VDV-Kernapplikation

Spezifikation Tag

Thema:	VDV-KA Tag
Dateiname:	SPEC_LUKA_KA-TAG_V1.0.doc
Version:	1.0
Erstellt am	11.01.2011
Zuletzt geändert am:	19.04.2011 08:59
Ersteller:	VDV-KA KG eSol GmbH

Inhaltsverzeichnis

1 Kennung des VDV KA Tag 1

2 Zustände des VDV-KA Tag-Mediums 2

2.1. Lebenszyklus des Mediums 2
2.1.1. Zustand DEVICE FACTORY 2
2.1.2. Zustand APP LOADABLE 2
2.1.3. Zustand APP SELECTABLE 3
2.1.4. Zustand DEVICE TERMINATED 3

3 Datenstrukturen 4

3.1. Root Daten 4
3.1.1. PrK.Config Daten 4
3.1.2. PuK.Root Daten 5
3.2. Tag-ID 6
3.3. Service-ID 6
3.4. Service-Verzeichnis 7
3.5. Eigentümerdaten 8
3.6. Service-Quittung 8
3.7. FCI 9

4 Prozesse 10

4.1. Konfiguration der Applikation 10
4.1.1. Sammelzustand CONFIGURATION (CFG) 11
4.2. Authentifizierung der Applikation 13
4.2.1. Authentifizierungsdaten 13
4.2.2. Ablauf der Authentifizierung 14
4.2.3. Berechnung des Kryptogramms CD.Auth 16
4.3. Eigentümer Authentisierung 16
4.4. Hinzufügen einer Service-ID 16
4.5. Löschen einer Service-ID 16
4.6. Ungesichertes Lesen einer Tag-ID und Service-ID 16
4.7. Authentifizierung einer Service-ID 17

5 Kommandos der VDV-KA Tag-Applikation 18

5.1. Festlegungen zur Kommandoausführung 18
5.1.1. Formale Prüfungen 18
5.1.2. Verketteter Modus 19
5.1.3. Secure Messaging 19
5.1.3.1. Kommandodaten 19
5.1.3.2. Antwortdaten 20
5.1.4. Returncodes 21
5.2. SELECT FILE 22
5.2.1.1. Kommando- und Antwortnachricht 22

5.3. CONFIGURE 23
5.3.1. Kommandoaufbau 23
5.3.2. Übersicht CONFIGURE Sub-Kommandos 24
5.3.3. INFO 25
5.3.3.1. Kommando- und Antwortnachricht 25
5.3.4. AUTH 26
5.3.4.1. Kommando- und Antwortnachricht 26
5.3.4.2. Ablaufbeschreibung 26
5.3.5. GENERATE_PKP 28
5.3.5.1. Kommando- und Antwortnachricht 28
5.3.5.2. Ablaufbeschreibung 28
5.3.6. GET_PUK_TAG 29
5.3.6.1. Kommando- und Antwortnachricht 29
5.3.6.2. Ablaufbeschreibung 30
5.3.7. SET_CERT_PUK_SUBCA_TAG 31
5.3.7.1. Kommando- und Antwortnachricht 31
5.3.7.2. Ablaufbeschreibung 32
5.3.8. SET_CERT_PUK_TAG 33
5.3.8.1. Kommando- und Antwortnachricht 33
5.3.8.2. Ablaufbeschreibung 33
5.3.9. RESET 35
5.3.9.1. Kommando- und Antwortnachricht 35
5.3.9.2. Ablaufbeschreibung 35
5.4. PUT DATA 36
5.4.1. Kommando- und Antwortnachricht 36
5.4.2. Ablaufbeschreibung 36
5.5. GET DATA 38
5.5.1.1. Kommando- und Antwortnachricht 38
5.5.2. Ablaufbeschreibung 38
5.6. AUTHENTICATE SERVICE-ID 39
5.6.1. Kommando- und Antwortnachricht 39
5.6.2. Ablaufbeschreibung 39
5.7. VERIFY CERTIFICATE 41
5.7.1. Kommando- und Antwortnachricht 41
5.7.2. Ablaufbeschreibung 41
5.8. GET CHALLENGE 43
5.8.1. Kommando- und Antwortnachricht 43
5.8.2. Ablaufbeschreibung 43
5.9. INTERNAL AUTHENTICATE 44
5.9.1.1. Kommando- und Antwortnachricht 44
5.9.2. Ablaufbeschreibung 44
5.10. EXTERNAL AUTHENTICATE 45
5.10.1.1. Kommando- und Antwortnachricht 45
5.10.2. Ablaufbeschreibung 45

6 Ergänzungskommandos des VDV-KA Nutzermediums 46
6.1. VERIFY SERVICE-ID 46
6.1.1. Kommando- und Antwortnachricht 46
6.1.2. Ablaufbeschreibung 47
6.2. GET SERVICE-ID 48
6.2.1. Kommando- und Antwortnachricht 48
6.2.2. Ablaufbeschreibung 49
7 Appendix 50
7.1. Glossar 50
7.2. Referenzen 51

I Abbildungen

Abbildung 1 Lebenszyklus eines Mediums 2
Abbildung 2 Übersicht Sammelzustand APP SELECTABLE 10
Abbildung 3 Übersicht Sammelzustand CONFIGURATION 12
Abbildung 4 Übersicht Authentifizierung der Tag-Applikation 14
Abbildung 5 Ablauf der Sevice-ID Authentifizierung 17

Tabellen

Tabelle 1 Datenobjekt PrK.Config 5
Tabelle 2 Datenobjekt PuK.Root 5
Tabelle 3 Struktur Tag-ID 6
Tabelle 4 Struktur Service-ID 6
Tabelle 5 Datenobjekt Service-Verzeichnis 7
Tabelle 6 Datenobjekt Eigentümerdaten 8
Tabelle 7 Datenobjekt Service-Quittung 8
Tabelle 8 Datenobjekt FCI 9
Tabelle 9 Unterzustände CONFIGURATION 11
Tabelle 10 Authentifizierungsdaten 13
Tabelle 11 Kommandoverfügbarkeit in den Zuständen CFG und OP 18
Tabelle 12 Secure Messaging Kommandodaten 19
Tabelle 13 Secure Messaging Antwortdaten 20
Tabelle 14 Returncodes 21

Dokumentenorganisation

Änderungshistorie

Version	Datum	Person	Beschreibung
0.1	2011-01-07	G. Galka	Initiale Version
0.2	2011-02-07	J. Lutgen	Review
0.3	2011-02-14	G. Galka	Einarbeitung Review Kommentare Ergänzung NM Kommandos VERIFY SERVICE-ID und GET SERVICE-ID
0.9	2011-02-18	J. Lutgen	Review
0.91	2011-02-19	O.Waltes	Review
0.92	2011-02-22	G.Galka	Einarbeitung Kommentare
0.93	2011-03-14	G.Galka	Abschließende Einarbeitung letzter Kommentare
1.0	2011-03-21	O.Waltes WSC	Endredaktion

1 Kennung des VDV KA Tag

Die von der VDV-KA KG für die Tag-Applikation vergebene AID lautet

'D2 76 00 01 35 4B 41 54 41 47 30 31 00'.

2 Zustände des VDV-KA Tag-Mediums

Die „VDV-KA Tag-Spezifikation“ macht keine Annahmen über die technische Ausprägung des Mediums, welches die Tag-Applikation enthält. Um die Einbringung grundlegender Daten in Tag-Medien zu vereinheitlichen, müssen Medien, die als VDV-KA-Tag eingesetzt werden sollen den VDV-KA Sicherheitsanforderungen für Secure Elements (s.[KASEC]) entsprechen.

Die Einbindung der Applikation und des Mediums in der VDV-KA Sicherheitsinfrastruktur findet durch die Einbringung der Root-Daten (s. §3.1.2) statt. Dies geschieht entweder durch den Medienhersteller in gesicherter Umgebung oder durch den App-Loader unter Verwendung von GlobalPlatform [GP] Prozessen. Die Root-Daten enthalten sowohl das Zertifikat der VDV-KA Root-CA, als auch ein Schlüsselpaar PkP.Config, mit dem die Applikation ihre Authentizität gegenüber der externen Welt nachweisen kann.

2.1. Lebenszyklus des Mediums

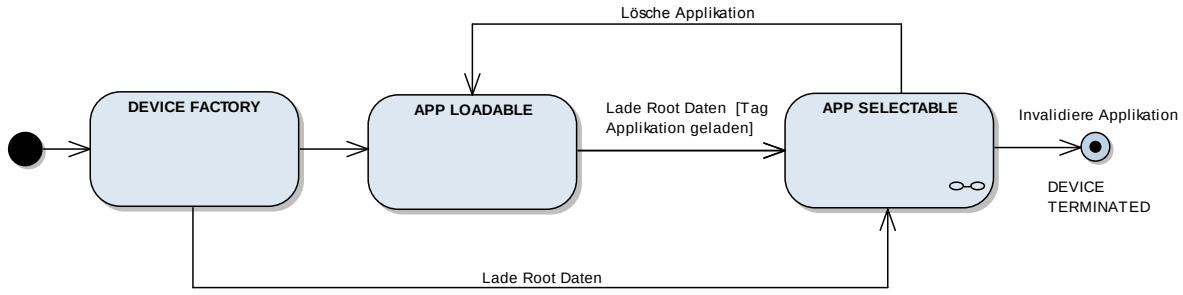

Abbildung 1 Lebenszyklus eines Mediums

2.1.1. Zustand DEVICE FACTORY

Im Zustand DEVICE FACTORY stehen betriebssystemspezifische Kommandos des Mediums zur Verfügung. Diese ermöglichen die Überführung des Mediums in einen Folgezustand, in dem entweder die Applikation geladen oder konfiguriert werden kann.

Abhängig von der technischen Ausprägung des Mediums kann die Applikation bereits im Produktionszustand DEVICE FACTORY im Medium enthalten sein, z.B. als Bestandteil des ROM. Ist dies der Fall, wird durch die Einbringung der Root-Daten die Applikation in den Folgezustand APP SELECTABLE überführt.

Ist die Applikation noch nicht im Medium enthalten, so wird durch die Einbringung herstellerspezifischer Daten das Medium in den Zustand APP LOADABLE überführt.

2.1.2. Zustand APP LOADABLE

Im Zustand APP LOADABLE ist die Applikation nicht selektierbar. Für die Einbringung der Applikation muss das Medium dem GlobalPlatform Standard [GP] entsprechen. Der Prozess der Applikationseinbringung wird nicht in dieser Spezifikation festgelegt.

Nach dem erfolgreichen Laden der Applikation wird das Medium durch die Einbringung der Root Daten (s. § 3.1) in den Zustand APP SELECTABLE überführt.

2.1.3. Zustand APP SELECTABLE

Der Zustand APP SELECTABLE wird durch die Einbringung der Root Daten (s. §4.1) durch den App-Loader oder Hersteller erreicht.

Im Zustand APP SELECTABLE sind je nach Zustand der Applikation entweder nur das Kommando CONFIGURE zur Konfiguration oder – nach erfolgter Konfiguration – die in Kapitel 5 spezifizierten Kommandos verfügbar.

2.1.4. Zustand DEVICE TERMINATED

Zur Beendung des Medienlebenszyklus wird das Medium in den Zustand DEVICE TERMINATED überführt. Dieser Zustand ist irreversibel und kann sowohl durch das Sperren der Applikation, als auch durch die Vernichtung des Mediums herbeigeführt werden.

3 Datenstrukturen

3.1. Root Daten

Die Root Daten werden - je nach technischer Ausprägung des Mediums - vom Medienhersteller oder vom App-Loader in das Medium eingebracht. Dies geschieht bei der Überführung der Applikation in den Zustand APP SELECTABLE (s. §2.1.3). Auf die eingebrachten Root-Daten wird im Rahmen der Konfiguration zurückgegriffen, um die Applikation gegenüber dem Konfigurator zu authentisieren und den Sitzungsschlüssel SKi festzulegen, der für die Absicherung der Kommunikation zwischen Konfigurator und Applikation verwendet wird.

3.1.1. PrK.Config Daten

Zur Authentifizierung der Tag-Applikation durch den Konfigurator wird vom App-Loader der PrK.Config eingebracht. Der PuK.Config wird anhand der vom App-Loader festgelegten Schlüssel-ID identifiziert. Die Schlüssel-ID ist 6 Byte lang und setzt sich aus der Org.-ID des App-Loaders und einer 4 Byte langen App-Loader-spezifischen ID zusammen. Der App-Loader stellt sicher, dass die Schlüssel-ID eindeutig ist.

Position	Länge	Wert	Beschreibung
1	1	'A0'	Datenobjekt PrK.Config
2	*3*	'82 XX XX'	Länge des Datenobjekts
3	1	'81'	Tag Schlüssel-ID des PrK.Config
4	1	'06'	Länge der Schlüssel-ID
5	2	'XX XX XX XX XX XX'	Org.-ID App-Loader (2 Byte) | App-Loader spezifische Schlüssel-ID (4 Byte)
6	1	'82'	Tag Schlüsseltyp
7	1	'02'	Länge des Schlüsseltyps
8	2	'00 01'	RSA CRT 2048 Bit
9		'A1'	Tag Schlüsseldaten PrK.Config in CRT Format
10		'82 XX XX'	Länge der Schlüsseldaten PrK.Config
11		'84'	Tag für den ersten Primfaktor des Modulus
12		'81 XX'	Länge
13		'XX … XX'	Wert von P, wobei PQ= Modulus
14		'85'	Tag für zweiten Primfaktor des Modulus
15		'81 XX'	Länge
16		'XX … XX'	Wert von Q, wobei PQ= Modulus
17		'86'	Tag für Exponent 1
18		'81 XX'	Länge
19		'XX … XX'	Wert von DP1 = d mod P-1 (wobei d der private Exponent

Position	Länge	Wert	Beschreibung
			ist)
20		'87'	Tag für Exponent 2
21		'81 XX'	Länge
22		'XX … XX'	Wert von DQ1 = d mod Q-1 (wobei d der private Exponent ist)
23		'88'	Tag für Koeffizient
24		'81 XX'	Länge
25		'XX … XX'	Wert von R = 1/Q mod P

Tabelle 1 Datenobjekt PrK.Config

3.1.2. PuK.Root Daten

Der für die Zertifikatsprüfungen erforderliche öffentliche Schlüssel der VDV-KA Root-CA (PuK.Root) sowie die dazugehörige CAR müssen vor der Konfiguration in die Applikation eingebracht werden. Die CAR dient während der Konfiguration zur Identifizierung der Root-CA.

Tag	Länge	Wert				
’7F 21’	’82 01 3F‘					
		Tag	**Länge**	**Wert**		
		’5F 29’	’01’	’07’ = CPI		
		’42’	’08’	'XX ... XX' = CAR (8 Byte)		
		’5F 20’	’0C’	'XX ... XX' = CHR (12 Byte)		
		’5F 4C’	‘07’	‘XX … XX’ = CHA (7 Byte)		
		’5F 24’	’04’	’JJ JJ MM TT’ = EOV		
		’06’	‘09‘	’2A 86 48 86 F7 0D 01 01 05’ = OID		
		’7F 49’	’82 01 01‘	Öffentlicher Schlüssel		
				Tag	**Länge**	**Wert**
				’81’	’81 F8‘	’XX … XX’ = Modulus (1984 Bit)
				’82’	’04’	'XX ... XX' = öffentlicher Exponent [4 Byte]

Tabelle 2 Datenobjekt PuK.Root

3.2. Tag-ID

Die Tag-ID setzt sich aus der VDV-KA Org.-ID des Tag-Eigentümers gefolgt von der Tag-Nummer zusammen. Die Tag-Nummer ist im Kontext des Tag-Eigentümers eindeutig. Sie wird während der Konfiguration in das Tag eingebracht und ist Bestandteil der CHR[1] des Komponentenzertifikats Cert.PuK.Tag.

Position	Länge	Beschreibung
1	2	Org.-ID des Tag Eigentümers
2	4	Tag-Nummer

Tabelle 3 Struktur Tag-ID

3.3. Service-ID

Die Service-ID ist der eindeutige Bezeichner eines Tag-Service. Sie setzt sich aus der VDV-KA Org.-ID des Serviceanbieters gefolgt von der Service-Nummer zusammen. Die Service-Nummer wird vom Serviceanbieter festgelegt und ist in dessen Kontext eindeutig.

Position	Länge	Beschreibung
1	2	Org.-ID des Serviceanbieters
2	4	Service-Nummer

Tabelle 4 Struktur Service-ID

[1] Die Informationen der Tag-ID werden wie folgt in die CHR des Zertifikats kodiert: Org.-ID | '**FE' (Indikatorbyte für Tag-Zertifikat)** | 'XX XX XX XX' (Tag-Nummer) | 'JJ JJ MM TT' (Gültigkeitsbeginn des Zertifikats) | '00' (RFU) (siehe KA SAM SPEC V1.107 §4.3.1.3.3)

3.4. Service-Verzeichnis

Das Service-Verzeichnis gibt eine Übersicht über alle in der Tag-Applikation installierten Service-IDs. Es können maximal 16 Service-IDs hinterlegt werden. Das Verzeichnis enthält den Verzeichnisheader gefolgt von maximal 16 Verzeichniseinträgen.

Der im Verzeichnisheader enthaltene Tag-Sequenz-Counter (TSC) wird von der Tag-Applikation verwaltet. Nach Konfiguration der Applikation ist der TSC mit 0 vorbelegt. Die Tag-Applikation inkrementiert den TSC bei jeder nicht flüchtigen Schreiboperation.

Position	Länge	Wert	Beschreibung
1	1	'E0'	Tag Verzeichnis
2	1 - 3	'XX ... XX'	Länge Tag Verzeichnis
3	1	'80'	Tag Verzeichnisheader
4	1	'05'	Länge Verzeichnisheader
5	1	'XX'	Anzahl der Verzeichniseinträge
6	4	'XX ..XX'	Aktueller Tag Sequenz Zähler (TSC) Datentyp: INT4
Maximal 16 Verzeichniseinträge			
1	1	'C0'	Tag Verzeichniseintrag
2	1	'0E'	Länge Verzeichniseintrag
3	6	'XX ... XX'	Service-ID (s. §3.3)
4	4	'XX ... XX'	Installationszeit und - datum Datentyp: DateTimeCompact
5	4	'XX ... XX'	TSC zum Zeitpunkt der Installation der Service-ID Datentyp: INT4

Tabelle 5 Datenobjekt Service-Verzeichnis

3.5. Eigentümerdaten

Die Eigentümerdaten ermöglichen es dem Tag-Eigentümer, Daten seiner Wahl im Tag zu hinterlegen. Die Länge des im Tag hierfür reservierten Speicherplatzes ist fix.

Position	Länge	Wert	Beschreibung
1	1	'E1'	Tag Eigentümerdaten
2	1 od. 2	['XX' od. '81 XX' od. '82 XX XX']	Länge Eigentümerdaten
2	1	'81'	Tag URL
3	1	'XX'	Länge URL (0 - 127)
4	var.	'XX ... XX'	Info-URL (String gem. ISO/IEC 8859-15)
5	1	'82'	Info
6	1	'XX'	Länge Info (0 - 127)
7	var.	'XX ... XX'	Info-URL (String gem. ISO/IEC 8859-15)

Tabelle 6 Datenobjekt Eigentümerdaten

3.6. Service-Quittung

Position	Länge	Wert	Beschreibung
1	1	'E2'	Tag Service-Quittung
2	1	'1A'	Länge Service-Quittung
2	6	'XX ... XX'	Tag-ID
3	4	'XX ... XX'	Aktueller Wert des Tag Sequenzzählers (TSC)
4	1	'C0'	Tag Verzeichniseintrag
5	1	'0E'	Länge Verzeichniseintrag
6	6	'XX ... XX'	Tag Service-ID (s. §3.3)
7	4	'XX ... XX'	Installationszeit und -datum der Service-ID Datentyp: DateTimeCompact
8	4	'XX ... XX'	TSC zum Zeitpunkt der Installation der Service-ID Datentyp: INT4

Tabelle 7 Datenobjekt Service-Quittung

3.7. FCI

Position	Länge	Wert	Beschreibung	Zustand
1	1	'6F'	Tag FCI	CFG/OP.
2	1 - 3	['XX' od. ['81'] 'XX' od. ['82'] 'XX XX'	Länge der FCI	
3	1	'84'	Tag DF Name	
4	1	'0C'	Länge DF Name	
5	12	Tag AID	AID der Tag-Applikation	
6	1	'A5'	Tag Kurzverzeichnis	
7	1 - 3	'XX' od. ['81'] 'XX' od. ['82'] 'XX XX'	Länge Tag Verzeichnis	
8	1	'80'	Tag Version	
9	1	'08'	Länge der Versionsinformationen	
10	2	'XX XX'	Org.-ID: Herstelleridentifikation vergeben durch VDV-KA KG	
11	2	'XX XX'	Tag-Spezifikationsversion (BCD-kodiert)	
12	4	'XX ... XX'	Herstellerspezifische Versionsnummer	
13	1	'42'	Tag Tag-CAR	
14	1	'08'	Länge Tag-CAR	
15	8	'XX … XX'	CAR der Tag-CA	
16	1	'81'	Tag Tag-ID	OP.
17	1	'06'	Länge der Tag-ID	
18	6	'XX … XX'	Tag-ID	
19	1	'82'	Tag Kapazität	
20	1	'01'	Länge Tag Kapazität	
21	1	'00' – '16'	Anzahl der noch verfügbaren Speicher-plätze für Service-IDs	
Für jede im Tag gespeicherte Service-ID ist ein Datenobjekt '83' vorhanden:				
22	1	'83'	Tag Service-ID	
23	1	'06'	Länge Service-ID	
24	6	'XX ... XX'	Service-ID	

Tabelle 8 Datenobjekt FCI

4 Prozesse

4.1. Konfiguration der Applikation

Durch das Einbringen der Root Daten (s. §3.1) wird die Applikation vom Zustand FACTORY in den Zustand APP SELECTABLE überführt. Die Applikation kann selektiert werden und steht für die Einbringung der für den VDV-KA Wirkbetrieb erforderlichen Daten bereit.

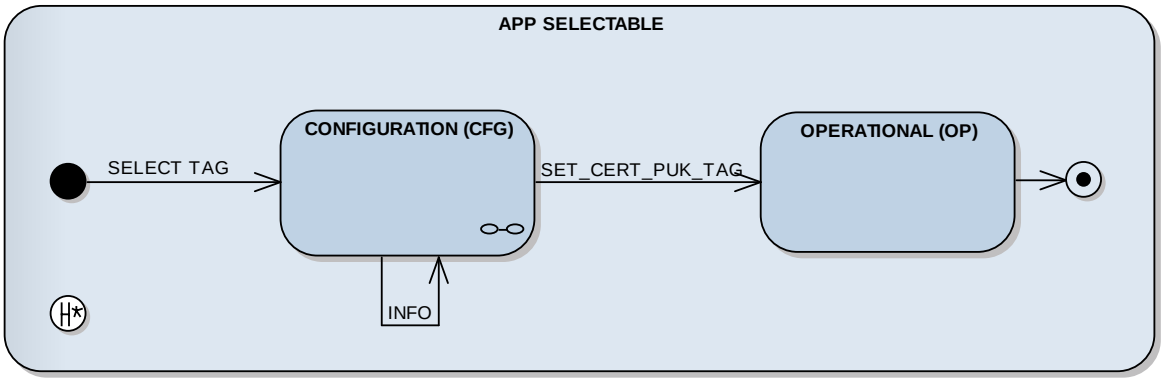

Abbildung 2 Übersicht Sammelzustand APP SELECTABLE

Eine nicht initialisierte Applikation befindet sich im Zustand CONFIGURATION (CFG). Dies ist genau dann der Fall, wenn die Applikation mit der AID 'D2 76 00 01 35 4B 41 54 41 47 30 31 00' selektierbar ist, die Root Daten eingebracht wurden und die Konfiguration noch nicht abgeschlossen wurde.

Neben dem Kommando SELECT FILE (s. §5.2) unterstützt die Applikation im Zustand CFG ausschließlich das Kommando CONFIGURE (s. §5.3). Andere Kommandos werden von der selektierten Applikation mit entsprechendem Returncode (s. §5.1.4) abgelehnt.

Mit dem Kommando CONFIGURE werden Daten in die Applikation eingebracht, die zur Überführung der Applikation in den Zustand OPERATIONAL erforderlich sind.

Folgende Datenstrukturen werden von der Applikation als leeres Datenobjekt selbst angelegt:

- Eigentümerdaten (s. §3.5)
- Service-Verzeichnis (s. §3.4)

4.1.1. Sammelzustand CONFIGURATION (CFG)

Während der Konfiguration durchläuft die im Zustand CFG befindliche Applikation eine Reihe von nicht-flüchtigen Unterzuständen. Der Ablauf der Zustandsübergänge wird mit dem Kommando CONFIGURE (s. §5.3) gesteuert.

Nach dem Reset startet die Applikation im Sicherheitszustand LOCKED. Bevor Daten eingebracht werden können, muss die externe Welt einen Sitzungsschlüssel festlegen. Die Festlegung erfolgt im Rahmen der Authentisierung der Applikation. Nach der Festlegung des Sitzungsschlüssels befindet sich die Applikation im flüchtigen Sicherheitszustand AUTH.

Zustand		**ID**	**Beschreibung**
LOCKED		'FF'	Die Applikation ist selektiert, der Konfigurator muss die Applikation authentifizieren.
AUTH ***flüchtig***		*Var.*	Die Applikation ist selektiert, das Initialisierungssystem hat die Applikation authentifiziert und der flüchtige Sitzungsschlüssel SKi sowie der Sendefolgenzähler SSC sind vereinbart worden.
	NO DATA	'00'	Ausgangszustand der Applikation nach dem Laden der Applikation oder nach erfolgreicher Ausführung des Kommandos CFG_RESET.
	PKP	'10'	Die Applikation hat ihr Schlüsselpaar erzeugt und nicht-flüchtig gespeichert.
	PUK_TAG	'20'	Der öffentliche Schlüssel der Tag-Applikation wurde ausgelesen und die Tag-ID gesetzt.
	CERT_PUK_SUBCA_TAG	'30'	Das Zertifikat einer VDV-KA Komponenten-CA für die Tag-Applikation wurde geladen.

Tabelle 9 Unterzustände CONFIGURATION

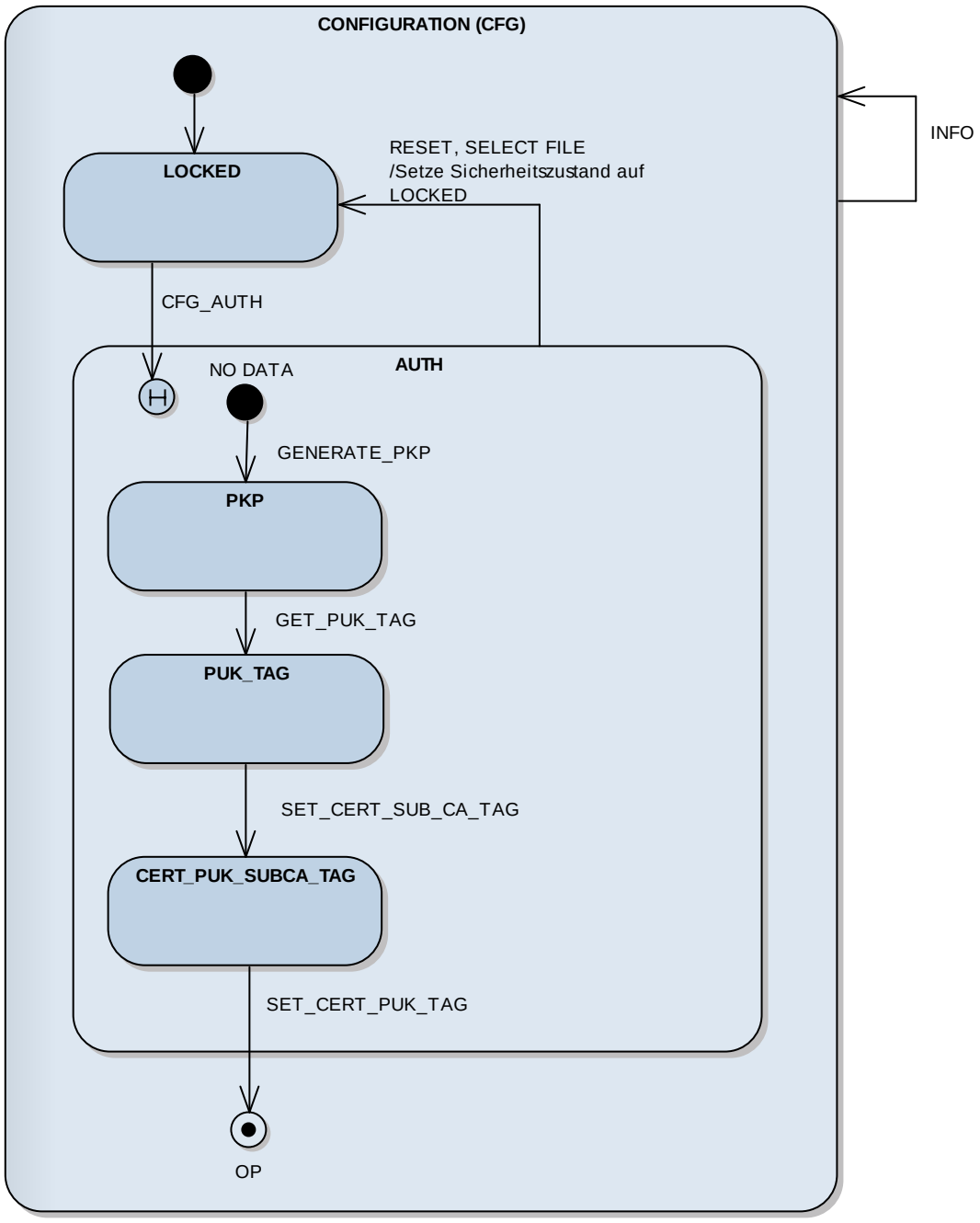

Abbildung 3 Übersicht Sammelzustand CONFIGURATION

4.2. Authentifizierung der Applikation

Um sicherzustellen, dass der Konfigurator keine Wirkdaten in nicht authentische Medien einbringt, muss das Konfigurationssystem in der Lage sein, Medien vor der Konfiguration zu authentifizieren. Im Rahmen der Authentifizierung wird der temporäre Sitzungsschlüssel SKi vereinbart, der zur Absicherung der weiteren Kommunikation mit dem Medium dient.

Für die Authentifizierung wird das asymmetrische Schlüsselpaar (PuK.Config, PrK.Config) genutzt, dass vom App-Loader erzeugt wurde und in die Applikation eingebracht wurde. Der öffentliche Teil PuK.Config wird Konfiguratoren zur Verfügung gestellt. Der private Teil PrK.Config wird bei der Überführung der Applikation in den Zustand CONFIGURATION (4.1.1) als Teil des PrK.Config-Datenobjekts (s. §3.1.1) in die Applikation eingebracht.

4.2.1. Authentifizierungsdaten

Bezeichung	Länge in Byte	Beschreibung
C.Config	16	Zufallszahl erzeugt durch den Konfigurator. Wird verschlüsselt an die Applikation übergeben.
CD.Auth	256	Kryptogramm über die Daten SKi | C.Config erzeugt mit dem PuK.Config
MAC.Auth	16	MAC über die Zufallszahl C.Config im CBC Modus, erzeugt mit dem Schlüssel SKi.
Ski	32	Sitzungsschlüssel für das Verfahren AES-256 zur Integritätssicherung der Kommunikation. Wird vom Konfigurator erzeugt.
SSC	16	Sequenzzähler zur Absicherung der Kommunikation zwischen Konfigurator und Applikation. Der Startwert des SSC ist C.Config.

Tabelle 10 Authentifizierungsdaten

4.2.2. Ablauf der Authentifizierung

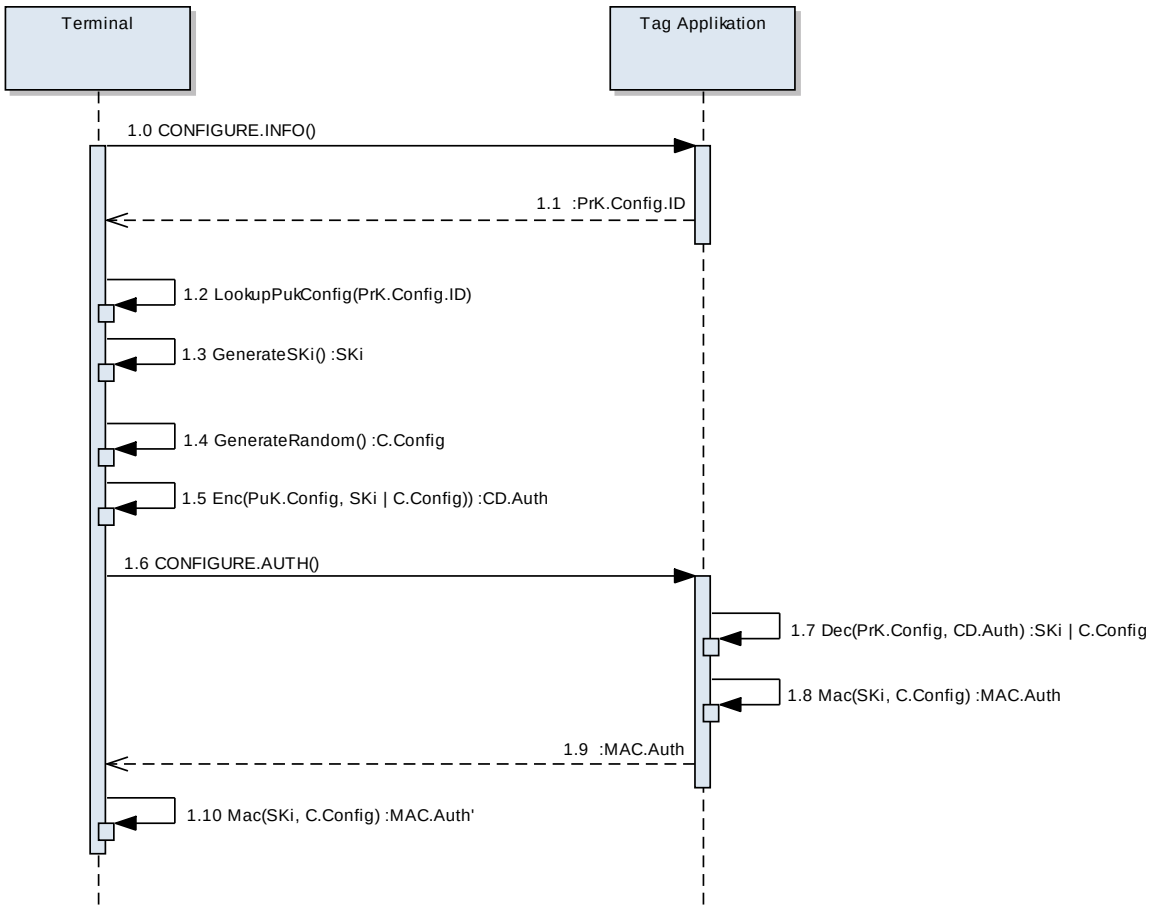

Abbildung 4 Übersicht Authentifizierung der Tag-Applikation

Der Konfigurator ermittelt die in der Tag-Applikation hinterlegte Schlüssel ID des PrK.Config und sucht den entsprechenden öffentlichen Schlüssel PuK.Config im eigenen System.

Der Konfigurator erzeugt die Sitzungsschlüssel SKi und die Zufallszahl C.Config, verschlüsselt die verketteten Daten SKi | C.Config mit dem PuK.Config zum Kryptogramm CD.Auth und übermittelt CD.Auth an die Applikation.

Die Tag-Applikation entschlüsselt CD.Auth. War die Entschlüsselung erfolgreich, übernimmt sie den Sitzungsschlüssel SKi und setzt den SSC := C.Config.

Anschließend erhöht die Tag-Applikation den SSC um 1 und berechnet den CBC MAC (MAC.Auth) über die Zufallszahl C.Config mit dem Ski und AES-256. Der MAC.Auth wird in den Antwortdaten als Quittung ausgegeben.

Der Konfigurator setzt seinerseits den SSC := C.Config, inkrementiert den SSC um 1 und berechnet den MAC.Auth'. Anschließend vergleicht der Konfigurator MAC.Auth' mit dem von der Tag-Applikation empfangenen MAC.Auth. Sind die Daten identisch, gilt die Tag-Applikation als authentisch und die

Applikation befindet sich im flüchtigen Sicherheitszustand AUTH. Sind MAC.Auth' und MAC.Auth ungleich, bricht der Konfigurator den Vorgang mit einer Fehlermeldung ab und löscht C.Config und SKi.

4.2.3. Berechnung des Kryptogramms CD.Auth

Über die Daten SKi | C.Config berechnet der Konfigurator die kodierte Nachricht EM der Länge 255 Bytes (Bytelänge des Modulus - 1) gemäß EME-OAEP-Encoding (siehe Abschnitt A.2.3 [VDVSAM]) unter Verwendung des leeren Parameterstrings P.

Der Konfigurator verschlüsselt EM mit dem öffentlichen Schlüssel RSA-Schlüssel PuK.Config zum Kryptogramm CD.Auth.

4.3. Eigentümer Authentisierung

Die Authentisierung des Eigentümers erfolgt durch die gegenseitige Authentisierung zwischen dem SAM des Eigentümers und der Tag-Applikation. Der Prozess ist in [VDVSAM] spezifiziert.

4.4. Hinzufügen einer Service-ID

Vor dem Hinzufügen einer Service-ID, muss sich der Eigentümer den Authentisierungszustand herstellen (s. §4.2). War dies erfolgreich, wird das Verzeichniseintrag Datenobjekt 'C0' (s. §3.4) für den Service zusammengestellt. Es fließen sowohl die Service-ID als auch aktuelles Datum und Uhrzeit ein. Der Tag Sequenz Zähler (TSC), der ebenfalls Bestandteil des Verzeichniseintrags ist, wird von der Tag-Applikation überschrieben.

Das Verzeichniseintrag-Datenobjekt wird anschließend mit Secure Messaging und dem Kommando PUT DATA an die Tag-Applikation übergeben.

4.5. Löschen einer Service-ID

Das Löschen einer Service-ID ist analog zum Hinzufügen. Die Tag-Applikation löscht einen Verzeichniseintrag, wenn dieser erneut hinzugefügt wird.

4.6. Ungesichertes Lesen einer Tag-ID und Service-ID

Die in einem VDV-KA Tag hinterlegten Service-IDs werden bei der Ausführung des Kommandos SELECT FILE (s. §5.2) als Bestandteil der FCI (s. §3.7) ausgegeben. Um kryptografisch sicherzustellen, dass ein Tag tatsächlich über die Service-IDs verfügt, muss das Terminal die Service-ID Authentifizieren (s. §4.7).

4.7. Authentifizierung einer Service-ID

Um nachvollziehbar zu prüfen, ob eine Service-ID in der VDV-KA Tag-Applikation authentisch hinterlegt ist, wird von der Tag-Applikation das Kommando AUTHENTICATE SERVICE-ID bereitgestellt. AUTHENTICATE SERVICE-ID signiert die Anfrage des Terminals mit dem privaten Schlüssel PuK.Tag der Tag-Applikation. Die Signatur wird anschließend von der Nutzermedium-Applikation mit dem öffentlichen Schlüssel PuK.Tag verifiziert, der dem Zertifikat Cert.PuK.Tag entnommen wurde.

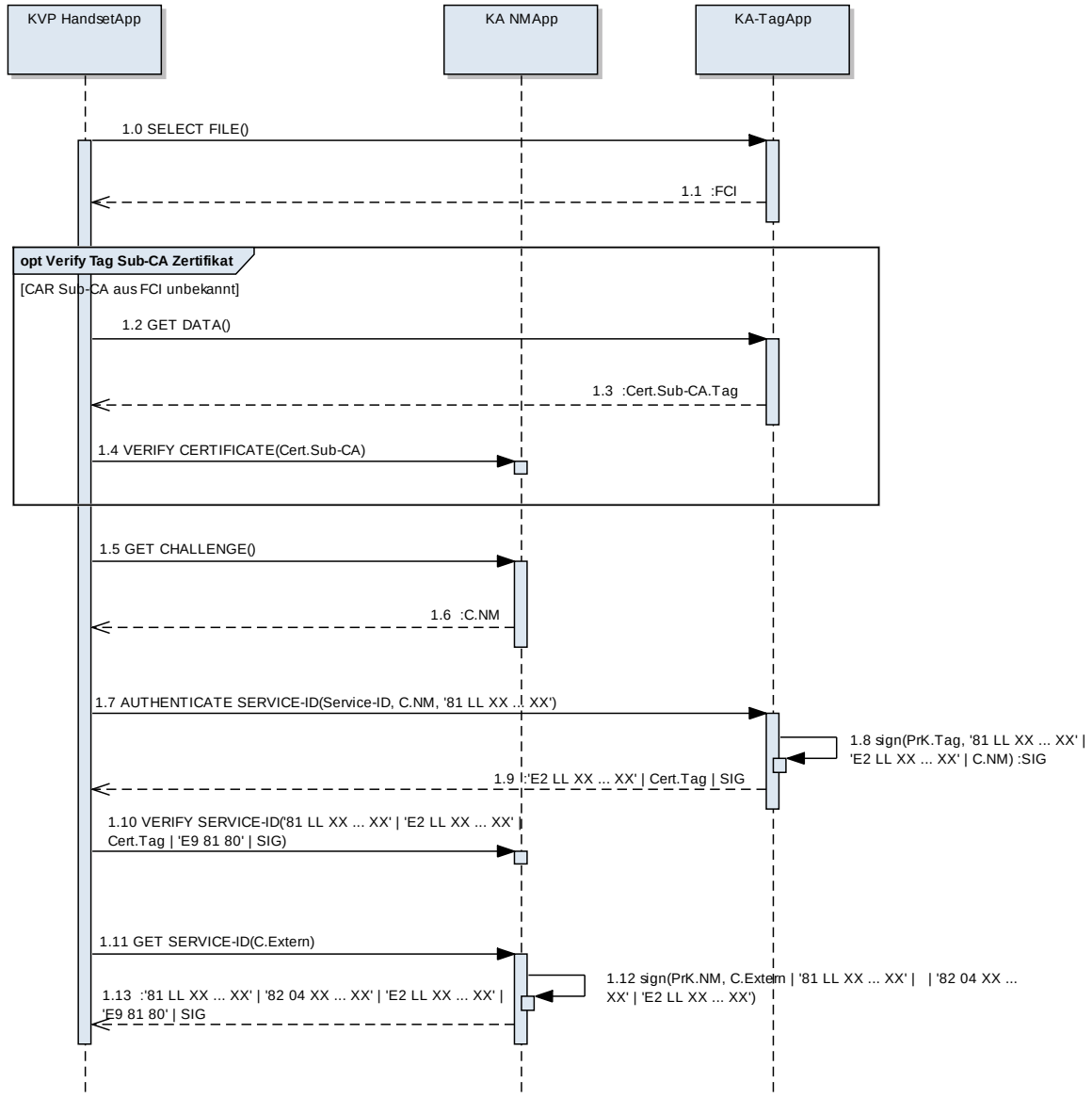

Abbildung 5 Ablauf der Sevice-ID Authentifizierung

5 Kommandos der VDV-KA Tag-Applikation

Je nach Applikationszustand stehen unterschiedliche Kommandos zur Verfügung.

	Verfügbarkeit	
	CFG	**OP**
SELECT FILE	●	●
CONFIGURE	●	○
PUT DATA	○	●
GET **DATA**	○	●
AUTHENTICATE SERVICE-ID	○	●
VERIFY CERTIFICATE	○	●
GET CHALLENGE	○	●
INTERNAL AUTHENTICATE	○	●
EXTERNAL AUTHENTICATE	○	●

Tabelle 11 Kommandoverfügbarkeit in den Zuständen CFG und OP

5.1. Festlegungen zur Kommandoausführung

5.1.1. Formale Prüfungen

Nach Empfang eines Kommandos führt die Applikation formale Prüfungen durch bevor die Kommandoverarbeitung fortgesetzt wird.

Das Format der Kommando-APDU wird auf Korrektheit geprüft, d.h. die Kodierung von Lc (wenn vorhanden) und Le (wenn vorhanden) wird gemäß ISO/IEC 7816-4 geprüft. Im Fehlerfall wird die Verarbeitung des Kommandos abgebrochen und der Returncode '67 00' ausgegeben.

Der Parameter INS wird auf Zulässigkeit geprüft, d.h. ob das Kommando bekannt ist. Im Fehlerfall wird die Verarbeitung des Kommandos abgebrochen und der Returncode '6D 00' ausgegeben.

Es wird geprüft, ob der Wert von CLA zulässig ist und die Kombination mit dem Wert von INS der Spezifikation entspricht. Im Fehlerfall wird die Verarbeitung des Kommandos abgebrochen und der Returncode '6E 00' ausgegeben.

Die Parameter P1/P2/Lc/Le werden im Kontext des durch CLA und INS definierten Kommandos geprüft. Hierbei werden zulässige Bereiche oder Konstanten überprüft. Im Fehlerfall wird die Verarbeitung des Kommandos abgebrochen und der Returncode '6A 86' bei unzulässigen Werten für P1/P2 oder '67 00' bei unzulässigen Werten für Lc oder Le ausgegeben.

Wenn im CLA-Byte Secure Messaging (SM) signalisiert wird, erfolgt die Prüfung des Datenobjekts für SM Datenobjekte. Im Fehlerfall wird die Verarbeitung des Kommandos abgebrochen und der Returncode '69 87' oder '69 88' oder '6A 80' oder '6A 88' ausgegeben.

Alle weiteren Parameter und Datenformate werden im Rahmen der kommandospezifischen Verarbeitung überprüft. Die Reihenfolge der beschriebenen Überprüfungen kann je nach Implementierung variieren.

5.1.2. Verketteter Modus

Übersteigt die Länge der an die Applikation zu übermittelnden Daten 255 Bytes, muss das Kommando im verketteten Modus (s.[VDVNM]) ausgeführt werden.

5.1.3. Secure Messaging

Secure Messaging wird durch das CLA Byte 'XC' angezeigt. Die Kommandodaten des Kommandos mit Secure Messaging enthalten optional ein Klartext-Datenobjekt und / oder ein Le-Datenobjekt gefolgt von einem MAC-Datenobjekt.

5.1.3.1. Kommandodaten

Klartextdatenobjekt (Kommandodaten vorhanden)			
	1	'81'	Tag für Klartextdatenobjekt
	1 bis 3	'XX' oder '81 XX' oder '82 XX XX'	Länge, L, der Kommandodaten aus der ungesicherten Command-APDU
	L	'XX ... XX'	Kommandodaten der ungesicherten Command-APDU
Le-Datenobjekt (wenn Antwortdaten erwartet werden).			
	1	'97'	Tag für Le-Datenobjekt
	1	'01'	Länge
	1	'00'	Le-Feld der ungesicherten Kommando-APDU
MAC-Datenobjekt			
	1	'8E'	Tag für MAC-Object
	1	'10'	Länge des MAC
	8	'XX ... XX'	MAC_SKi (kryptographische Checksumme)

Tabelle 12 Secure Messaging Kommandodaten

Der Message Authentication Code (MAC) wird mit dem AES Krypto-Algorithmus im CBC-Modus unter Verwendung des Schlüssels SKi berechnet (256 Bit).

Der MAC wird über den mit ISO-Padding auf Blocklänge erweiterten Kommandoheader gefolgt vom Klartextdatenobjekt und dem optionalen Le-Datenobjekt berechnet.

Entspricht die Länge der Eingangsdaten nicht einem Vielfachen der Blocklänge, werden sie ebenfalls mit ISO-Padding auf ein Vielfaches der Blocklänge erweitert.

Folgende Daten werden zusammengestellt:

```
HDR := CLA|INS|P1|P2|'80 00 00 00 00 00 00 00 00 00 00 00'
```

```
DATA := '81'|Lc|'XX ... XX'|['97 01'|Le|]'80 [00 ... 00]'
```

Vor der MAC-Berechnung wird der SSC um 1 inkrementiert.

Anschließend wird der MAC über die verketteten Daten `HDR|DATA` unter Verwendung des Sitzungsschlüssels SK_i berechnet. Als ICV dient der zuvor berechnete SSC.

5.1.3.2. Antwortdaten

Werden im Rahmen der abgesicherten Kommunikation Antwortdaten zurückgegeben, enthalten diese ein Klartext-Datenobjekt sowie ein Status-Datenobjekt, gefolgt von einem MAC-Datenobjekt.

Klartextdatenobjekt (wenn Antwortdaten zurückgegeben werden)			
	1	'81'	Tag für Klartextdatenobjekt
	1 bis 3	'XX' oder '81 XX' oder '82 XX XX'	Länge, La, der Antwortdaten
	L	'XX ... XX'	Antwortdaten zur Kommando-APDU
Status-Datenobjekt.			
	1	'99'	Tag für Status-Datenobjekt
	1	'02'	Länge
	1	'SW1 SW2'	Status-Bytes
MAC-Datenobjekt			
	1	'8E'	Tag für MAC-Object
	1	'10'	Länge des MAC
	8	'XX ... XX'	MAC_SKi (kryptographische Checksumme)

Tabelle 13 Secure Messaging Antwortdaten

Der MAC wird über das mit ISO-Padding auf Blocklänge erweiterte Klartextdatenobjekt (sofern vorhanden) und das Status-Datenobjekt berechnet.

```
DATA := ['81'|La|'XX ... XX'|]'99 02'|SW1 SW2|'80 [00 ... 00]'.
```

Vor der MAC-Berechnung wird der SSC um 1 inkrementiert.

Anschließend wird der MAC über die Daten DATA unter Verwendung des Sitzungsschlüssels SK_i berechnet. Als ICV dient der zuvor berechnete SSC.

5.1.4. Returncodes

Treten bei der Ausführung des Kommandos CONFIGURE Fehler auf, wird in den Antwortdaten einer der folgenden Fehlercodes ausgegeben.

Returncode (SW1SW2)	Beschreibung
'67 00'	Falsche Länge
'63 CF'	Überprüfung der Signatur fehlgeschlagen
'69 85'	Nutzungsbedingungen nicht erfüllt
'6A 81'	Funktion/Sub-Kommando nicht unterstützt
'6A 80'	Fehlerhafte Kommandodaten
'68 83'	Finales Kommando der Kommandokette erwartet
'69 88'	SM Datenobjekte fehlen
'69 88'	SM Datenobjekte inkorrekt
'6A 84'	Unzureichender Speicherplatz in der Datei
'6A 86'	Unerlaubter Parameter P1 oder P2
'6A 88'	Daten nicht gefunden
'6D 00'	Unbekanntes Kommando (INS)
'6F XX'	Applikationsspezifischer Fehlercode

Tabelle 14 Returncodes

5.2. SELECT FILE

Das Kommando SELECT FILE selektiert die VDV-KA Tag-Applikation und gibt die FCI zusammen mit den Verzeichnisinformationen (FCI) des Tag aus.

5.2.1.1. Kommando- und Antwortnachricht

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'00'	CLA
2	1	'A4'	INS
3	1	'04'	P1
4	1	'00'	P2
5	1	'0C'	Lc
6	12	'D2 76 00 01 35 4B 41 41 4D 30 31 00'	AID des VDV KA TAG
7	1	'00'	Le

Antwort APDU

Als Antwortdaten wird die FCI zusammen mit den Verzeichnisdaten (s. §3.7) der Tag-Applikation ausgegeben. Die Verzeichnisinformation wird im Datenobjekt 'A5' hinterlegt und ist abhängig vom Applikationszustand (s. §3.7).

Position	Länge	Wert	Beschreibung
1	n	'XX...XX'	FCI mit Verzeichnisdaten (s. §3.7)
2	2	'90 00'	Returncode

5.3. CONFIGURE

Mit dem Kommando CONFIGURE werden die für den Wirkbetrieb erforderlichen Daten in die Applikation eingebracht. P1 gibt bei der Kommandoausführung die auszuführende Operation (Sub-Kommando) an. Einige Sub-Kommandos liefern Antwortdaten, andere nicht. Deshalb wird das Le-Byte teilweise vorhanden sein, aber nicht immer. Wenn es vorhanden ist, wird hier immer der Wert ‘00‘ verwendet.

CONFIGURE und damit auch die Sub-Kommandos kann ausschließlich im Zustand CONFIGURE genutzt werden. Befindet sich die Applikation im Zustand OPERATIONAL, ist das Kommando unbekannt, d.h. die Applikation antwortet mit dem Returncode '6D 00'.

5.3.1. Kommandoaufbau

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'XX'	CLA
2	1	'16'	INS
3	1	'XX'	P1 Sub-Kommando
4	1	'00'	P2
5	1	'XX'	Lc
6	var.	'XX ... XX'	Kommandodaten
7	1	'00'	Le (falls Antwortdaten erwartet)

Antwort APDU

Antwortdaten sind abhängig vom gewählten Sub-Kommando.

Position	Länge	Wert	Beschreibung
1	var.	'XX ... XX'	Antwortdaten
2	2	'90 00'	Returncode

5.3.2. Übersicht CONFIGURE Sub-Kommandos

Sub-Kommando (P1)		*Ausgangszustand*	CONFIGURATION (CFG)					*OP*
			LOCKED	NO DATA	PKP	PUK_TAG	CERT_PUK_SUBCA_TAG	
			'FF'	'00'	'10'	'20'	'30'	
INFO	'01'		'FF'	'00'	'10'	'20'	'30'	-
AUTH	'10'		***H***[1)]	-	-		-	-
GENERATE_PKP	'02'		-	'10'	-	-	-	-
GET_PUK_TAG	'03'		-	-	'20'	-	-	-
SET_CERT_PUK_SUBCA_TAG	'04'		-	-	-	'30'		-
SET_CERT_PUK_TAG	'05'		-	-	-	-	***OP***[2)]	-
RESET	'FF'		-	'FF'	'FF'	'FF'	'FF'	-
			Folgezustand					

1) H letzter gültiger, nicht-flüchtig gespeicherter Zustand.

2) OP Zustand OPERATIONAL, Konfiguration abgeschlossen

5.3.3. INFO

INFO gibt den aktuellen Zustand der Initialisierung und zusätzliche Informationen zur Ermittlung des PuK.Config aus.

5.3.3.1. Kommando- und Antwortnachricht

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'00'	CLA
2	1	'16'	INS
3	1	'00'	P1 = '00' für INFO
4	1	'00'	P2
5	1	'00'	Le

Antwort APDU

Nach erfolgreicher Ausführung gibt das Kommando die Antwortdaten gefolgt vom Returncode '90 00' aus.

Position	Länge	Wert	Beschreibung
1	21 oder 26	'XX ... XX'	Antwortdaten
2	2	'90 00'	Returncode

Antwortdaten

Position	Länge	Wert	Beschreibung
1	1	'XX'	Aktueller Zustand (s.4.1.1)
2	8	'XX … XX'	CAR der Root-CA [VDVSAM]
3	6	'XX … XX'	Schlüssel-ID des PrK.Config
4	6	'XX … XX'	Wurde der PuK.Tag bereits ausgelesen, wird die beim Auslesen eingebrachte Tag-ID zurückgegeben.

5.3.4. AUTH

Mit dem Sub-Kommando AUTH wird die Applikation in den flüchtigen Sicherheitszustand AUTH (s. §4.1.1) überführt. Dafür muss der Aufrufer über den Schlüssel PuK.Config verfügen und das Kryptogramm CD.Auth erzeugen (s. §4.2.3), das in den Kommandodaten an die Applikation übergeben wird. Gleichzeitig wird der für den weiteren Konfigurationsvorgang erforderliche Sitzungsschlüssel SKi sowie der SSC festgelegt. Die Applikation belegt ihre Authentizität mit der Quittung MAC.Auth, die in den Antwortdaten ausgegeben wird.

5.3.4.1. Kommando- und Antwortnachricht

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'X0'	CLA *mit Command Chaining*[2]
2	1	'16'	INS
3	1	'10'	P1 = '10' für AUTH
4	1	'00'	P2
5	1	var.	Lc
6	var.	'XX ... XX'	Kommandodaten
7	1	'00'	Le

Antwort APDU

Nach erfolgreicher Ausführung gibt das Kommando die Antwortdaten gefolgt vom Returncode '90 00' aus.

Position	Länge	Wert	Beschreibung
1	16	'XX ... XX'	MAC.Auth
2	2	'90 00'	Returncode

5.3.4.2. Ablaufbeschreibung

Die Applikation prüft die Kommandodaten formal (s. §5.1.1).

Die Applikation prüft, ob das Sub-Kommando im aktuellen Zustand zugelassen ist. Ist dies nicht der Fall, wird die Ausführung mit dem Returncode '69 85' abgebrochen.

Nach dem Empfang des letzten Teilkommandos entschlüsselt die Applikation das empfangene Kryptogramm CD.Auth mit dem PrK.Config. Schlägt die Entschlüsselung fehl, wird das Kommando mit dem Returncode '6A 80' abgebrochen.

[2] Ein Kryptogramm von 256 Byte Länge wird in zwei Kommandoaufrufen (verkettet) an die Tag-Applikation übergeben.

Die Applikation speichert den im Klartext vorliegenden Schlüssel SKi flüchtig für die weitere Verwendung und setzt den SSC := C.Config. Anschließend erhöht die Applikation den SSC um 1 und berechnet den MAC.Auth indem der CBC-MAC (ICV := SSC) mit dem SKi über C.Config gebildet wird. Das Ergebnis der Berechnung wird in den Antwortdaten ausgegeben.

5.3.5. GENERATE_PKP

Das Sub-Kommando GENERATE_PKP erzeugt das RSA Schlüsselpaar der Tag-Applikation. Das Schlüsselpaar wird in der sicheren Umgebung der Applikation ("on-Card") generiert. Nach erfolgreicher Ausführung des Kommandos kann der öffentliche Schlüssel PuK.Tag ausgelesen werden.

5.3.5.1. Kommando- und Antwortnachricht

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'0C'	CLA (für Secure Messaging)
2	1	'16'	INS
3	1	'02'	P1 = '02' für GENERATE_PKP
4	1	'00'	P2
5	1	'10'	Lc
6	1	'8E'	Tag für MAC Datenobjekt
7	1	'10'	Länge
8	16	'XX ... XX'	MAC

Antwort APDU

Im Erfolgsfall gibt die Applikation den Returncode '90 00' aus.

Position	Länge	Wert	Beschreibung
1	2	'90 00'	Returncode

5.3.5.2. Ablaufbeschreibung

Die Applikation prüft die Kommandodaten formal (s. §5.1.1).

Die Applikation prüft, ob das Sub-Kommando im aktuellen Zustand zugelassen ist. Ist dies nicht der Fall, wird die Ausführung mit dem Returncode '69 85' abgebrochen.

Die Applikation erhöht den intern, flüchtig gespeicherten SSC-Wert um Eins und prüft den MAC über die Kommandodaten mit ICV = SSC. Schlägt die Prüfung fehl, wird das Kommando mit dem Returncode '69 88' abgebrochen.

Liegt das Schlüsselpaar der Applikation bereits vor, wird der Returncode '90 00' ausgegeben. Ansonsten generiert die Applikation das Schlüsselpaar PkP.Tag, speichert die Daten nicht-flüchtig und gibt im Anschluss den Returncode '90 00' aus.

Nach erfolgreicher Ausführung des Sub-Kommandos befindet sich die Applikation im nicht-flüchtigen Zustand PKP.

5.3.6. GET_PUK_TAG

Das Sub-Kommando GET_PUK_TAG dient zum gesicherten Auslesen des öffentlichen Schlüssels PuK.Tag der Applikation. Gleichzeitig wird die Tag-ID der Applikation gesetzt, die zur eindeutigen Identifikation der Applikation im Verlauf der Initialisierung Verwendung finden kann.

5.3.6.1. Kommando- und Antwortnachricht

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'0C'	CLA (für Secure Messaging)
2	1	'16'	INS
3	1	'03'	P1 = '03' für GET_PUK_TAG
4	1	'00'	P2
5	1	'1A'	Lc
6	1	'81'	Klartext Datenobjekt Tag-ID
7	1	'06'	Länge
8	6	'XX ... XX'	Tag-ID
9	1	'8E'	Tag für MAC-Datenobjekt
10	1	'10'	Länge
11	16	'XX ... XX'	MAC
12	1	'00'	Le

Antwort APDU

Im Erfolgsfall gibt die Applikation folgende Antwortdaten gefolgt vom Returncode '90 00' aus.

Position	Länge	Wert	Beschreibung
1	1	'81'	Klartext Datenobjekt Schlüsselda-ten
2	2	'81 9F'	Länge
3	2	'7F 49'	Tag Datenobjekt PuK.Tag
4	2	'81 89'	Länge
5	1	'81'	Tag Modulus
6	2	'81 80'	Länge Modulus
7	128	'XX ... XX'	Modulus
8	1	'82'	Tag öffentlicher Exponent
9	1	'04'	Länge öffentlicher Exponent

10	4	'XX ... XX'	Öffentlicher Exponent
11	1	'8E'	Tag für MAC Datenobjekt
12	1	'10'	Länge
13	16	'XX ... XX'	MAC
14	2	'90 00'	Returncode

5.3.6.2. Ablaufbeschreibung

Die Applikation prüft die Kommandodaten formal (s. §5.1.1).

Die Applikation prüft, ob das Sub-Kommando im aktuellen Zustand zugelassen ist. Ist dies nicht der Fall, wird die Ausführung mit dem Returncode '69 85' abgebrochen.

Die Applikation erhöht den intern, flüchtig gespeicherten SSC-Wert um Eins und prüft den MAC über die Kommandodaten mit ICV = SSC. Schlägt die Prüfung fehl, wird das Kommando mit dem Returncode '69 88' abgebrochen.

Die Applikation übernimmt die Tag-ID aus dem Klartext Datenobjekt und speichert diese nicht-flüchtig.

Anschließend stellt die Applikation den PuK.Tag im Klartext Datenobjekt der Antwortdaten bereit, erhöht den SSC-Wert um Eins, berechnet den MAC über das Klartext Datenobjekt (s. §5.1.3.2) mit ICV = SSC und gibt die Antwortdaten gefolgt vom Returncode '90 00' aus.

Nach erfolgreicher Ausführung des Sub-Kommandos befindet sich die Applikation im nicht-flüchtigen Zustand PUK_TAG.

5.3.7. SET_CERT_PUK_SUBCA_TAG

Mit dem Sub-Kommando SET_CERT_PUK_SUBCA_TAG wird das Zertifikat Cert.PuK.Sub-CA über den öffentlichen Schlüssel der Sub-CA, die das Zertifikat über den öffentlichen Schlüssel der Applikation ausstellt, verifiziert und eingebracht. Das Zertifikat ist ein CV-Zertifikat mit Signatur gemäß PKCS#1_v1.5.

Ferner wird der öffentliche Schlüssel aus dem Zertifikat so abgelegt, dass dieser im OPERATIONAL Zustand durch das Kommando VERIFY CERTIFICATE der Applikation direkt in einer einstufigen Verifikation eines SAM-Authentisierungszertifikats verwendet werden kann.

Das Kommando verwendet Command-Chaining und Secure Messaging.

5.3.7.1. Kommando- und Antwortnachricht

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'XC'	CLA '0C' für einfachen oder letzten Kommandoaufruf im verketteten Modus '1C' für den ersten bis vorletzten Kommandoaufruf im verketteten Modus
2	1	'16'	INS
3	1	'04'	P1 = '04' für SET_CERT_PUK_SUBCA_TAG
4	1	'00'	P2
5	1	'XX'	Lc
6	1	'81'	Klartext Datenobjekt
7	1 od. 2	['81'] 'XX'	Länge des Klartext Datenobjekts
8	*var.*	'XX ... XX'	Zertifikatsdaten
9	1	'8E'	Tag für MAC-Datenobjekt
10	1	'10'	Länge des MAC
11	16	'XX ... XX'	MAC

Antwort APDU

Im Erfolgsfall gibt die Applikation den Returncode '90 00' aus.

Position	Länge	Wert	Beschreibung
1	2	'90 00'	Returncode

5.3.7.2. Ablaufbeschreibung

Die Applikation prüft die Kommandodaten formal (s. §5.1.1).

Die Applikation prüft, ob das Sub-Kommando im aktuellen Zustand zugelassen ist. Ist dies nicht der Fall, wird die Ausführung mit dem Returncode '69 85' abgebrochen.

Die Applikation erhöht den intern, flüchtig gespeicherten SSC-Wert um Eins und prüft den MAC über die Kommandodaten mit ICV=SSC. Schlägt die Prüfung fehl, wird das Kommando mit dem Returncode '69 88' abgebrochen.

Wurde ein Teilkommando empfangen, übernimmt die Applikation die im Klartext Datenobjekt enthaltenen Daten für die spätere Prüfung und gibt den Returncode '90 00' aus.

Wurde das letzte Teilkommando empfangen, prüft die Applikation das Zertifikat. Hierzu wird die Signatur mit dem in den Config-Daten empfangenen PuK.Root überprüft. Schlägt die Prüfung fehl, wird das Kommando mit dem Returncode '63 CF' abgebrochen.

War die Prüfung des Zertifikats erfolgreich, wird das Zertifikat nicht-flüchtig gespeichert und der Returncode '90 00' ausgegeben.

Nach erfolgreicher Ausführung des Sub-Kommandos befindet sich die Applikation im nicht-flüchtigen Zustand CERT_PUK_SUBCA_TAG.

5.3.8. SET_CERT_PUK_TAG

Mit dem Sub-Kommando SET_CERT_PUK_TAG wird das Zertifikat Cert-PuK-Tag über den öffentlichen Schlüssel PuK.Tag verifiziert und in die Applikation eingebracht. Nach erfolgreicher Einbringung wird die Applikation in den nicht-flüchtigen Zustand OPERATIONAL (s. §4.1) überführt.

5.3.8.1. Kommando- und Antwortnachricht

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'XC'	CLA '0C' für einfachen oder letzten Kommandoaufruf im verketteten Modus '1C' für den ersten bis vorletzten Kommandoaufruf im verketteten Modus
2	1	'16'	INS
3	1	'05'	P1 = '05' für SET_CERT_PUK_TAG
4	1	'00'	P2
5	3	'XX'	Lc
6	1	'81'	Tag für Klartext Datenobjekt
7	3	'81 XX'	Länge des Klartext Datenobjekts
8	*var.*	'XX ... XX'	CV-Zertifikat des NM gemäß [VDVSAM]
9	1	'8E'	Tag für MAC-Datenobjekt
10	1	'10'	Länge des MAC
11	16	'XX ... XX'	MAC

Antwort APDU

Im Erfolgsfall gibt die Applikation den Returncode '90 00' aus.

Position	Länge	Wert	Beschreibung
1	2	'90 00'	Returncode

5.3.8.2. Ablaufbeschreibung

Die Applikation prüft die Kommandodaten formal (s. §5.1.1).

Die Applikation prüft, ob das Sub-Kommando im aktuellen Zustand zugelassen ist. Ist dies nicht der Fall, wird die Ausführung mit dem Returncode '69 85' abgebrochen.

Die Applikation erhöht den intern, flüchtig gespeicherten SSC-Wert um Eins und prüft den MAC über die Kommandodaten mit ICV=SSC. Schlägt die Prüfung fehl, wird das Kommando mit dem Returncode '69 88' abgebrochen.

Wurde ein Teilkommando empfangen, übernimmt die Applikation die im Klartext Datenobjekt enthaltenen Daten für die spätere Prüfung und gibt den Returncode '90 00' aus.

Wurde das letzte Teilkommando empfangen, prüft die Applikation das empfangene Zertifikat.

Hierzu wir die Signatur mit dem zuvor eingebrachten öffentlichen Schlüssel der Sub-CA geprüft. Schlägt die Prüfung fehl, wird das Kommando mit dem Returncode '63 CF' abgebrochen.

Die in den Zertifikatsdaten enthaltene Tag-ID wird mit dem beim Auslesen des PuK.Tag eingebrachten Wert (s. §5.3.6) verglichen[3]. Sind die Daten unterschiedlich, wird das Kommando mit dem Returncode '6A 80' abgebrochen.

War die Prüfung des Zertifikats erfolgreich, wird das Zertifikat nicht-flüchtig gespeichert und der Returncode '90 00' ausgegeben.

Nach erfolgreicher Ausführung des Sub-Kommandos wird die Applikation nicht-flüchtig in den Zustand OPERATIONAL (s. §4.1) überführt.

[3] Kodierung der CHR des Zertifikats (= Org.-ID | '**FE' (Indikatorbyte für Tag-Zertifikat)** | 'XX XX XX XX' (Tag-Nummer) | 'JJ JJ MM TT' (Gültigkeitsbeginn des Zertifikats) | '00' (RFU)) im Vergleich zur Tag-ID = Org.-ID | 'XX XX XX XX' (Tag-Nummer).

5.3.9. RESET

Das Sub-Kommando RESET setzt den Konfigurationszustand auf NO DATA (s. §4.1.1) zurück. Alle im Rahmen der Konfiguration eingebrachten oder generierten Daten werden gelöscht. Nach erfolgreicher Ausführung des Kommandos, wird der flüchtige Sicherheitszustand auf LOCKED gesetzt und alle temporären Schlüssel gelöscht.

5.3.9.1. Kommando- und Antwortnachricht

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'0C'	CLA
2	1	'16'	INS
3	1	'FF'	P1 = 'FF' für RESET
4	1	'00'	P2
5	1	'12'	Lc
6	1	'8E'	Tag für MAC-Datenobjekt
7	1	'10'	Länge
8	16	'XX ... XX'	MAC

Antwort APDU

Im Erfolgsfall gibt die Applikation den Returncode '90 00' aus.

Position	Länge	Wert	Beschreibung
1	2	'90 00'	Returncode

5.3.9.2. Ablaufbeschreibung

Die Applikation prüft die Kommandodaten formal (s. §5.1.1).

Die Applikation prüft, ob das Sub-Kommando im aktuellen Zustand zugelassen ist. Ist dies nicht der Fall, wird die Ausführung mit dem Returncode '69 85' abgebrochen.

Die Applikation erhöht den intern, flüchtig gespeicherten SSC-Wert und prüft den MAC über die Kommandodaten mit ICV=SSC. Schlägt die Prüfung fehl, wird das Kommando mit dem Returncode '69 88' abgebrochen.

Alle im Rahmen der Konfiguration eingebrachten Daten werden gelöscht. Bei erneuter Durchführung der Konfiguration müssen die Daten erneut eingebracht werden. Root Daten werden nicht gelöscht.

Der nicht-flüchtige Subzustand NO DATA (s. §4.1.1) wird gesetzt und der flüchtige Sicherheitszustand mit LOCKED überschrieben. Die Applikation stellt sicher, dass SKi und SSC ungültig sind.

Abschließend wird der Returncode '90 00' ausgegeben.

5.4. PUT DATA

Mit dem Kommando PUT DATA werden BER-TLV kodierte Daten in die Tag-Applikation eingebracht. PUT DATA kann nur nach erfolgreicher gegenseitiger Authentisierung zwischen dem SAM des Tag-Eigentümers und der Applikation genutzt werden. Die Kommandodaten werden mit Secure Messaging integritätsgeschützt.

5.4.1. Kommando- und Antwortnachricht

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'XC'	Secure Messaging erforderlich. Verketteter Modus: '0C' einfacher oder letzter Kommandoaufruf '1C' erster bis vorletzter Kommandoaufruf
2	1	'DA'	INS
3	1	'02'	P1 = '02'
4	1	'XX'	P2 = Tag des zu schreibenden Datenobjekts
5	1	var.	Lc
6	1	'81'	Tag für Klartextdatenobjekt
7	1 od. 2	['81'] Lc'	Lc' der ungesicherten Kommandodaten
8	Lc'	'XX ... XX'	Kommandodaten des ungesicherten Kommandos
9	1	'8E'	Tag für MAC Datenobjekt
10	1	'08'	Länge des MAC
11	8	'XX ... XX'	MAC.SKi

Antwort APDU

Nach erfolgreicher Ausführung gibt das Kommando den Returncode '90 00' aus.

Position	Länge	Wert	Beschreibung
1	2	'90 00'	Returncode

5.4.2. Ablaufbeschreibung

Die Applikation prüft das Kommando formal (s. §5.1.1).

Ist das in P2 angegebene Datenobjekt unbekannt, so wird das Kommando mit dem Returncode '6A 88' abgebrochen.

P2 muss entweder ein Verzeichniseintrag 'C0' oder die Eigentümerdaten 'E1' adressieren. Ist dies nicht der Fall, wird das Kommando mit dem Returncode '6A 88' abgebrochen.

Ist das Kommando ein Folgekommando, prüft die Applikation, ob das durch P2 referenzierte Datenobjekt dem im ersten Teilkommando referenzierten entspricht. Schlägt die Prüfung fehl, wird das Kommando mit dem Returncode '6A 86' abgebrochen.

Die Applikation erhöht den intern flüchtig gespeicherten SSC-Wert um Eins und prüft den MAC.SKi. Schlägt die Prüfung fehl, wird die Kommandoausführung mit dem Returncode '69 88' abgebrochen.

Ist das Kommando das einzige oder erste Kommando[4], prüft die Applikation, ob das in den Kommandodaten enthaltene Tag dem in P2 übergebenen Wert entspricht. Ist dies nicht der Fall, wird das Kommando mit dem Returncode '6A 80' abgebrochen.

Die Applikation speichert die empfangenen Daten flüchtig und prüft über alle in der Kommandokette empfangenen Daten, ob diese die für das zu schreibende Datenobjekt maximal zulässige Länge überschreiten. Ist die der Fall, wir das Kommando mit dem Returncode '6A 84' abgebrochen.

Im Fall eines Teilkommandos einer Kommandokette wird die Kommandoausführung mit dem Returncode '90 00' beendet.

Ist das empfangene Kommando das einzige oder letzte Kommando einer Kommandokette, wird das flüchtig gespeicherte Datenobjekt formal geprüft. Schlägt die Prüfung fehl, wird das Kommando mit dem Returncode '6A 80' abgebrochen. War die Prüfung erfolgreich, wird das durch P2 referenzierte Datenobjekt mit dem zuvor flüchtig gespeicherte Datenobjekt wie folgt überschrieben:

Handelt es sich bei dem zu speichernden Datenobjekt um ein Verzeichniseintrag-Datenobjekt (s. §3.4), prüft die Applikation, ob die im Datenobjekt enthaltene Service-ID bereits im Tag vorliegt. Ist dies der Fall, inkrementiert die Applikation den TSC um Eins und löscht die bekannte, bereits gespeicherte Service-ID. Wurde die Service-ID nicht gefunden, prüft die Applikation, ob noch Platz zum Speichern des neuen Verzeichniseintrags vorhanden ist. Schlägt die Prüfung fehlt, wird das Kommando mit dem Returncode '6A 84' abgebrochen. Ist freier Speicherplatz vorhanden, inkrementiert die Applikation den TSC um Eins, übernimmt den Wert in den zu speichenden Verzeichniseintrag und speichert das neue Verzeichniseintrag Datenobjekt nicht flüchtig.

Handelt es sich bei dem zu speichernden Datenobjekt um ein Eigentümer-Daten Datenobjekt, inkrementiert die Applikation den TSC um Eins und überschreibt das vorhandene Eigentümer-Daten Datenobjekt.

Anschließend wird die Kommandoausführung mit dem Returncode '90 00' beendet.

[4] Die Applikation muss den Zustand für die Kommandokette verwalten

5.5. GET DATA

Mit dem Kommando GET DATA werden sowohl das Zertifikat Cert.PuK.Sub-CA der Sub-CA, das Zertifikat der Tag-Applikation Cert.PuK.Tag als auch das Service-Verzeichnis (s. §3.4) und die Eigentümerdaten (s. §3.5) ausgelesen.

Übersteigt die Länge der von der Applikation zu lesenden Daten 256 Bytes, muss das Kommando im verketteten Modus (s. [VDVNM]) ausgeführt werden.

5.5.1.1. Kommando- und Antwortnachricht

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'XX'	CLA 'X0' ohne Secure Messaging '00' für einfachen oder letzten Kommandoaufruf im verketteten Modus '10' für den ersten bis vorletzten Kommandoaufruf im verketteten Modus
2	1	'CA'	INS
3	1	'01' od. '02'	P1
4	1	'XX'	P1 = '01' & P2 = '11': auslesen des Cert.PuK.Tag P1 = '01' & P2 = '12': auslesen des Cert.PuK.Sub-CA Wenn P1 = '02', dann enthält P2 das Tag des zu lesende Datenobjekts.
5	1	'00'	Le

Antwort APDU

Nach erfolgreicher Ausführung gibt das Kommando die Antwortdaten gefolgt vom Returncode '90 00' aus.

Position	Länge	Wert	Beschreibung
1	var.	'XX...XX'	Antwortdaten
2	2	'90 00'	Returncode SW1 SW2

5.5.2. Ablaufbeschreibung

Die Applikation prüft das Kommando formal (s. §5.1.1).

Die Applikation stellt abhängig von P1 das in P2 referenzierte Datenobjekt gefolgt vom Returncode '90 00' in den Antwortdaten bereit. Überschreitet die Größe des Datenobjekts 256 Byte, kann das Datenobjekt im Verketteten Modus ausgegeben werden (s. §5.1.2).

5.6. AUTHENTICATE SERVICE-ID

Mit dem Kommando AUTHENTICATE SERVICE-ID belegt die Tag-Applikation gegenüber der externen Welt, dass die übergebene Service-ID im Tag installiert ist. Zusätzlich können optional Zusatzdaten an das Tag übergeben werden, die in die Signaturberechnung einbezogen werden.

Da die Länge der Antwortdaten 256 überschreitet, wird das Kommando immer im verketteten Modus genutzt.

5.6.1. Kommando- und Antwortnachricht

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'XC'	Secure Messaging erforderlich. Verketteter Modus: '0C' einfacher oder letzter Kommandoaufruf '1C' erster bis vorletzter Kommandoaufruf
2	1	'8A'	INS
3	1	'00'	P1
4	1	'00'	P2
5	1	'XX'	Lc
6	1	'80'	Tag Authentication Daten
7	1	'0A'	Länge Authentication Daten
8	8	'XX ... XX'	Zufallszahl C.NM
9	6	'XX ... XX'	Service-ID (s. §3.3)
10	1	'81'	Tag Zusatzdaten
11	1	'XX'	Länge Zusatzdaten (max. 64)
12	Var.	'XX ... XX'	Zusatzdaten
13	1	'00'	Antwortdaten erwartet

Antwort APDU

Nach erfolgreicher Ausführung gibt das Kommando die Antwortdaten gefolgt vom Returncode '90 00' aus.

Position	Länge	Wert	Beschreibung
1	var.	'XX...XX'	Antwortdaten
2	2	'90 00'	Returncode

5.6.2. Ablaufbeschreibung

Die Applikation prüft das Kommando formal (s. §5.1.1).

Fehlt ein Datenobjekt oder sind die Datenobjekte fehlerhaft codiert, wird das Kommando mit dem Returncode '6A 80' abgebrochen.

Ist das Kommando ein einfacher Kommandoaufruf und liegen keiner Ergebnisdaten (s.u.) zur Ausgabe vor, wird die Ausführung mit dem Returncode '6A 88' abgebrochen.

Ist das Kommando das erste Kommando einer Kommandokette[5], speichert die Applikation die Kommandodaten flüchtig. Anschließend sucht die Applikation den Verzeichniseintrag zu der in den Kommandodaten übergebene Service-ID. Wird kein passender Verzeichniseintrag gefunden, bricht die Applikation die Kommandoausführung mit dem Returncode '6A 88' ab.

Aus der Tag-ID, dem aktuellen TSC sowie dem Verzeichniseintrag wird das Service-Quittung Datenobjekt (s.§3.6) zusammengestellt.

Anschließend stellt die Applikation die Daten für die Signaturberechnung aus den in den Kommandodaten übergebenen Datenobjekten '80' und '81' und dem Service-Quittung Datenobjekt zusammen.

SIG := sign(PrK.Tag)['81 LL XX ... XX' | 'E2 LL XX ... XX' | C.NM]

Die Signatur SIG wird in den Ergebnisdaten SIG zusammen mit dem Verzeichniseintrag und dem Zertifikat Cert.PuK.Tag des Tag für die Ausgabe bereitgestellt.

Ergebnisdaten := 'E2 LL XX ... XX' | Cert.PuK.Tag | SIG

Anschließend werden die ersten 256 Byte der Ergebnisdaten gefolgt vom Returncode '90 00' ausgegeben.

Ist das Kommando ein Teilkommando oder das letzte Kommando einer Kommandokette werden die Kommandodaten mit den zuvor flüchtig gespeicherten Daten des ersten Teilkommandos der Kommandokette verglichen. Sind die Daten nicht identisch, wird die Ausführung mit dem Returncode '6A 80' abgebrochen. Die Applikation stellt noch verbliebende Ergebnisdaten in den Antwortdaten gefolgt vom Returncode '90 00' bereit. Wurden die Ergebnisdaten komplett ausgegeben, wird nur der Returncode '90 00' ausgegeben.

[5] Die Applikation muss den Zustand für die Kommandokette verwalten

5.7. VERIFY CERTIFICATE

Das Kommando VERIFY CERTIFICATE dient zur Einbringung des öffentlichen Authentisierungsschlüssels des SAM, der im Rahmen der gegenseitigen Authentisierung zwischen Tag und SAM verwendet wird. Die Zertifikatsprüfung ist zweistufig, d.h. es muss zuerst das Sub-CA Zertifikat des SAM geprüft werden, bevor das Authentisierungszertifikat des SAM geprüft wird.

Spezifikation und Ablauf des Kommandos sind analog zum Ablauf des in [VDVNM] spezifizierten VERIFY CERTIFICATE Kommandos.

Die Applikation prüft ein Sub-CA-Zertifikat unter Verwendung des intern gespeicherten öffentlichen Root-Schlüssels PuK.Root der VDV-KA PKI. Dieser wurde im Rahmen der Konfiguration in die Applikation eingebracht und nicht flüchtig gespeichert.

Alternativ kann zur Zertifikatsprüfung ein im Kontext der Applikation nicht flüchtig gespeicherter PuK.Sub-CA herangezogen werden (einstufige Verifikation des SAM Zertifikats). Dieser öffentliche Sub-CA-Schlüssel der VDV-PKI wurde zuvor mit VERIFY CERTFICATE als Bestandteil eines Zertifikats eingebracht.

VERIFY CERTIFICATE kann im verketteten Modus mehrfach ausgeführt werden, wenn die Kommandodaten eine Länge von 255 Byte übersteigen.

Das Kommando VERIFY CERTIFICATE kann für Zertifikate gemäß [VDVSAM] (Kapitel 3.3 Zertifikate) durchgeführt werden und muss eine Moduluslänge von mindestens 1984 Bit unterstützen.

5.7.1. Kommando- und Antwortnachricht

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'X0'	CLA '00' einfacher oder letzter Kommandoaufruf verketteter Modus '10' Kommandoaufruf im verketteten Modus
2	1	'2A'	INS
3	1	'00'	P1
4	1	'AE'	P2
5	1	'XX'	Lc Länge der Kommandodaten
6	variabel	'XX..XX'	Daten für die Zertifikatsprüfung, Datenobjekte '5F4E', '5F37'

Antwort APDU

Wurde das Kommando erfolgreich ausgeführt, wird der Returncode '90 00' zurückgegeben.

Position	Länge	Wert	Beschreibung
1	2	'90 00'	

5.7.2. Ablaufbeschreibung

Die Applikation prüft das Kommando formal (s. §5.1.1).

Schlägt die Prüfung der Codierung oder der Reihenfolge der Datenobjekte ('5F4E' || '5F37') fehl, wird das Kommando mit dem Returncode '69 85' abgebrochen.

Anschließend prüft die Applikation den Inhalt des Zertifikat- Datenobjekts '5F 4E'. Falls hierbei inkonsistente Daten gefunden werden, wird das Kommando mit dem Returncode '6A 80' abgebrochen.

Die Applikation vergleicht die in der CHR des SAM-Authentisierungszertifikats enthaltene Org.-ID mit der Org.-ID des Tag-Eigentümers. Sind sie Org.-IDs nicht identisch, wird das Kommando mit dem Returncode '6A 80' abgebrochen.

Dann wird der Schlüssel mit der Rollen-ID des Zertifikatsinhalts zur Prüfung des Zertifikats gesucht.

Für den gefundenen öffentlichen Schlüssel wird überprüft, ob und wie er verwendet werden darf. Falls der Schlüssel kein CA-Schlüssel ist (Rollen-ID 0x30), wird das Kommando mit dem Returncode '69 85' abgebrochen.

Zertifikate, die eine andere Rollen-ID als 0x30 oder 0x4X mit X=2 oder 3 beinhalten, werden mit dem Returncode '69 85' abgewiesen.

Anhand der Algorithmus-ID wird festgestellt, ob VERIFY CERTIFICATE im Rahmen des dem Schlüssel zugeordneten Sicherheitsalgorithmus auf den Schlüssel zugreifen darf. Dies ist nur für das Verfahren zur Einbringung eines öffentlichen Schlüssels mittels Zertifikatsprüfung gemäß [VDVSAM] der Fall. Sollte dem Schlüssel ein anderer Sicherheitsalgorithmus zugeordnet sein, wird das Kommando mit dem Returncode '69 85' abgelehnt.

Die Applikation prüft wie in [VDVSAM] beschrieben die Signatur. Schlägt die Signaturprüfung fehl, wird das Kommando mit dem Returncode '63 CF' abgebrochen.

Falls die Zertifikatsprüfung erfolgreich verläuft, wird geprüft, ob die OID gleich '2A 86 48 86 F7 0D 01 01 05' im Fall eines Sub-CA-Zertifikats (CPI = 3) oder die OID gleich '2B 24 03 05 02 02 01' im Fall eines SAM-Authentisierungszertifikats (CPI = 4) ist. Ist dies nicht der Fall, wird das Kommando mit dem Returncode '69 85' abgebrochen.

Die Schlüsselkomponenten des öffentlichen Schlüssels aus dem Zertifikat werden zusammen mit den für die weitere Verwendung notwendigen Zusatzinformationen (Rollen-ID aus CHA, SAM-ID aus CHR, Algorithmus-ID aus OID) flüchtig gespeichert. Es ist insbesondere sicherzustellen, dass der öffentliche Schlüssel eines Trust Centers (CHA.7=0x30) nur im Kommando VERIFY CERTIFICATE und der öffentliche Authentisierungsschlüssel eines SAM (CHA.7=0x42, 0x44 oder 0x48) nur in den Kommandos INTERNAL und EXTERNAL AUTHENTICATION verwendet werden kann. Das Kommando wird mit dem Returncode '90 00' beendet.

5.8. GET CHALLENGE

Mit dem Kommando GET CHALLENGE wird von der Tag-Applikation die 8 Byte lange Zufallszahl C.Tag angefordert.

5.8.1. Kommando- und Antwortnachricht

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'00'	CLA, Class-Byte
2	1	'84'	INS
3	1	'00'	P1
4	1	'00'	P2
5	1	'08'	Le Antwortdaten erwartet.

Antwort APDU

Wurde das Kommando erfolgreich ausgeführt, wird der Returncode '90 00' zurückgegeben.

Position	Länge	Wert	Beschreibung
1	8	'XX...XX'	Zufallszahl der Länge Le
2	2	'90 00'	Returncode SW1 SW2

5.8.2. Ablaufbeschreibung

Die Applikation prüft das Kommando formal (s. §5.1.1).

Anschließend generiert die Applikation die Zufallszahl C.Tag, speichert diese flüchtig für die weitere Verwendung und gibt sie in den Antwortdaten gefolgt vom Returncode '90 00' aus.

5.9. INTERNAL AUTHENTICATE

Mit dem Kommando INTERNAL AUTHENTICATE authentisiert sich die Tag-Applikation gegenüber dem SAM unter Nutzung des Schlüssels PuK.Auth.SAM.

Zu diesem Zweck werden von der externen Welt dynamische Authentikationsdaten erzeugt und in den Kommandodaten an die Applikation übergeben. Diese werden im Kommandoablauf von der Applikation vervollständigt, mit dem PrK.Tag signiert und das Ergebnis mit dem über die ID.Extern referenzierten Authentisierungsschlüssel PuK.Auth.SAM des SAM verschlüsselt. Das so erhaltene Kryptogramm wird in der Antwortnachricht ausgegeben.

5.9.1.1. Kommando- und Antwortnachricht

Das Datenfeld in der Kommandonachricht von INTERNAL AUTHENTICATE enthält immer eine 8 Byte lange Zufallszahl C.Extern sowie Identifikationsdaten ID.Extern der externen Welt (SAM).

Kommando APDU

Die Kommandonachricht des Kommandos INTERNAL AUTHENTICATE hat folgenden Aufbau:

Position	Länge	Wert	Beschreibung
1	1	'00'	CLA
2	1	'88'	INS
3	1	'00'	P1
4	1	'00'	P2 = '00': Schlüssel bekannt
5	1	'10'	Lc Länge der Kommandodaten
6	X	''XX ... XX''	Kommandodaten (s. [VDVNM])
7	1	'00'	Antwortdaten erwartet

Antwort APDU

Wurde das Kommando erfolgreich ausgeführt, werden die Antwortdaten gefolgt von dem Returncode '90 00' zurückgegeben:

Position	Länge	Wert	Beschreibung
1	X	'XX..XX'	Antwortdaten (s.[VDVNM])
2	2	'90 00'	Returncode SW1 SW2

5.9.2. Ablaufbeschreibung

Die Applikation prüft das Kommando formal (s. §5.1.1).

Der weitere Ablauf des Kommandos ist identisch zu dem in [VDVNM] spezifizierten.

5.10. EXTERNAL AUTHENTICATE

EXTERNAL AUTHENTICATE wird im Rahmen einer gegenseitigen Authentisierung als Kommando zur Authentisierung der externen Welt (SAM) gegenüber dem Tag verwendet. Das Ergebnis einer erfolgreichen gegenseitigen Authentisierung mit asymmetrischen Schlüsseln sind zwei Sessionkeys SKc und SKi mit einer Länge von jeweils 16 Byte sowie ein Send Sequence Counter SSC (8 Byte), die zur Absicherung der weiteren Kommunikation zwischen der Applikation und externer Welt verwendet werden.

5.10.1.1. Kommando- und Antwortnachricht

Kommando APDU

Die Kommandonachricht des Kommandos INTERNAL AUTHENTICATE hat folgenden Aufbau:

Position	Länge	Wert	Beschreibung
1	1	'00'	CLA
2	1	'82'	INS, Instruction Code
3	1	'00'	P1
4	1	'00'	P2 Schlüssel bekannt
5	1	'XX'	Lc Länge der Kommandodaten
6	X	''XX ... XX''	Enc.SAM (s. §[VDVSAM])

Antwort APDU

Wurde das Kommando erfolgreich ausgeführt, wird der Returncode '90 00' zurückgegeben:

Position	Länge	Wert	Beschreibung
1	2	'90 00'	Returncode SW1 SW2

5.10.2. Ablaufbeschreibung

Die Applikation prüft das Kommando formal (s. §5.1.1).

Der weitere Ablauf des Kommandos ist identisch zu dem in [VDVNM] spezifizierten.

6 Ergänzungskommandos des VDV-KA Nutzermediums

6.1. VERIFY SERVICE-ID

Mit dem Ergänzungskommando VERIFY SERVICE-ID prüft das Nutzermedium die von einem Tag erstellte Signatur über die Service-Quittung (s. §3.6) und Zusatzdaten (s. §5.6). Anschließend werden die Daten zur späteren Verwendung nicht-flüchtig im Nutzermedium abgelegt. Da die Länge der Kommandodaten 255 Byte überschreitet, wird das Kommando im verketteten Modus ausgeführt.

6.1.1. Kommando- und Antwortnachricht

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'X0'	Verketteter Modus: '00' einfacher oder letzter Kommandoaufruf '10' erster bis vorletzter Kommandoaufruf
2	1	'56'	INS
3	1	'00'	P1
4	1	'00'	P2
5	1	'XX'	Lc
6	*variable*	'XX … XX'	Kommandodaten

Kommandodaten

Position	Länge	Wert	Beschreibung
1	1	'81'	Tag Zusatzdaten
2	1	'XX'	Länge Zusatzdaten (max. 64)
3	*variabel*	'XX ... XX'	Zusatzdaten
4	1	'E2'	Service-Quittung
5	1	'1A'	Länge Service-Quittung
6	26	'XX ... XX'	Quittungsdaten
7	2	'7F 21'	Tag CV-Zertifikat Tag
8	2	'81 C6'	Länge Zertifikat
9	198	'XX …XX'	Zertifikatsdaten
10	1	'9E'	Tag Signatur
11	2	'81 80'	Länge Signatur
12	128	'XX …XX'	Signatur

Antwort APDU

Nach erfolgreicher Ausführung gibt das Kommando den Returncode '90 00' aus.

Position	Länge	Wert	Beschreibung
2	2	'90 00'	Returncode

6.1.2. Ablaufbeschreibung

Die Applikation prüft das Kommando formal (s. §5.1.1). Fehlt ein Datenobjekt oder sind die Datenobjekte fehlerhaft codiert, so wird das Kommando mit dem Returncode '6A 80' abgebrochen.

Die verkettete Ausführung des Kommandos wird analog zu [VDVNM] behandelt. Dies gilt insbesondere für die Behandlung der Kommandoreihenfolge und die Kriterien zur Unterbrechung der Kommandokette.

Handelt es sich um den ersten Kommandoaufruf bis zum vorletzten Kommandoaufruf einer Kommandokette, so speichert die Applikation die Kommandodaten flüchtig und gibt den Returncode '90 00' aus.

Ist das Kommando das letzte oder einzige Kommando, so werden die Kommandodaten flüchtig gespeichert.

Die Applikation prüft die Kommandodaten formal. Fehlen Datenobjekte oder werden falsche Längenangaben festgestellt, wird das Kommando mit dem Returncode '6A 80' abgebrochen.

Die Applikation prüft, ob eine mit GET CHALLENGE erzeugte, ungenutzte Zufallszahl C.NM vorliegt. Ist dies nicht der Fall, wird das Kommando mit dem Returncode '6A 88' abgebrochen.

Anschließend wird das in den Kommandodaten übergebene Tag-Zertifikat geprüft. Die Prüfung erfolgt analog zu der in [VDVNM] spezifizierten Prüfung für CV-Zertifikate. Schlägt die Prüfung fehl, wird das Kommando mit einem entsprechenden Fehlercode abgebrochen.

Die Applikation prüft nun mit dem aus dem Zertifikat gewonnenen PuK.Tag die in den Kommandodaten übergebene Signatur SIG über die Daten '81 LL XX ... XX' | 'E2 LL XX ... XX' | C.NM. Anschließend wird die Zufallszahl C.NM gelöscht. Schlägt die Signaturprüfung fehl, wird das Kommando mit dem Returncode '63 CF' abgebrochen.

Die Applikation erhöht nun den im Kontext der Applikation eindeutigen, nicht-flüchtigen Tag Verification Counter (TVC) um 1 und speichert die in den Kommandodaten empfangene Service-Quittung und Zusatzdaten nicht-flüchtig. Anschließend wird die Kommandoausführung mit dem Returncode '90 00' beendet.

6.2. GET SERVICE-ID

Mit dem Kommando GET SERVICE-ID werden die zuletzt im Nutzermedium erfolgreich verifizierte Service-Quittung (s. §3.6) und Zusatzdaten (s. §5.6) gesichert aus dem Nutzermedium ausgelesen. Zusätzlich zu den Quittungsdaten wird der Wert des TVC in die Ausgabedaten einbezogen, um die Mehrfachnutzung der gespeicherten Quittungsdaten erkennen zu können.

6.2.1. Kommando- und Antwortnachricht

Kommando APDU

Position	Länge	Wert	Beschreibung
1	1	'00'	CLA
2	1	'58'	INS
3	1	'00'	P1
4	1	'00'	P2
5	1	'08'	Lc
6	8	'XX … XX'	C.Extern – Zufallszahl der externen Welt
7	1	'00'	Antwortdaten erwartet

Antwortdaten

Position	Länge	Wert	Beschreibung
1	1	'81'	Tag Zusatzdaten
2	1	'XX'	Länge Zusatzdaten (max. 64)
3	*variabel*	'XX ... XX'	Zusatzdaten
4	1	'82'	Tag TVC
5	1	'04'	Länge TVC
6	4	'XX … XX'	Aktueller Wert des TVC
7	1	'E2'	Service-Quittung
8	1	'1A'	Länge Service-Quittung
9	26	'XX ... XX'	Quittungsdaten
10	1	'9E'	Tag Signatur
11	2	'81 80'	Länge Signatur
12	128	'XX …XX'	Signatur über die Datenobjekte '81', '82' und 'E2'

Antwort APDU

Nach erfolgreicher Ausführung gibt das Kommando den Returncode '90 00' aus.

Position	Länge	Wert	Beschreibung
2	2	'90 00'	Returncode

6.2.2. Ablaufbeschreibung

Die Applikation prüft das Kommando formal (s. §5.1.1). Fehlt ein Datenobjekt oder sind die Datenobjekte fehlerhaft codiert, wird das Kommando mit dem Returncode '6A 80' abgebrochen.

Sind Service-Quittung und Zusatzdaten nicht im Kontext der Applikation gespeichert, wird das Kommando mit dem Returncode '6A 88' abgebrochen.

Die Applikation stellt die Daten für die Signaturberechnung aus der in den Kommandodaten übergebenen Zufallszahl C.Extern und den im Kontext der Applikation nicht flüchtig gespeicherten Service-Quittung, Zusatzdaten sowie dem TVC zusammen.

MSG := '81 LL XX ... XX' | '82 04' TVC | 'E2 LL XX ... XX' | C.Extern

Anschließend berechnet die Applikation die Signatur SIG über MSG mit dem Verfahren RSASSA-PSS gemäß PKCS#1 (s. auch Anhang A zu [VDVSAM]) unter Verwendung des Hash-Algorithmus SHA-1.

SIG := sign(PrK.NM)[MSG]

Nach erfolgreicher Signaturberechnung werden die Daten MSG | SIG gefolgt vom Returncode '90 00' in den Antwortdaten ausgegeben.

7 Appendix

7.1. Glossar

Abkürzung	Bedeutung
CHR	Certificate Holder Reference
KA	Kernapplikation
VDV	Verband Deutscher Verkehrsunternehmen
VDV-KA KG	VDV Kernapplikations GmbH & Co. KG
ROM	Read Only Memory

7.2. Referenzen

[VDVSAM]	VDV-Kernapplikation: KA SPEC-SAM, V1.107
[VDVNM]	VDV-Kernapplikation: KA SPEC-Nutzermedium, V1.107
[KASEC]	VDV-Kernapplikation Systemlastenheft – Anforderungen an das Nutzermedium, V1.107
[GP]	GlobalPlatform Card Specification Version 2.1