgegeben und steht ggf. auch intern im Sicherheitsmodul zur weiteren Verwendung zur Verfügung)
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Zufallszahlengenerierung

Tabelle 147: GET CHALLENGE Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'xx xx'	Other	OS-spezifische Fehlermeldung

4.8 Kommandos für das Management von PINs

4.8.1 Kommando CHANGE REFERENCE DATA

Kommando **CHANGE REFERENCE DATA** ([ISO 7816-4], Kap. 11.5.7)

Implementierungsdetails:

- Kommando-Variante „Setzen einer PIN“ (Variante 1): Das Setzen einer PIN ist nur möglich, wenn sich das betreffende PIN-Objekt im Zustand „initialisation“ befindet. Bei erfolgreichem Setzen der PIN wird der Zustand des PIN-Objektes auf „operational state - activated“ gesetzt.
- Kommando-Variante „Wechseln einer PIN“ (Variante 2): Das Wechseln einer PIN ist nur möglich, wenn sich das betreffende PIN-Objekt im Zustand „operational state - activated“ befindet. Unabhängig vom Ausgang der Kommando-Ausführung verbleibt das PIN-Objekt im Zustand „operational state – activated“.
- Die Ausführung des Kommandos CHANGE REFERENCE DATA (beide Varianten) ist nicht mit dem Setzen von Sicherheitszuständen verbunden.
- Die Ausführung des Kommandos CHANGE REFERENCE DATA in der Variante „Wechsel einer PIN“ ist nicht notwendig mit einem Zurücksetzen von Sicherheitszuständen oder gar der Applikationsebene verbunden.

  Hinweis: Aus Sicherheitssicht wird empfohlen, nach einem Wechel der PIN über das Kommando CHANGE REFERENCE DATA in der Variante „Wechsel einer PIN“ die Applikationsebene zurückzusetzen, was über das Kommando MANAGE CHANNEL

ermöglicht wird. Mittels des Kommandos MANAGE CHANNEL wird das Sicherheitsmodul re-initialisiert, und insbesondere werden alle Sicherheitszustände zurückgesetzt, bestehende sichere Kanäle geschlossen und Session Keys gelöscht. Siehe Kap. 4.10.1.

- Die Codierung der PINs im Kommando-Datenfeld erfolgt als Character String (ASCII-Codierung) wie in [TR-03110-3], Kap. D.2.1.4 definiert.

Variante 1: Setzen einer PIN

Tabelle 148: CHANGE REFERENCE DATA Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'24'	Instruction Byte gemäß [ISO 7816-4], hier „normal presentation of verification data“
P1	'01'	Kommandodatenfeld enthält die zu setzende PIN
P2	'XX'	PIN Reference (siehe Tabelle 91 in [ISO 7816-4], Kap. 11.5.1)
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX …YY'	Neue PIN
Le	---	---

Tabelle 149: CHANGE REFERENCE DATA Antwort APDU im Erfolgsfall

Daten	**Inhalt**	**Beschreibung**
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Setzen der PIN

Tabelle 150: CHANGE REFERENCE DATA Antwort APDU im Fehlerfall

Trailer	**Inhalt**	**Beschreibung**
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 88'	PasswordNotFound	PIN-Objekt nicht gefunden
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'69 82'	ObjectActivated	PIN-Objekt befindet sich im Zustand „operational state - activated“
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

Variante 2: Wechseln einer PIN

Tabelle 151: CHANGE REFERENCE DATA Kommando APDU

	Inhalt	Beschreibung
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'24'	Instruction Byte gemäß [ISO 7816-4], hier „normal presentation of verification data“
P1	'00'	Kommandodatenfeld enthält die alte und neue PIN
P2	'XX'	PIN Reference (siehe Tabelle 91 in [ISO 7816-4], Kap. 11.5.1)
Lc	length	Anzahl der in den Kommandodaten enthaltenen Oktette
Data	'XX …YY'	'Alte PIN' || 'Neue PIN'
Le	---	---

Tabelle 152: CHANGE REFERENCE DATA Antwort APDU im Erfolgsfall

Daten	Inhalt	Beschreibung
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Änderung der PIN
'63 CX'	WrongSecretWarning	Alte PIN falsch

Tabelle 153: CHANGE REFERENCE DATA Antwort APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'6A 88'	PasswordNotFound	PIN-Objekt nicht gefunden
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'6A 87'	LongPassword	Neue PIN zu lang
'6A 87'	ShortPassword	Neue PIN zu kurz
'69 82'	ObjectInitialisation	PIN-Objekt befindet sich im Zustand „initialisation“
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.9 Kommandos für das Life Cycle Management des Sicherheitsmoduls

4.9.1 Kommando TERMINATE CARD USAGE

Kommando **TERMINATE CARD USAGE** ([ISO 7816-9], Kap. 6.7)

Implementierungsdetails:

- Hinweis: Das Kommando realisiert lediglich eine Umsetzung des Life Cycle-Status des Sicherheitsmoduls auf den Wert „terminiert“. In der Regel ist das Kommando *nicht* mit einem Löschen von Speicherbereichen im Sicherheitsmodul verbunden.
- Im terminierten Zustand gibt das Sicherheitsmodul nur noch seinen ATR/ATS (siehe Kap. 3.6.2) aus. Alle an das Kommando gesandten Kommandos werden mit der Fehlermeldung '6D 00' abgewiesen.

Tabelle 154: TERMINATE CARD USAGE Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'FE'	Instruction Byte gemäß [ISO 7816-4]
P1	'00'	---
P2	'00'	---
Lc	---	---
Data	---	---
Le	---	---

Tabelle 155: TERMINATE CARD USAGE Antwort APDU im Erfolgsfall

Daten	**Inhalt**	**Beschreibung**
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiche Terminierung des SecMod

Tabelle 156: TERMINATE CARD USAGE Antwort APDU im Fehlerfall

Trailer	**Inhalt**	**Beschreibung**
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'65 81'	MemoryFailure	Schreibvorgang nicht erfolgreich

4.10 Kommandos für das Management der Applikationsebene des Sicherheitsmoduls

4.10.1 Kommando MANAGE CHANNEL

Kommando **MANAGE CHANNEL** ([ISO 7816-4], Kap. 11.1.2)

Implementierungsdetails:

- Das Kommando MANAGE CHANNEL wird in der Option „Zurücksetzen des Basiskanals und Schließen aller anderen logischen Kanäle“ verwendet. Da eine Realisierung von weiteren logischen Kanälen zusätzlich zum Basiskanal durch die vorliegende TR nicht gefordert wird, beschränkt sich die Umsetzung des Kommandos MANAGE CHANNEL auf die Funktion „Reset basic logical channel“.
- Die Ausführung des Kommandos beinhaltet eine Re-Initialisierung des Sicherheitsmoduls mit Werten, wie sie für das Sicherheitsmodul bei einem Cold- / Warm-Reset vorgesehen sind. Insbesondere werden im Sicherheitsmodul alle internen Sicherheitszustände zurückgesetzt, bestehende sichere Kanäle (wie z.B. ein PACE-Kanal) geschlossen, Session Keys gelöscht und das MF selektiert.

Tabelle 157: MANAGE CHANNEL Kommando APDU

	Inhalt	**Beschreibung**
CLA	'00'	CLA Byte gemäß [ISO 7816-4]
INS	'70'	Instruction Byte gemäß [ISO 7816-4]
P1	'40'	---
P2	'01'	---
Lc	---	---
Data	---	---
Le	---	---

Tabelle 158: MANAGE CHANNEL Antwort APDU im Erfolgsfall

Daten	**Inhalt**	**Beschreibung**
---	---	---
Trailer	**Inhalt**	**Beschreibung**
'90 00'	NoError	Erfolgreiches Zurücksetzen der Applikationsebene des SecMod

Tabelle 159: MANAGE CHANNEL Antwort APDU im Fehlerfall

Trailer	**Inhalt**	**Beschreibung**
'6D 00'	InstructionNotSupported	Sicherheitsmodul im Zustand „terminiert“

4.11 Secure Messaging

Codierung des CLA-Bytes:
CLA = 'XC', wobei Bit 8 = 0 Interindustry Class, Bit 5 = 1 Command Chaining anzeigt
(d.h. Secure Messaging mit authentisiertem Header)

Fehlermeldungen der Kommandos bzgl. Secure Messaging:
'69 87' Secure Messaging Data Object fehlt
'69 88' Secure Messaging Data Object inkorrekt

5 Sicherheitszertifizierung des Sicherheitsmoduls

Das Sicherheitsmodul unterliegt einer Sicherheitszertifizierung nach Common Criteria (CC) auf Basis des Protection Profiles [PP 0077].

Die Implementierung des Sicherheitsmoduls erfolgt auf der Basis der in der vorliegenden TR enthaltenen technischen Spezifikation für das Sicherheitsmodul. Im Rahmen der Sicherheitszertifizierung des Sicherheitsmoduls wird nachgeprüft, dass die Implementierung des Sicherheitsmoduls die Vorgaben der vorliegenden TR erfüllt.

In die Sicherheitszertifizierung des Sicherheitsmoduls fließen insbesondere die Anforderungen an das Sicherheitsmodul aus [TR-03109-3] ein.

Im Rahmen seiner Produktion bzw. Initialisierung im Sicherheitsmodul aufgebrachte zusätzliche Ordner (DFs), Datenfelder (EFs) sowie Key- und PIN-Objekte, die über die für das Initialisierungsfile in Kap. 3.1.2 vordefinierten Ordner, Datenfelder, Key- und PIN-Objekte hinausgehen, sowie alle weiteren Funktionalitäten des Betriebssystems des Sicherheitsmoduls, die über die nach der vorliegenden TR verpflichtend zu implementierenden Funktionalitäten hinausgehen (wie z.B. zusätzliche Kommandos oder Kommando-Varianten, weitere Werte für den Life Cycle-Status des Sicherheitsmoduls und die zugehörigen Kommandos zum Weiterschalten des Life Cycle-Status, usw.), liegen außerhalb der vorliegenden Spezifikation des Sicherheitsmoduls, wie dieses für seinen Einsatz im SMGW bzw. Smart Meter-System vorgesehen ist. Es muss aber sichergestellt und im Rahmen der CC-Zertifizierung des Sicherheitsmoduls geprüft werden, dass diese zusätzlichen Ordner, Datenfelder, Key- und PIN-Objekte mit ihren zugehörigen Management- und Krypto-Kommandos sowie diese zusätzlichen Funktionalitäten des Betriebssystems des Sicherheitsmoduls *nicht* die für das Smart Meter-Sicherheitsmodul und seinen Einsatz im SMGW bzw. Smart Meter-System vorgesehenen Sicherheitsstrukturen verändern, umgehen oder außer Kraft setzen. Insbesondere sind hierbei die in den Kap. 3.3.1, 3.6.3 und 3.6.4 stehenden Anforderungen bzgl. der CC-Zertifizierung des Sicherheitsmoduls zu berücksichtigen.

Die Benutzerdokumentation zu einem CC-zertifizierten Sicherheitsmodul beinhaltet insbesondere Informationen, Nutzungshinweise und Auflagen für den Nutzer des Sicherheitsmoduls bzgl. der Hersteller-spezifischen Implementierung der Schlüsselsuche im Sicherheitsmodul. Dies betrifft insbesondere den Aspekt, inwieweit deaktivierte und terminierte DFs in die Schlüsselsuche einbezogen oder in dieser Suche übersprungen werden.

Literaturverzeichnis

[TR-03109] BSI TR-03109 (Dachdokument), BSI, aktuelle Fassung

[TR-03109-1] BSI TR-03109-1 Smart Meter Gateway - Anforderungen an die Interoperabilität der Kommunikationseinheit eines intelligenten Messsystems, BSI, aktuelle Fassung

[TR-03109-1A] BSI TR-03109-1 Anhang: Betriebsprozesse, BSI, aktuelle Fassung

[TR-03109-2A] BSI TR-03109-2 Anhang: Smart Meter Gateway - Sicherheitsmodul - Use Cases, BSI, Version 1.1, 2014

[TR-03109-3] BSI TR-03109-3 Kryptographische Vorgaben, BSI, aktuelle Fassung

[TR-03109-4] BSI TR-03109-4 Public Key Infrastruktur für Smart Meter Gateways, BSI, aktuelle Fassung

[PP 0073] BSI-CC-PP-0073 „Protection Profile for the Gateway of a Smart Metering System“, BSI, aktuelle Fassung

[PP 0077] BSI-CC-PP-0077-V2 „Protection Profile for the Security Module of a Smart Meter Gateway (Security Module PP)“, BSI, Version 1.03, 2014

[ISO 7816-3] ISO/IEC 7816-3: Identification cards - Integrated circuit cards with contacts - Part 3: Electrical interface and transmission protocols, ISO/IEC, IS 2006

[ISO 7816-4] ISO/IEC 7816-4: Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange, ISO/IEC, IS 2013

[ISO 7816-8] ISO/IEC 7816-8: Identification cards - Integrated circuit cards - Part 8: Commands for security operations, ISO/IEC, IS 2004

[ISO 7816-9] ISO/IEC 7816-9: Identification cards - Integrated circuit cards - Part 9: Commands for card management, ISO/IEC, IS 2004

[ISO 14443-4] ISO/IEC 14443-4: Identification cards - Contactless integrated circuit cards - Proximity cards - Part 4: Transmission protocol, ISO/IEC, IS 2008

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

[TR-03110-1] BSI TR-03110-1 Advanced Security Mechanisms for Machine Readable Travel Documents – Part 1 – eMRTDs with BAC/PACEv2 and EACv1, BSI, Version 2.10, 2012

[TR-03110-2] BSI TR-03110-2 Advanced Security Mechanisms for Machine Readable Travel Documents – Part 2 – Extended Access Control Version 2 (EACv2), Password Authenticated Connection Establishment (PACE), and Restricted Identification (RI), BSI, Version 2.10, 2012

[TR-03110-3] BSI TR-03110-3 Advanced Security Mechanisms for Machine Readable Travel Documents - Part 3 - Common Specifications, BSI, Version 2.11, 2013

[TR-03116-3] BSI TR-03116-3 eCard-Projekte der Bundesregierung - Kryptographische Vorgaben für die Infrastruktur von Messsystemen, BSI, 2014

[TR-03117] BSI TR-03117 eCards mit kontaktloser Schnittstelle als sichere Signaturerstellungseinheit, BSI, Version 1.0, 2009

[EN 14890-1] EN 14890-1: Application Interface for smart cards used as Secure Signature Creation Devices - Part 1: Basic Services, EN, 2011

[EN 14890-2] EN 14890-2: Application Interface for smart cards used as Secure Signature Creation Devices - Part 2: Additional Services, EN, 2011

[RFC 5114] RFC 5114 - Additional Diffie-Hellman Groups for Use with IETF Standards, M. Lepinski, S. Kent, 2008

Stichwort- und Abkürzungsverzeichnis

Siehe Kap. 1.3.