santésuisse | Versichertenkarte
Carte d'assuré
Tessera d'assicurato

Santésuisse - Versichertenkarte

Detailspezifikation

Version 1.4ech

Implementierungsanleitung der Versichertenkarte basierend auf VVK 832.105 und eCH-0064

Freigabe SASIS AG: 09.08.2010

Inhaltsverzeichnis

1 Kommunikationsprotokoll

1.1 T=1

Für die Versichertenkarte wird das Kommunikationsprotokoll T=1 nach [7816-3] verwendet. Hierfür sind im Besonderen folgende Kommunikations- und Protokollparameter festgelegt:

- PPS: unterstützt
- Chaining: unterstützt
- S(WTX): unterstützt
- S(Abort): darf nur von der ICC genutzt werden
- NAD: 00h
- IFSC (TA3): min. 128 Bytes
- IFSD: 254 Bytes
- FI/DI (TA1): 13h, ebenfalls müssen FI/DI-Werte 12h und 18h unterstützt werden

Aus den angegebenen FI und DI im ATR erhält man eine Übertragungsgeschwindigkeit von 38400 bps. 12h und 18h entsprechen 19200 und 115200 bps.

1.2 ATR

Der ATR (Answer to Reset) ist im Rahmen von ISO/IEC 7816-4 und ISO/IEC 7816-3 definiert - ist aber von den Chipkartenbetriebssystemtypen abhängig und verschieden.
Die Parameter IFSC, FI und DI sind im ATR in den Bytes TA2 und TA1 kodiert. ([7816-3])
In den Historical Data stehen mindestens die Objekte:

- 31 F8: Applikationsanwahl mit voller AID oder Teil-AID. EF.DIR und EF.ATR verfügbar und mit READ BINARY auslesbar.
- 6x …: Betriebssystembezeichnung. („xxxxx“ hh hh)
- 81 xx: LCS-Byte. 07 bei aktivem Chip. 0F bei terminiertem Chip. Im terminierten Zustand reagiert der Chip auf keine APDU.

Daraus ergibt sich ein [7816-3]-konformer ATR:

Bezeichnung	Wert	Beschreibung
TS	3Bh	direct convention
T0	9Fh	TD1, TA1 vorhanden. x Bytes Historical Data vorhanden.
TA1	13h	FI / DI => 38400 bps möglich
TD1	81h	TD2 vorhnaden. T=1-Protokoll.
TD2	B1h	TD3, TB3, TA3 vorhanden. T=1-Protokoll.
TA3	80h	IFSC 128 Bytes
TB3	37h	BWI=3 / CWI=7
TD3	1Fh	TA4 vorhanden. „T=15“-Protokoll
TA4	03h bei MTCOS auf ST: 03h	Clock stop / Class indicator: Clock stop nicht unterstützt. Spannungsklasse A und B.
Historical Bytes	80h	Historical Bytes enthalten COMPACT-TLV-kodierte Daten. Ein Statusindikator-Objekt kann vorhanden sein.
	31h F8h	Card Service Data: Applikationsanwahl mit voller AID oder Teil-AID. EF.DIR und EF.ATR vorhanden und mit READ BINARY auslesbar.
	6xh …	Card Issuer Data: Betriebssystembezeichnung und

Bezeichnung	Wert	Beschreibung
		Version.
	81h xxh	xx: Card Life Cycle Status (LCS): 07h bei aktivem Chip, 0Fh bei terminiertem Chip.

1.3 APDU-Format

Die Versichertenkarte unterstützt alle sieben APDU-Klassen, also sowohl Short APDU als auch Extended APDU. Hierfür steht ein APDU-Puffer von 1033 Bytes (Header, Lc, Le und 1024 Bytes Daten) zur Verfügung. Genaue Angaben stehen in EF.ATR. Siehe auch [7816-4] 5.1 und Amd. 2.

2 Dateisystem

Das Dateisystem enthält die im eCH-0064 definierten Dateien der Kartenanwendungen. Im Masterfile (MF) sind die allgemeinen den Versicherten oder die Versicherungskarte betreffenden Daten enthalten. Dies umfasst zum Beispiel die Administrationsschlüssel, die Kartenhalter-PIN und die Schlüssel und Zertifikate für die Card-to-Card-Authentifizierung. Im Dedicated File DF.NOT stehen die Notfall-Datenkategorien. Das Dedicated File DF.PKCS#15 ist für kantonale Modellversuche gedacht und enthält RSA-Schlüssel und Container für X.509-Zertifikate für Webportalauthentifizierungen.

3 Übersicht File Identifier

File Identifier	Dateiname: Hauptverzeichnis
[3F00]	MF
[3F00 2F00]	EF.DIR
[3F00 2F01]	EF.ATR
[3F00 2F05]	EF.ICCSN
[3F00 0011]	iEF.PIN1
[3F00 0012]	iEF.PIN2 für iEF.PrK.X509, iEF.PrK.Dec, 5 Versuche
[3F00 0014]	iEF.PUK
[3F00 2F06]	EF.ID
[3F00 2F07]	EF.AD
[3F00 5600]	EF.VERSION
[3F00 2F03]	EF.CVC.PDC
[3F00 2F08]	EF.CVC.CA_ORG_PDC
[3F00 0015]	iEF.PrK.SB (nur für Internal Authentication)
[3F00 2F04]	EF.CVC.CA_ROOT_VK
[3F00 001C]	EF.PuK.CA_ROOT_VK
[3F00 001D]	iEF.C2CSTATE
[3F00 0001]	EF.GPKeys

File Identifier	Dateiname: Notfalldaten
[3F00 DF01]	DF.NOT (D7 56 83 21 05 00)
[3F00 DF01 1F01]	EF.BGTD
[3F00 DF01 1F02]	EF.IMMD
[3F00 DF01 1F03]	EF.TPLD
[3F00 DF01 1F04]	EF.KHUF
[3F00 DF01 1F05]	EF.ZUSE
[3F00 DF01 1F06]	EF.MEDI
[3F00 DF01 1F07]	EF.ALLG
[3F00 DF01 1F08]	EF.ADDR
[3F00 DF01 1F09]	EF.VERF

File Identifier	Dateiname: PKCS#15 bzw. ISO/IEC 7816-15
[3F00 DF02]	DF.PKCS#15 aka DF.CIA (A0 00 00 00 63 50 4B 43 53 2D 31 35)
[3F00 DF02 5032]	EF.CIAInfo aka EF.TokenInfo
[3F00 DF02 5031]	EF.OD aka EF.ODF
[3F00 DF02 1F01]	EF.PrKD aka EF.PrKDF
[3F00 DF02 1F02]	EF.PuKD aka EF.PuKDF
[3F00 DF02 1F03]	EF.CD aka EF.CDF
[3F00 DF02 1F04]	EF.DCOD
[3F00 DF02 1F05]	EF.AOD aka EF.AODF
[3F00 DF02 1F06]	EF.CERT
[3F00 DF02 1F07]	EF.PuK.DEC
[3F00 DF02 1F08]	EF.PuK.X509
[3F00 DF02 0016]	iEF.PrK.DEC
[3F00 DF02 0017]	iEF.PrK.X509

4 Hauptverzeichnis

4.1 Master File (MF)

Das Root-Verzeichnis, wird implizit nach einem Reset der Chipkarte vom Betriebssystem selektiert. In ihm befinden sich die anderen Verzeichnisse und Dateien. Das Master File ist ein Sonderfall eines Dedicated File und stellt den gesamten in der Chipkarte für den Dateibereich verfügbaren Speicher dar und ist in jeder Chipkarte vorhanden.

Objekt	**MF**	**Wurzelverzeichnis**
Objekttyp	Deditcated File	
FID	'3F00'	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always

4.2 EF.DIR

EF.DIR enthält die Anwendungsvorlagen gemäss [ISO 7816-4] für die in der Versichertenkarrte vorhandenen Anwendungen. Diese Datei ist für die Selektion der Anwendungen von kantonalen Modellversuchen besonders geeignet. EF.DIR weist eine transparente Struktur auf, die zusammen mit dem EF.ATR in den Historical Bytes angezeigt und objekttypengleich implementiert werden muss (Card Service Data [ISO/IEC 7816-4] 8.1.1.2.3).

Objekt	**EF.DIR**	**Directory File**
Objekttyp	Transparent	
Dateikategorie	Working	
FID	'2F00'	
SFID	'F0'	
Grösse	84 Bytes	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ BINARY	Always

Datenobjekte aus eCH-0064 im EF.DIR

Data-Object			**DO.DIR**	
		L	61-L-{4F-L-V-50-L-V-51-L-V}	
61		25	Variable [V]	
	4F	6	D7 56 83 21 05 00	National AID: DF.NOT
	50	9	Emergency	Application Label
	51	4	3F 00 DF 01	Ref. to DF
		L	61-L-{4F-L-V-50-L-V-51-L-V}	
61		29	Variable [V]	
	4F	12	A0 00 00 00 63 50 4B 43 53 2D 31 35	AID PKCS#15
	50	7	PKCS-15	Application Label
	51	4	3F 00 DF 02	Ref. to DF

4.3 EF.ATR

Die transparente Datei EF.ATR enthält Datenobjekte zur Identifizierung der Karte und ein Datenobjekt zur Anzeige der Grösse der Ein-/Ausgabe-Puffer. Im Datenobjekt Ein-/Ausgabe-Puffergrössen mit dem Tag 'E0' sind in den vier eingebetteten Datenobjekten mit Tag '02' die Anzahl der Bytes der APDU abgelegt. Weitere Informationen über die Karte in im Datenobjekt mit dem Tag '66' [ISO 7816-6; Card Data] enthalten.

Objekt	**EF.ATR**	**Answer to Reset Datei**
Objekttyp	Transparent	
Dateikategorie	Working	
FID	'2F01'	
SFID	'01'	
Grösse	43 Bytes	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ BINARY	Always

Datenobjekte im EF.ATR

Data-Object: DO.ATR				**Meaning**
T	T	L	E0-L-{02-L-V-02-L-V-02-L-V-02-L-V}-66-L-{46-L-V}	
E0			Variable [V]	
	02	2	04 00h	[Bytes]: 1024, C-APDU
	02	2	04 00h	[Bytes]: 1024, R-APDU
	02	2	04 00h	[Bytes]: 1024, SM-C-APDU
	02	2	04 00h	[Bytes]: 1024, SM-R-APDU
66				
	46	21	02 Inter1 .. .. .. 4d 54 43 4f 53 20 70 20 32 2e 31	- Chip producer ID[1] - Card Manufacturer [9] - MTCOS p 2.1 [11]

4.4 EF.ICCSN

Objekt	**EF.ICCSN**	**Versichertenkarten-Identifikationsdatei**
Objekttyp	Linear variable	
Dateikategorie	Working	
FID	'2F05'	
SFID	'05'	
Grösse	45 Bytes	
Anzahl Records	3	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ RECORD	Always

Daten im ICCSN

Data-Object:	DO.ICCSN
Record Nr.	**Data**
1	5A 0A ‖ ICCSN (10 Oktetts); Kennnummer der Versichertenkarte
2	Reference Number (8 Oktetts); Spezifiziert durch Versicherer
3	Generalized Time: YYYYMMDDHHMMZ (13 Oktetts) [ISO 8601:2004]

4.5 iEF.PIN1 [PIN-Schutz für Notfalldaten]

Bei der Kartenausgabe wird das Aktivierungsbyte auf 0 gesetzt, d.h. auf die Notfalldaten kann ohne PIN-Schutz zugegriffen werden. Wenn eine Person eine im eCH-0064 mit (*) bezeichnete Datei durch einen PIN-Code schützen will, wird das Aktivierungsbyte auf 1 gesetzt.

Objekt	**iEF.PIN1**	**Cardholder Verification File**			
Objekttyp	Linear fixed (abhängig vom Chipkartenbetriebssystemhersteller)				
Dateikategorie	Internal				
FID	'0011'				
SFID	'11'				
Grösse	10 Bytes				
Zugriffsrechte	**Zugriffsart [AM]**			**Sicherheitsbedingungen [SC]**	
	SELECT			Always	
	VERIFY			Always	
	CHANGE REFERENCE DATA			with PIN 1	
	RESET RETRY COUNTER			with PUK	
	DISABLE VERIFICATION REQUIREMENT			Always	
	ENABLE VERIFICATION REQUIREMENT			Always	
Aktivierungs-Byte	PIN-Identifier	PIN-Value	Pre-set attempt value	Unblocking PIN Value	Number of unblocking mechanisms
0/1; no / yes	01h	6-8 Ziffern	5 Versuche	8 Ziffern	10 Versuche

4.6 iEF.PIN2 [PIN-Schutz für Kantonale Modellversuche]

Dieser PIN2 kann ausschliesslich nur für kantonale Modellversuche verwendet werden. Die Versicherer liefern die Versichertenkarte mit einem Zufallswert von 8 Ziffern aus. Dieser Zufallswert wird dem Versicherten nicht mitgeteilt und auch nicht anderswo gespeichert. Durch die Eingabe eines korrekten PUKs kann der Versicherte den PIN2 neu setzen. Dies ist eine technologisch notwendige Bedingung für einen Versicherten, bevor er an einem kantonalen Versuch teilnehmen kann.

Objekt	iEF.PIN2	PIN2 für Kantonale Modellversuche		
Objekttyp	Linear fixed (abhängig vom Chipkartenbetriebssystemhersteller)			
Dateikategorie	Internal			
FID	'0012'			
SFID	'12'			
Grösse	10 Bytes			
Zugriffsrechte	**Zugriffsart [AM]**		**Sicherheitsbedingungen [SC]**	
	SELECT		Always	
	VERIFY		Always	
	CHANGE REFERENCE DATA		mit PIN2 oder PUK	
	RESET RETRY COUNTER		mit PUK	
PIN-Identifier	PIN-Value	Pre-set attempt value	Unblocking PIN Value	Number of unblocking mechanisms
02h	8 Ziffern	5 Versuche	8 Ziffern	10 Versuche

4.7 iEF.PUK [Personal Unblocking Key]

Nach Eingabe eines PUK (Personal Unblocking Key), der dem Versicherten bekannt ist, kann der Fehlbedienungszähler von PIN1 zurückgesetzt, eine neue PIN2 definiert sowie der Fehlbedienungszähler für PIN2 zurückgesetzt werden. Die Versicherer erzeugen einen zufälligen PUK-Kode, welcher auf der Karte enthalten ist und geben den PUK dem Versicherten bei der Auslieferung der Karte bekannt.

Objekt	iEF.PUK	Personal Unblocking Key	
Objekttyp	Linear fixed (abhängig vom Chipkartenbetriebssystemhersteller)		
Dateikategorie	Internal		
FID	'0014'		
SFID	'14'		
Grösse	10 Bytes		
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**	
	SELECT	Always	
	VERIFY	Always	
PUK-Identifier	PUK-Value	Pre-set attempt value	
04h	8 Ziffern	10 Versuche	

4.8 EF.ID

Die Identifikationsdaten beruhen auf den "Technischen und grafischen Anforderungen an die Versicherten-karte für die obligatorische Krankenpflegeversicherung" VVK-EDI, dem Standard ISO 21549-5:2006 und den Kodierungsvorschriften nach ISO/IEC 8825-1:2002 (BER-TLV).

Objekt	**EF.ID**	**Identification Data**
Objekttyp	Transparent	
Dateikategorie	Working	
FID	'2F06'	
SFID	'06'	
Grösse	84 Bytes	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ BINARY	Always
	GET DATA	Always

Daten im EF.ID

Data-Object			**DO.ID**	
		L	65-L-{80-L-V-82-L-V-83-L-V-84-L-V}	
65			Variable [V]	
	80	50	Name (21), Given Name (21)	UTF8InternationalString
	82	8	DateOfBirth: yyyymmdd	NUMERIC STRING
	83	13	cardholderIdentifier: Versichertennummer der AHV	UTF8InternationalString
	84	1	sex: Geschlecht, 1=male, 2=female, 0=not known, 9=not appl.	ENUMERATED

Beispiel:

65	**Tag**	**Value**	**encoded**
	80	Näf, Jörg	'4E C3A4 66 2C 20 4A C3B6 72 67'
	82	19290918	'31 39 32 39 30 39 31 38'
	83	7569999999939	'37 35 36 39 39 39 39 39 39 39 39 33 39'
	84	1	'01'

Format:

- UTF8InternationalString ::= UTF8String (FROM (BasicLatin UNION Latin-1Supplement))
- EUMERATED::= Octet (01, 02, 00, 09)
- NUMERIC STRING::= UTF8String FROM BasicLatin (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

4.9 EF.AD

Die Administrativen Daten beruhen auf den "Technischen und grafischen Anforderungen an die Versichertenkarte für die obligatorische Krankenpflegeversicherung" VVK-EDI, dem Standard ISO 21549-6:2006 und den Kodierungsvorschriften nach ISO/IEC 8825-1:2002 (BER-TLV).

Objekt	**EF.AD**	**Administrative Data**
Objekttyp	Transparent	
Dateikategorie	Working	
FID	'2F07'	
SFID	'07'	
Grösse	95 Bytes	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ BINARY	Always
	GET DATA	Always

Daten im EF.AD

Data-Object			**DO.AD**	
		L	65-L-{90-L-V-91-L-V-92-L-V-93-L-V-94-L-V}	
65			Variable [V]	
	90	2	IssuingStateIDNumber: ID des herausgebenden Staates	UTF8InternationalString
	91	50	nameOfTheInstitution: Name des Versicherers	UTF8InternationalString
	92	5	identificationNumberOfTheInstitution: BAG-Nummer des Versicherers	NUMERIC STRING
	93	20	InsuredPersonNumber: Kennnummer der Versichertenkarte	UTF8InternationalString
	94	8	ExpiryDate: Ablaufdatum: yyyymmdd	NUMERIC STRING

Beispiel:

65	**Tag**	**Value**
	90	CH
	91	Testversicherer
	92	01234
	93	80756012340000000065
	94	20140630

Format:

- UTF8InternationalString ::= UTF8String (FROM (BasicLatin UNION Latin-1Supplement))
- EUMERATED::= Octet (01, 02, 00, 09)
- NUMERIC STRING::= UTF8String FROM BasicLatin (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

4.10 EF.VERSION

Das System PDC-HPC wird nicht ewig mit den gleichen Kartenversionen auskommen. Auf der FMH-HPC wurde ein EF.VERSION vorbereitet. Dieses Versionsfeld ist 4 Byte lang und hat den File-Identifier 0x5600. Gemäss einer schriftlichen Absprache ist dieses Feld mit dem FID: '5600' von allen Anbietern von HPCs und PDCs zu übernehmen. Damit kann der Aufwand bei der Umsetzung einer Middleware reduziert werden.

Objekt	EF.VERSION	Versionsdatei
Objekttyp	Transparent	
Dateikategorie	Working	
FID	'5600'	
SFID	'B0'	
Grösse	4 Bytes	
Zugriffsrechte	Zugriffsart [AM]	Sicherheitsbedingungen [SC]
	SELECT	Always
	READ BINARY	Always

Daten in EF.VERSION

FMH	Kodierung
FID	'5600'
Kürzel "FMH"	0x46 0x4D 0x48
Version 00	0x00 (mit MSB 0 für HPC)
DO.Version	0x46 0x4D 0x48 0x00

SASIS	Kodierung
FID	'5600'
Kürzel "SAS"	0x53 0x41 0x53
Version 00	0x80 (mit MSB 1 für PDC)
DO.Version	0x53 0x41 0x53 0x80

4.11 EF.CVC.PDC

Die Datei EF.CVC.PDC enthält das CVC-Zertifikat der Versichertenkarte

Objekt	EF.CVC.PDC	Datei für CVC-Zertifikat
Objekttyp	Transparent	
Dateikategorie	Working	
FID	'2F03'	
SFID	'03'	
Grösse	618 Bytes	
Zugriffsrechte	Zugriffsart [AM]	Sicherheitsbedingungen [SC]
	SELECT	Always
	READ BINARY	Always

4.12 EF.CVC.CA_ORG_PDC

Die Datei EF.CVC.CA_ORG_PDC enthält das CVC-Zertifikat der Versicherer (SubCA).

Objekt	EF.CVC.CA_ORG_PDC	Datei für CVC-Zertifikat
Objekttyp	Transparent	
Dateikategorie	Working	
FID	'2F08'	
SFID	'08'	
Grösse	624 Bytes	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ BINARY	Always

4.13 EF.CVC.CA_ROOT_VK

Die Datei EF.CVC.CA_ROOT_VK enthält das CVC-Zertifikat der neutralen Root-CA.

Objekt	EF.CVC.CA_ROOT_VK	Datei für CVC-Zertifikat
Objekttyp	Transparent	
Dateikategorie	Working	
FID	'2F04'	
SFID	'04'	
Grösse	624 Bytes	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ BINARY	Always

4.14 EF.PuK.CA_ROOT_VK

Für das C2C-Authentisierungsverfahren auf der Basis von CV-Zertifikaten wird ein globaler öffentlicher Root-Schlüssel PuK.CA_ROOT_VK benötigt, welcher auf der entsprechenden Offline-CVC-PKI erzeugt wurde und in einer Schlüssel-Datei unterhalb des MF abgelegt ist. Für Testzwecke beim Produktionsprozess wurde dieser öffentliche Schlüssel als auslesbar (working) angelegt. Die Schlüsseldateien und ihre Datenformatierungen-/Kodierungen sind stark vom Chipkartenbetriebssystem abhängig und schwer standardisierbar. Deshalb sollen die öffentlichen Schlüssel immer aus den standardisierten CVC-Zertifikaten bzw. den X.509-Zertifikaten ausgelesen werden.

Objekt	EF.PuK.CA_ROOT_VK	Public Root Key für asymmetrische Authentifizierung
Objekttyp	Linear fixed	
Dateikategorie	Working	
FID	'001C'	
SFID	'1C'	
Grösse	299 Bytes	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	MSE	Always
	READ RECORD	Always

4.15 iEF.PrK.SB

Für das Card-to-Card-Authentisierungsverfahren auf der Basis von CV-Zertifikaten wird ein globaler privater Schlüssel S_B benötigt, der in einer Schlüssel-Datei unterhalb des MF abgelegt ist. Der zugehörige öffentliche Schlüssel ist im Zertifikat CVC.PDC integriert, welches sich im Container EF.CVC.PDC befindet.

Objekt	**iEF.PrK.SB**	**Private Key für asymmetrische Authentisierung**
Objekttyp	Linear variable	
Dateikategorie	Internal	
FID	'0015'	
SFID	'15'	
Grösse	1224 Bytes	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	INTERNAL AUTHENTICATE	Always

4.16 iEF.C2CSTATE

Interne, chipkartenbetriebssystemspezifische Statusdatei. Wird als Zwischenspeicher für die Gültigkeitsüberprüfung bezüglich Ablaufdatum von CVC-Zertifikaten benutzt.

Objekt	**iEF.C2CSTATE**	**Chipkartenbetriebssystemspezifische Statusdatei**
Objekttyp	Transparent	
Dateikategorie	Internal	
FID	' 001D'	
SFID	'1D'	
Grösse	6 Bytes	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	PSO	Always

4.17 EF.GPKeys

Diese interne Datei wurde für die Prä-Personalisierung und für die Personalisierung verwendet und enthält 6 Schlüssel bzw. 2 Schlüsseltriplets. Die Versichertenkarte wurde mittels den Verfahren nach dem sicherheitstechnisch bewährten internationalen Standard "GlobalPlatform, Card Specification, Version 2.2" personalisiert. Nach abgeschlossener Personalisierung wird diese Datei terminiert, d.h. die Datei wird unbrauchbar gemacht. Die Versichertenkarte bleibt danach bezüglich Ihren Dateistrukturen, schreibgeschützten Daten und Funktionen während der ganzen Lebenszeit unveränderbar geschützt. Diese doppelte Sicherheit (Schlüssel + terminierte Datei) erhöht die Sicherheit wesentlich.

Objekt	**EF.GPKeys**	**Terminierte Datei für GP 2.2 Personalisierungsprozess**
Objekttyp	Linear fixed	
Dateikategorie	Working	
FID	' 0001'	
SFID	'01'	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	No Access	Always

5 Daten nach Artikel 42*a* Absatz 4 KVG (Notfalldaten)

5.1 Zugriffsregeln

Entität	CHA Profil-ID	CHA ['5F4C']	Schlüssel 1 [eCH-0064]	Schlüssel 2 [eCH-0064]	Schlüssel 3 [eCH-0064]	Beschrieb
CA_ROOT_VK	CHA_0	00h				Kein Zugriff auf Daten
CA_ORG_HPC	CHA_0	00h				Kein Zugriff auf Daten
CA_ORG_PDC	CHA_0	00h				Kein Zugriff auf Daten
PDC	CHA_0	00h				Kein Zugriff auf Daten
Ärzte	CHA_1	01h	x	x		Zugriff auf Notfalldaten
Apotheker	CHA_2	02h	x		x	Zugriff auf Notfalldaten
Zahnärzte	CHA_3	03h	x	x		Zugriff auf Notfalldaten
Chiropraktoren	CHA_4	04h	x	x		Zugriff auf Notfalldaten
Hebammen	CHA_5	05h	x			Zugriff auf Notfalldaten
Physiotherapeuten	CHA_6	06h	x			Zugriff auf Notfalldaten
Ergotherapeuten	CHA_7	07h	x			Zugriff auf Notfalldaten
Pflegefachleute	CHA_8	08h	x			Zugriff auf Notfalldaten
Logopäden	CHA_9	09h	x			Zugriff auf Notfalldaten
Ernährungsberater	CHA_{10}	10h	x			Zugriff auf Notfalldaten

5.2 Dateien

5.2.1 Verzeichnisdatei für Notfalldaten

Objekt	**DF.NOT**	**Verzeichnis für Notfalldaten**
Objekttyp	Dedicated File	
FID	'DF01'	
AID	'D7 56 83 21 05 00'	
Grösse	23615 Bytes	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always

5.2.2 Blutgruppen- und Transfusionsdaten

Objekt	EF.BGTD	Blutgruppen- und Transfusionsdaten
Objekttyp	Elementary File	
Dateistruktur	Transparent	
Dateikategorie	Working	
FID	'1F01'	
SFID	'01'	
Grösse	302 Bytes	
max. Rec. Länge	300 Bytes	
Anzahl Records	1	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ BINARY	Schlüssel 1 + PIN 1
	GET DATA	Schlüssel 1 + PIN 1
	UPDATE BINARY	Schlüssel 2 + PIN 1
	ENABLE VERIFICATION REQUIREMENT	Always
	DISABLE VERIFICATION REQUIREMENT	Always

5.2.3 Immunisierungsdaten

Objekt	EF.IMMD	Immunisierungsdaten
Objekttyp	Elementary File	
Dateistruktur	Linear variable	
Dateikategorie	Working	
FID	'1F02'	
SFID	'02'	
Grösse	5628 Bytes	
max. Rec. Länge	263 Bytes	
Anzahl Records	21	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ RECORD	Schlüssel 1 + PIN 1
	UPDATE RECORD	Schlüssel 2 + PIN 1
	APPEND RECORD	Schlüssel 2 + PIN 1
	ENABLE VERIFICATION REQUIREMENT	Always
	DISABLE VERIFICATION REQUIREMENT	Always

5.2.4 Transplantationsdaten

Objekt	EF.TPLD	Transplantationsdaten
Objekttyp	Elementary File	
Dateistruktur	Linear variable	
Dateikategorie	Working	
FID	'1F03'	
SFID	'03'	
Grösse	333 Bytes	
max. Rec. Länge	132/190 Bytes	
Anzahl Records	2/1	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ RECORD	Schlüssel 1 + PIN 1
	UPDATE RECORD	Schlüssel 2 + PIN 1
	APPEND RECORD	Schlüssel 2 + PIN 1
	ENABLE VERIFICATION REQUIREMENT	Always
	DISABLE VERIFICATION REQUIREMENT	Always

5.2.5 Krankheiten und Unfallfolgen

Objekt	EF.KHUF	Krankheiten und Unfallfolgen
Objekttyp	Elementary File	
Dateistruktur	Linear variable	
Dateikategorie	Working	
FID	'1F04'	
SFID	'04'	
Grösse	2820 Bytes	
max. Rec. Länge	136 Bytes	
Anzahl Records	20	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ RECORD	Schlüssel 1 + PIN 1
	UPDATE RECORD	Schlüssel 2 + PIN 1
	APPEND RECORD	Schlüssel 2 + PIN 1
	ENABLE VERIFICATION REQUIREMENT	Always
	DISABLE VERIFICATION REQUIREMENT	Always

5.2.6 Zusätzliche Einträge

Objekt	EF.ZUSE	Zusätzliche Einträge
Objekttyp	Elementary File	
Dateistruktur	Linear variable	
Dateikategorie	Working	
FID	'1F05'	
SFID	'05'	
Grösse	2622 Bytes	
max. Rec. Länge	500/154 Bytes	
Anzahl Records	5/17	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ RECORD	Schlüssel 1 + PIN 1
	UPDATE RECORD	Schlüssel 2 + PIN 1
	APPEND RECORD	Schlüssel 2 + PIN 1
	ENABLE VERIFICATION REQUIREMENT	Always
	DISABLE VERIFICATION REQUIREMENT	Always

5.2.7 Medikation

Objekt	EF.MEDI	Medikation
Objekttyp	Elementary File	
Dateistruktur	Linear variable	
Dateikategorie	Working	
FID	'1F06'	
SFID	'06'	
Grösse	4968 Bytes	
max. Rec. Länge	270 Bytes	
Anzahl Records	18	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ RECORD	Schlüssel 1 + PIN 1
	UPDATE RECORD	Schlüssel 2 + PIN 1
	APPEND RECORD	Schlüssel 2 + PIN 1
	UPDATE RECORD	Schlüssel 3 + PIN 1
	APPEND RECORD	Schlüssel 3 + PIN 1
	ENABLE VERIFICATION REQUIREMENT	Always
	DISABLE VERIFICATION REQUIREMENT	Always

5.2.8 Allergien

Objekt	EF.ALLG	Allergien
Objekttyp	Elementary File	
Dateistruktur	Linear variable	
Dateikategorie	Working	
FID	'1F07'	
SFID	'07'	
Grösse	4380 Bytes	
max. Rec. Länge	345/350 Bytes	
Anzahl Records	12	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ RECORD	Schlüssel 1 + PIN 1
	UPDATE RECORD	Schlüssel 2 + PIN 1
	APPEND RECORD	Schlüssel 2 + PIN 1
	ENABLE VERIFICATION REQUIREMENT	Always
	DISABLE VERIFICATION REQUIREMENT	Always

5.2.9 Kontaktadressen für den Notfall

Objekt	EF.ADDR	Kontaktadressen
Objekttyp	Elementary File	
Dateistruktur	Linear variable	
Dateikategorie	Working	
FID	'1F08'	
SFID	'08'	
Grösse	844 Bytes	
max. Rec. Länge	374 Bytes	
Anzahl Records	2	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ RECORD	Schlüssel 1
	UPDATE RECORD	Schlüssel 1
	APPEND RECORD	Schlüssel 1

5.2.10 Patientenverfügungen und Organspendeausweise

Objekt	EF.VERF	Patientenverfügungen und Organspendeausweise
Objekttyp	Elementary File	
Dateistruktur	Linear variable	
Dateikategorie	Working	
FID	'1F09'	
SFID	'09'	
Grösse	1048 Bytes	
max. Rec. Länge	476 Bytes	
Anzahl Records	2	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ RECORD	Schlüssel 1
	UPDATE RECORD	Schlüssel 1
	APPEND RECORD	Schlüssel 1

5.3 Zugriffsmechanismen und Bearbeitung auf und von Daten nach Artikel 42*a* Absatz 4 KVG (Notfalldaten)

5.3.1 APDU Kommandos

5.3.1.1 Select mit File Identifier

Dieses Kommando ist kodiert nach [ISO/IEC 7816-4] 7.1.1.

Terminal	**SELECT**					
CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	00h	P2	02h	FID	Ne

P2	Le	Beschreibung
00h	-	Selektion MF, DF oder EF
00h	00h	Selektion MF, DF oder EF; Rückgabe von optionalen FCI
04h	00h	Selektion MF, DF oder EF; Rückgabe von optionalen FCP

5.3.1.2 Select mit DF Name

Dieses Kommando ist kodiert nach [ISO/IEC 7816-4] 7.1.1.

Terminal	**SELECT**					
CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	04h	P2	02h	FID	Ne

P2	Le	Beschreibung
04h	00h	Selektion eines DF mit dem DF Name ; Rückgabe von optionalen FCP

5.3.1.3 Select mit Pfad Name

Dieses Kommando ist kodiert nach [ISO/IEC 7816-4] 7.1.1.

Terminal	**SELECT**					
CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	P1	00h	04h	PATH	Ne

P1	Le	Beschreibung
08h	-	Selektion einer Datei durch die Angabe des FID basierten Pfades vom MF aus
08h	00h	Selektion einer Datei durch die Angabe des FID basierten Pfades vom MF aus; Rückgabe von optionalen FCI
09h	-	Selektion einer Datei durch die Angabe des FID basierten Pfades vom aktuellen DF aus
09h	00h	Selektion einer Datei durch die Angabe des FID basierten Pfades vom aktuellen DF aus; Rückgabe von optionalen FCI

5.3.1.4 Notfall-Datenkategorien

FID	PATH	Daten	Struktur	Card2Card	PIN1-Schutz
'DF01'	-	Verzeichnis Notfalldaten	-	-	-
'1F01'	'DF011F01'	Blutgruppen- und Transfusionsdaten	transparent	ja	ja
'1F02'	'DF011F02'	Immunisierungsdaten	linear	ja	ja
'1F03'	'DF011F03'	Transplantationsdaten	linear	ja	ja
'1F04'	'DF011F04'	Krankheiten und Unfallfolgen	linear	ja	ja
'1F05'	'DF011F05'	Zusätzliche Einträge	linear	ja	ja
'1F06'	'DF011F06'	Medikation	linear	ja	ja
'1F07'	'DF011F07'	Allergien	linear	ja	ja
'1F08'	'DF011F08'	Kontaktadressen für den Notfall	linear	ja	nein
'1F09'	'DF011F09'	Patientenverfügungen und Organspendeausweise	linear	ja	nein

5.3.1.5 Zugriff auf Daten in transparenten EF

Es ist möglich auf Daten in einem transparenten EF lesend (READ BINARY) oder schreibend (UPDATE BINARY) zuzugreifen. Bei im transparenten EF abgelegten Datenobjekten in TLV-Struktur kann mit GET DATA ebenfalls lesend zugegriffen werden.

5.3.1.5.1 Read Binary

Dieses Kommando ist kodiert nach [ISO/IEC 7816-4] 7.2.3.

Terminal	READ BINARY					
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	B0h	00h	00h	-	-	Ne

Auslesen der Datenkategorie EF.BGTD. Das transparente EF.BGTD wird vor der Leseoperation mittels SELECT Kommando ausgewählt. Die entsprechend festgelegten Zugriffsbedingungen müssen vorab erfüllt sein.
Le = 00h bzw. 000000h; Lese alle Daten bis zum Ende der Datei.
Le > 00h; Le ist die Anzahl der zu lesenden Bytes.

5.3.1.5.2 UPDATE BINARY

Dieses Kommando ist kodiert nach [ISO/IEC 7816-4] 7.2.5.

Terminal	UPDATE BINARY					
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	D0h	00h	00h	Lc	Binärer Datenstring der Länge Lc	-

Das Kommando UPDATE BINARY wird zum Schreiben von Daten in der transparenten Datenkategorie EF.BGTD verwendet und überschreibt allfällig bereits vorhandene Daten durch Daten, die im Datenfeld der Kommandonachricht enthalten sind. Lc ist die Anzahl der zu schreibenden Bytes des binären Datenstrings und muss angegeben werden. Das transparente EF.BGTD wird vor der Schreiboperation mittels SELECT Kommando ausgewählt. Die entsprechend festgelegten Zugriffsbedingungen müssen vorab erfüllt sein.

5.3.1.5.3 EREASE BINARY

Dieses APDU Kommando wird nicht unterstützt. Der Löschvorgang soll mittels einem UPDATE BINARY mit einem binären Nullstring der Länge Lc durchgeführt. Dabei werden bereits vorhandene Daten eines transparenten EF durch Oktette mit dem Wert '00' überschrieben.

5.3.1.5.4 GET DATA

Dieses Kommando ist kodiert nach [ISO/IEC 7816-4] 7.4.2.

Terminal		GET DATA				
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	CAh	00h	TAG	Lc	Datenelement	Ne

Als Ergänzung der auf Dateien basierendem Lesekommandos wird auch das Kommando GET DATA unterstützt, welches für den direkten Zugriff auf Datenobjekte gedacht ist. Mit GET DATA lassen sich Datenobjekte lesen. Das Kommando verarbeitet ein oder mehrere TLV-kodierte Datenobjekte. Die entsprechend festgelegten Zugriffsbedingungen müssen vorab erfüllt sein.

TAG:	Gewähltes TAG von Datenelement in Datenobjekt
Lc:	Länge des gewählten Datenelements
Ne:	000000h

5.3.1.6 Zugriff auf strukturierte Daten

Es ist möglich auf Records in einer Recordliste in strukturierten EF lesend (READ RECORD) oder schreibend (UPDATE RECORD) zuzugreifen. Zudem lassen sich neue Records angelegen (APPEND RECORD). Records lassen sich nicht löschen. Es ist aber möglich den Inhalt eines Records zu löschen (ERASE RECORD).

5.3.1.6.1 READ RECORD

Dieses Kommando ist kodiert nach [ISO/IEC 7816-4] 7.3.3.

Terminal		READ RECORD				
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	B2h	Recordnummer	P2	-	-	000000h

Das Kommando READ RECORD dient dem Auslesen eines Records einer Recordliste in einer strukturierten linearen Notfall-Datenkategorie, wobei Ne die Anzahl der erwarteten Oktette in den Antwortdaten ist. Das linear strukturierte EF wird vor der Leseoperation mittels SELECT Kommando ausgewählt. Die entsprechend festgelegten Zugriffsbedingungen müssen vorab erfüllt sein. Grundsätzlich können sämtliche Kommandos, wie sie in ISO/IEC 7816-4 definiert sind, verwendet werden. Sinnvollerweise soll eine Einschränkung, wie in folgender Tabelle definiert, vorgenommen werden:

P1	**P2**								**Bedeutung**
	b8	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	
P1	-	-	-	-	-	1	0	0	Lese den Record mit der Recordnummer, übergeben in P1; Lese alle Bytes bis zum Ende des Records mit extended length
00h	-	-	-	-	-	0	1	0	Lese den nächsten Record mit dem Record Identifier, welcher auf den aktuellen Record zeigt; Lese alle Bytes bis zum Ende des Records mit extended length

5.3.1.7 UPDATE RECORD

Terminal		UPDATE RECORD				
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	DCh	Recordnummer	04h	Länge Record	neue Record-Daten	-

Das Kommando UPDATE RECORD ersetzt den Oktettstring eines bereits vorhandenen Records in der Recordliste in einer strukturierten linearen Notfall-Datenkategorie. Das linear strukturierte EF wird vor der Überschreiboperation mittels SELECT Kommando ausgewählt. Die entsprechend festgelegten Zugriffsbedingungen müssen vorab erfüllt sein.

5.3.1.8 APPEND RECORD

Terminal		APPEND RECORD				
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	E2h	00h	00h	Länge Record	neue Record-Daten	-

Das Kommando APPEND RECORD fügt einen neuen Record in die Recordliste in einer strukturierten linearen Notfall-Datenkategorie an, wobei die Daten für den Oktettstring des neuen Records im Datenfeld der Kommandonachricht enthalten sind. Das betroffene linear strukturierte EF wird vor der Operation mittels SELECT ausgewählt. Die entsprechend festgelegten Zugriffsbedingungen müssen vorab erfüllt sein.

5.3.1.9 EREASE RECORD

Das Kommando ERASE RECORD ersetzt gemäss ISO/IEC 7816-4 (Kap. 7.3.8) den Oktettstring eines bereits vorhandenen Records in der Recordliste eines linear strukturierten EF durch einen Oktettstring, der nur Oktette mit dem Wert '00' besitzt. Dieses Kommando kann mit dem Kommando UPDATE RECORD substituiert werden, wobei die Record-Daten aus einem Oktettstring der Länge Le gebildet wird, welcher nur Oktette mit dem Wert '00' besitzt. Merke: Ein gelöschter Record wird in einem ISO/IEC 7816-4 System definitionsgemäss nie aus dem Speicher entfernt, sondern nur mit Nullen überschrieben.

5.3.2 Speicherverwaltung bei linearen Dateistrukturen nach ISO/IEC 7816-4

Die meisten Notfall-Datenkategorien sind als variable lineare Strukturen angelegt. Beim Anlegen eines neuen Notfall Datenobjektes mit dem APPEND RECORD Kommando wird die im Parameter Lc angegebene Länge des Datenobjektes in der Recordliste persistent eingetragen. Dieser neuer Record kann nie mehr entfernt werden. Ein Löschen beinhaltet gemäss ISO/IEC 7816-4 ein Überschreiben des Records mit einem Oktettstring von lauter Nullen. Aus diesem Grund müssen die Records immer mit der maximal definierten Länge Lc angelegt werden.

5.3.3 Beispiele

Bedingung: Vorangehend erfolgreich durchgeführte Card2Card-Authentisierung.

Daten zu Blutgruppe und Transfusionen auslesen

Terminal	SELECT					
CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	08h	00h	04h	DF011F01	-
Response Data						SW12
-						90 00h

Terminal	READ BINARY					
CLA	INS	P1	P2	Lc	Data	Le
00h	B0h	00h	00h	-	-	000000h
Response Data						SW12
Datenstring TLV-Objekt (EF.BGTD)						90 00h

PIN-Geschützte Daten zu Krankheiten und Unfallfolgen auslesen und überschreiben: Record Nr. 01

Terminal	SELECT					
CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	08h	00h	04h	DF011F04	-
Response Data						SW12
-						90 00h

VERIFY: PIN-Eingabe						
CLA	INS	P1	P2	Lc	Data	Le
00h	20h	00h	01h	08h	PIN1	-
Response Data						SW12
-						90 00h

Terminal	READ RECORD					
CLA	INS	P1	P2	Lc	Data	Le
00h	B2h	01h	04h	-	-	000000h
Response Data						SW12
Record-Daten						90 00h

Terminal	UPDATE RECORD					
CLA	INS	P1	P2	Lc	Data	Le
00h	B2h	01h	04h	88h	Record-Daten	-
Response Data						SW12
-						90 00h

Löschen eines Eintrages in den Daten zu Krankheiten und Unfallfolgen: Record Nr. 5

Terminal		**UPDATE RECORD**				
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	B2h	05h	04h	88h	'00 00 …. 00 00'	-
Response Data						SW12
-						90 00h

Eintrag einer neuen Impfung

Terminal		**SELECT**				
CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	08h	00h	04h	DF011F02	-
Response Data						SW12
-						90 00h

Terminal		**APPEND RECORD**				
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	E2h	00h	00h	0107h	neue Record-Daten	-
Response Data						SW12
-						90 00h

6 Card-to-Card-Authentifizierung und Autorisierung

6.1 Spezifikation CVC-Zertifikate

6.1.1 Grundlagen

6.1.1.1 Normative Grundlagen:

- ISO/IEC 7816-4
- ISO/IEC 7816-6
- ISO/IEC 7816-8
- PKCS#1 v1.5
- Technical Guideline TR-03110, v1.11, BSI, Extended Access Control (EAC)
- Technical Guideline TR-03110, v2.0, BSI, Extended Access Control (EAC)

Padding-Technologie:	EMSA-PKCS-v1_5 [PKCS#1 v1.5]
Zertifikatstyp:	Selbstbeschreibendes Zertifikat in TLV-Struktur, keine Headerliste notwendig
Kodierung:	Distinguished Encoding Rules (DER), nach ISO/IEC 8825-1:2002

6.1.1.2 Implementationstechnologie

- Modernes, nach ISO/IEC 8825-1:2002 DER-codiertes CVC-Zertifikatsformat analog wie bei X.509 Zertifikaten (RFC 5280)
- Signierungstechnologie von CVC-Zertifikaten mit Padding nach PKCS#1 v1.5 gleich wie bei X.509-Zertifikaten für TLS 1.0/SSL 3.0
- Für zukünftig eingesetzte Algorithmen wie elliptische Kurven ECC kann die gleiche CVC-Zertifikatstruktur verwendet werden (EAC). Eine Zertifikatsstruktur mit Padding nach ISO/IEC 9796-2 DS1 ist bei Verfahren mit elliptischen Kurven ECC nicht umsetzbar. Möglichkeit von nahtlosem Übergang von RSA-EMSA-PKCS-v1_5 zu elliptischen Kurven mit Parallelanwendung auf Lesegeräten/Anwendungsmodulen und Versichertenkarten Beginn im Jahre 2013 bis spätestens im Jahre 2015.
- Öffentlicher Schlüssel und Exponent kann direkt aus CVC-Zertifikat ausgelesen werden (Tag 81h bzw. Tag 82h) und damit kein Auslesen des öffentlichen Schlüssels von der Chipkarte, welcher in einer proprietären, vom Chipkartenbetriebssystem abhängigen Datenstruktur innerhalb einer Chipkartendatei (transparent, bzw. linear) abgespeichert ist.
- RSA-Schlüssellänge von 2048 Bit und Hashfunktion SHA-256 für die Signierung von Root- und Sub-CA-Zertifikaten ist kryptografisch genügend stark, auch bis ins Jahr 2018.
- Unterstützung von extended APDUs durch Lesegeräte für das Auslesen der Notfalldaten sowie für den Webportalzugriff mittels SSL/TLS bei kantonalen Versuchen mit RSA-Schlüsseln von 2048 Bit unter Verwendung des Paddings nach PKCS#1 v1.5 ohnehin erforderlich.
- Kostengünstige Infrastruktur: Einsatz von gleichen OpenSource Libraries für Abarbeitung von CVC-Zertifikaten sowie Anwendungen von kryptografischen Funktionen wie bei X.509-Zertifikaten (analoge Strukturen). OpenSource-CVC-PKIs vorhanden und getestet.

6.1.2 Zertifikatsstruktur

Tag + Länge					Bedeutung / Inhalt
T:7F21h	L				CV Certificate
	T:7F4Eh	L			Certificate Body
		T:5F29h	L:01h		Certificate Profile Identifier (CPI)
		T:42h	L:10h		Certification Authority Reference
		T:7F49h	L		Öffentlicher Schlüssel
			T:06h	L:07h	OID für C2C mit RSA-v1-5-SHA-256:
			T:81h	L:0100h	Modulus n
			T:82h	L:03h	Öffentlicher Exponent e:= 010001h
		T:5F20h	L:xxh		Certificate Holder Reference (CHR)
		T:7F4Ch	L		Certificate Holder Authorization Template
			T:06h	L:07h	OID für VK
			T:53h	L:01h	Certificate Holder Authorization (CHA)
		T:5F25h	L:06h		Certificate Effective Date YYMMDD (unpacked BCD)
		T:5F24h	L:06h		Certificate Expiration Date YYMMDD (unpacked BCD)
	T:5F37h	L:0100h			Signatur über das Certificate Body Objekt

6.1.2.1 CVC-Zertifikate ['7F21']

Inhaber	CVC-Zertifikat ['7F21']
Versicherter (Personenzertifikat)	CVC.PDC$_n$
Leistungserbringer (Personenzertifikat)	CVC.HPC$_n$
Versichererherausgeberorganisation	CVC.CA_ORG_PDC$_m$
Leistungserbringerherausgeberorganisation	CVC.CA_ORG_HPC$_m$
Nationale, neutrale CVC-Root-CA	CVC.CA_ROOT_VK

6.1.2.2 Signatur ['5F37']

Inhaber	Signatur ['5F37']
Versicherter: CVC.PDC$_n$	- Sig(CA_ORG_PDC$_m$) - RSA 2048 - SHA 256 - Padding: EMSA-PKCS#1 v1.5
Leistungserbringer: CVC.HPC$_n$	- Sig(CA_ORG_HPC$_m$) - RSA 2048 - SHA 256 - Padding: EMSA-PKCS#1 v1.5
Versichererherausgeberorganisation: CVC.CA_ORG_PDC$_m$	- Sig(CA_ROOT_VK) - RSA 2048 - SHA 256 - Padding: EMSA-PKCS#1 v1.5
Leistungserbringerherausgeberorganisation: CVC.CA_ORG_HPC$_m$	- Sig(CA_ROOT_VK) - RSA 2048 - SHA 256 - Padding: EMSA-PKCS#1 v1.5
Nationale, neutrale CVC-Root-CA: CVC.CA_ROOT_VK	- Sig(CA_ROOT_VK) - RSA 2048 - SHA 256 - Padding: EMSA-PKCS#1 v1.5

6.1.2.3 CPI - Certificate Profile Identifier ['5F29']

CPI - Certificate Profile Identifier ['5F29']	CPI Kodierung: ['01'..'7E']
Versichertenkarte: PDC	05h
Leistungserbringerkarte: HPC	03h
Versichererherausgeberorganisation: CA_ORG_PDC	04h
Leistungserbringerherausgeberorganisation: CA_ORG_HPC	02h
Nationale, neutrale CA_ROOT_VK	01h

6.1.2.4 CAR- Certification Authority Reference (Authority Key Identifier)

CAR ['42']	CA Name [5 Bytes]		Erweiterung für Schlüsselreferenzierung		
	Land	Name	Service-Indikator	CA - spezifische Information	Seriennummer
Länge	[2 Byte]	[3 Byte]	[1 BCD]	[1 BCD]	[10 Byte]
PDC	'CH'	'NNN'	'6'	'2'	'nnnnnnnnnn'
HPC	'CH'	'NNN'	'6'	'1'	'nnnnnnnnnn'
CA_ORG_PDC	'CH'	'NNN'	'6'	'0'	'nnnnnnnnnn'
CA_ORG_HPC	'CH'	'NNN'	'6'	'0'	'nnnnnnnnnn'
CA_ROOT_VK	'CH'	'NNN'	'6'	'0'	'nnnnnnnnnn'

Land:	Ländercode entsprechend ISO 3166 (2 Bytes CH = Schweiz)
CA- spezifische Information	- [0]: Zertifikat ausgestellt durch nationale, neutrale CVC-Root-CA - [1]: Zertifikat ausgestellt durch CA der Leistungserbringerherausgeber-organisation - [2]: Zertifikat ausgestellt durch CA der Versichererherausgeberorganisation
Seriennummer	Seriennummer des für die Signatur benutzte und referenzierte Herausgeber-organisationszertifikats. Kann für die Revokation benutzt werden.

6.1.2.5 CHR- Certificate Holder Reference (Subject Key Identifier)

Herausgeberorganisationszertifikate

CHR ['5F20']	CA Name [5 Bytes]		Erweiterung für Schlüsselreferenzierung		
	Land	Name	Service-Indikator	CA - spezifische Information	Seriennummer
Länge	[2 Byte]	[3 Byte]	[1 BCD]	[1 BCD]	[10 Byte]
CA_ORG_PDC	'CH'	'NNN'	'6'	'2'	'nnnnnnnnnn'
CA_ORG_HPC	'CH'	'NNN'	'6'	'1'	'nnnnnnnnnn'
CA_ROOT_VK	'CH'	'NNN'	'6'	'0'	'nnnnnnnnnn'

Land:	Ländercode entsprechend ISO 3166 (2 Bytes CH = Schweiz)
CA- spezifische Information	- [0]: Zertifikat der nationalen, neutrale CVC-Root-CA - [1]: Zertifikat einer Leistungserbringerherausgeberorganisation - [2]: Zertifikat einer Versichererherausgeberorganisation
Seriennummer	Eigene Seriennummer des Herausgeberorganisationszertifikats. Kann für die Revokation benutzt werden.

Leistungserbringerzertifikate, LE-Zertifikate

CHR ['5F20']			
	LE-Kennziffer / EAN	Besitzer	ICCSN
Länge	[14 BCD]	[2 BCD]	[10 Byte]
HPC Inhaber	'.....'	'00'	'aaaaaaaaaa'
HPC delegiert_1	dito LE-Kennziffer	'01'	'bbbbbbbbbb'
HPC delegiert_n	dito LE-Kennziffer	'99'	'zzzzzzzzzz'

Versichertenzertifikate

CHR ['5F20']	
	ICCSN
Länge	[10 Byte]
PDC	'xxxxxxxxxx'

Legende

Name: NNN	[3 Byte]
CA_ROOT_VK	RVK
VeKa-Center SASIS	SAS
z.B. Helsana	HLS

Name: NNN	[3 Byte]
z.B. FMH	FMH
z.B. OFAC	OFC
z.B. Pflegefachleute	SBK
usw.	.....

Service Indikator:	[1 BCD]
Digital Signature	'0'
Entity Authentication	'1'
Key Encipherment	'2'
Data Encipherment	'3'
Key Agreement	'4'
Entity Authentication(C)	'5'
CertSign (no service indicated)	'6
CertSign for Authentication and Key Encipherment	'7'
CertSign for Authentication and Key Agreement	'8'

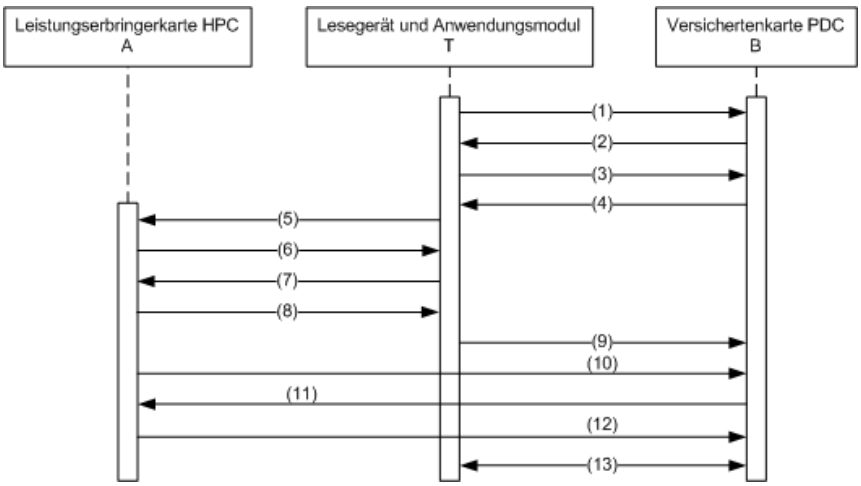
6.1.2.6 CHA - Certificate Holder Authorisation

Karte/Zertifikat	CHA Profil-ID	CHA ['5F4C']	Schlüssel 1 [eCH-0064]	Schlüssel 2 [eCH-0064]	Schlüssel 3 [eCH-0064]	Beschrieb
CA_ROOT_VK	CHA_0	00h				Kein Zugriff auf Notfalldaten
CA_ORG_HPC	CHA_0	00h				Kein Zugriff auf Notfalldaten
CA_ORG_PDC	CHA_0	00h				Kein Zugriff auf Notfalldaten
PDC	CHA_0	00h				Kein Zugriff auf Notfalldaten
HPC	CHA_1	01h	x	x		Zugriff auf Notfalldaten
HPC	CHA_2	02h	x		x	Zugriff auf Notfalldaten
HPC	CHA_3	03h	x	x		Zugriff auf Notfalldaten
HPC	CHA_4	04h	x	x		Zugriff auf Notfalldaten
HPC	CHA_5	05h	x			Zugriff auf Notfalldaten
HPC	CHA_6	06h	x			Zugriff auf Notfalldaten
HPC	CHA_7	07h	x			Zugriff auf Notfalldaten
HPC	CHA_8	08h	x			Zugriff auf Notfalldaten
HPC	CHA_9	09h	x			Zugriff auf Notfalldaten
HPC	CHA_{10}	10h	x			Zugriff auf Notfalldaten

6.1.2.7 OID-Kodierung

Item	OID-Kodierung ['06']	OID-Nummer	OID-Name	Registration Authority
Application	'60 85 74 05 22 01 01'	2.16.756.5.34.1.1	id-VVK-832105-SwissInsuranceCard-v1	BAKOM
Algorithm	'60 85 74 05 22 02 01'	2.16.756.5.34.2.1	id-C2C-RSA-v1-5-SHA-256	BAKOM

6.2 Verfahren: Offline Card-to-Card- Authentifizierung und Autorisierung



SEQ	COM	Beschreibung
1	(1)	SelectFile ‖ ReadRecord ($ICCSNF_B$)
	REM	Selektieren und Auslesen des Chipkarten Identifier Files
2	(2)	{$ICCSNF_B$}
	REM	Übertragung des Chipkarten Identifier Files der Versichertenkarte
3	(3)	SelectFile ‖ ReadBinary (EF.CVC.PDC)
	REM	Selektieren und Auslesen des CVC-Personenzertifikats des Versicherten
4	(4)	{CVC.PDC}
	REM	Übertragung des CVC-Personenzertifikats des Versicherten
5	(T)	MSE SET<PuK.CA_ORG_PDC_m←(CVC. CA_ORG_PDC_m)>
	REM	Setzten des öffentlichen CA-Schlüssels der Versicherer-Organisation
6	(T)	PSO VERIFY CERTIFICATE(CVC.PDC) → nok→break; ok→continue; Store (CHR_B); Extract ($ICCSN_B$)
	REM	Prüfen des CVC-Personenzertifikats des Versicherten
6a	(T)	optional: Compare($ICCSNF_B$[$ICCSN_B$]; EF.CVC.PDC[$ICCSN_B$])
	REM	Überprüfung der Chipkartenseriennummer durch Vergleich der Nummern auf der Karte und im CVC-Personenzertifikat
7	(5)	SelectFile‖ ReadBinary ($ICCSNF_A$)
	REM	Selektieren und Auslesen des Chipkarten Identifier Files der Leistungserbringerkarte
8	(6)	{$ICCSNF_A$}
	REM	Übertragung des Chipkarten Identifier Files der Leistungserbringerkarte
9	(7)	SelectFile ‖ ReadBinary (EF.CVC.HPC)

SEQ	COM	Beschreibung
	REM	Selektieren und Auslesen des CVC-Leistungserbringerzertifikats
10	(8)	{CVC.HPC}
	REM	Übertragung des CVC-Leistungserbringerzertifikats
11	(T)	Read ‖ Extract(CVC.HPC) -> (CPI = 04, CAR, CHR, CHA) -> Store
	REM	Auslesen und Speichern der notwendigen Leistungserbringer-Attribute aus dem CVC-Leistungserbringerzertifikat
12	(T)	Search((CPI, CAR, CHR, CHA), (CVC.CA_ORG_HPC$_m$))$_m$; -> Select(C VC.CA_ORG_HPC$_m$)
	REM	Anhand der Leistungserbringer-Attribute wird das entsprechende Leistungserbringer-organisationszertifikat selektiert.
13	(T)	SelectFile ‖ WriteBinary(CVC.CA_ORG_HPC$_m$) ->
	REM	Anwählen und Beschreiben des dedizierten Containers mit dem Leistungserbringer-organisationszertifikat
14	(9)	{CVC.CA_ORG_HPC$_m$} -> EF.CVC.CA_ORG_HPC
	REM	Übertragung und Speicherung des Leistungserbringerorganisationszertifikats
15	(10)	{CVC.HPC[P_A]}
	REM	Das CVC-Leistungserbringerzertifikat wird der Versichertenkarte zur Verifikation bereitgestellt
16	(B)	MSE SET<PK.CA_ORG_PDC$_m$>
	REM	Setzten des öffentlichen Root-Schlüssels der neutralen CA_ROOT_VK
17	(B)	PSO VERIFY CERTIFICATE(CVC.CA_ORG_HPC$_m$) → nok→break; ok→continue; Store (PK.CA_ORG_HPC$_m$)
	REM	Prüfen des CVC-Zertifikats der entsprechenden Leistungserbringer- Herausgeberorganisation
18	(B)	MSE SET<PK.CA_ORG_HPC$_m$←(CVC.CA_ORG_HPC$_m$)>
	REM	Setzten des öffentlichen CA-Schlüssels der entsprechenden Leistungserbringer- Herausgeberorganisation
19	(B)	PSO VERIFY CERTIFICATE(CVC.HPC) → nok→break; ok→continue;
	REM	Prüfen des CVC-Leistungserbringerzertifikats
20	(B)	MSE SET<PuK.HPC←(CVC.HPC)>
	REM	Setzen des Schlüssels des Leistungserbringerzertifikats
21	(B)	Generate (R_B)
	REM	Zufallszahl erzeugen
22	(11)	{R_B}

SEQ	COM	Beschreibung
	REM	Zufallszahl übermitteln
23	(A)	$sS_A(R_B)$
	REM	Zufallszahl signieren
24	(12)	$\{sS_A(R_B)\}$
	REM	Signierte Zufallszahl übermitteln
25	(B)	$sP_A(sS_A(R_B))$; $R_B = R_B$ → ok→continue; nok→break
	REM	Verifikation der Signatur der signierten Zufallszahl
26	(B)	(CHA_n); Autorisation ok!
	REM	Der Karteninhaberautorisierungsmerkmalswert CHA_n wird freigeschaltet
27	(13)	Lesegerät/Anwendungsmodul kann erst jetzt mittels dem Karteninhaberautorisierungsmerkmalswert (CHA_n) auf die Versichertenkarte zugreifen
	REM	Datenaustausch zwischen dem Lesegerät/Anwendungsmodul und der Versichertenkarte

6.2.1 Selektieren und Auslesen des Chipkarten Identifier Files

Terminal	**SELECT**					
CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	00h	00h	02h	$EF.ICCSN_B$ [FID:2F05]	-

Terminal	**READ RECORD**					
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	B2h	01h	04h	-	-	0Ah

6.2.2 Übertragung des Chipkarten Identifier Files der Versichertenkarte

Die ICCSN entspricht der Kennnummer der Versichertenkarte.

PDC → Terminal (2)	**Datenübertragung**
Response Data	SW12
DO $ICCSN_B$ (Identifier (tag) '5A') [10 Byte]	90 00h

6.2.3 Selektieren und Auslesen des CVC-Personenzertifikats des Versicherten

Terminal (3)	**SELECT**					
CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	00h	00h	02h	EF.CVC.PDC [FID:2F03]	-

Terminal (3)	READ BINARY					
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	B0h	00h	00h	-	-	000000h

6.2.4 Übertragung des CVC-Personenzertifikats des Versicherten

PDC → Terminal (4)	Datenübertragung
Response Data	SW12
CVC.PDC	90 00h

6.2.5 Setzten des öffentlichen CA-Schlüssels der Versicherer-Organisation

Terminal	MSE SET <PuK.CA_ORG_PDC ← (CVC.CA_ORG_PDC)				
Load	Parse	Tag	L	Data	Action
$CVC.CA\_ORG\_PDC_m$	CVC.CA_ORG_PDC	81h	0100h	PuK.CA_ORG_PDC	Store_1

6.2.6 Prüfen des CVC-Personenzertifikats des Versicherten

Terminal	PSO VERIFY CERTIFICATE (CVC.PDC)							
Load	Parse	Tag	L	Data	Parse	Tag	L	Data
CVC.PDC	**CVC.PDC**	**5F37h**	**0100h**	**Sig (CVC.PDC)**	**CVC.PDC**	**7F4Eh**	**015Bh**	**Certificate Body**
Action	$PAD_{PKCS\#1\_v1.5}$[SHA256(Certificate Body)]				Recall_1	Verify Signature		

$PAD_{PKCS\#1\_v1.5}$[SHA256(Certificate Body)]	Padding nach EMSA-PKCS-v1_5	
Wert	**Länge**	**Beschreibung**
´00 01 [FF ..] 00´	L – 51	Padding für Signatur
´30 31 30 0D 06 09 60 86 48 01 65 03 04 02 01 05 00 04 20´	19	
…	32	SHA-256-Hash über (Certificate Body)

4.2.6a Überprüfung der Chipkartenseriennummer durch Vergleich der Nummern auf der Karte und im CVC-Personenzertifikat

Terminal: Optional	$COMPARE(ICCSNF_B[ICCSN_B]; EF.CVC.PDC[ICCSN_B])$					
Load	**Parse**	**Tag**	**L**	**Data**	**Load**	**Action**
CVC.PDC	CVC.PDC	5F20h	0Ah	$ICSSN_B$	$ICSSN_B$-F	Compare

6.2.7 Selektieren und Auslesen des Chipkarten Identifier Files der Leistungserbringerkarte

Terminal (5)	SELECT				
CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	00h	00h	02h	FID: $EF.ICCSN_A$	-

Terminal (5)	READ RECORD					
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	B2h	01h	04h	-	-	0Ah

6.2.8 Übertragung des Chipkarten Identifier Files der Leistungserbringerkarte

HPC → Terminal (6)	Datenübertragung
Response Data	SW12
DO $ICCSN_A$ (Identifier (tag) '5A') [10 Byte]	90 00h

6.2.9 Selektieren und Auslesen des CVC-Leistungserbringerzertifikats

Terminal (7)	SELECT					
CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	00h	00h	02h	FID: EF.CVC.HPC	-

Terminal (7)	READ BINARY					
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	B0h	00h	00h	-	-	000000h

6.2.10 Übertragung des CVC-Leistungserbringerzertifikat

HPC → Terminal (8)	Datenübertragung
Response Data	SW12
CVC.HPC	90 00h

6.2.11 Auslesen und Speichern der notwendigen Leistungserbringer-Attribute aus dem CVC- Leistungserbringerzertifikat

Terminal	READ	EXTRACT (CVC.HPC)							
Read	Parse	Tag	L	Data	Parse	Tag	L	Data	Action
CVC.HPC	CVC.HPC	5F29h	01h	CPI=03	CVC.HPC	5F20h	10h	CHR	Store_2

6.2.12 Selektion Leistungserbringer-Organisationszertifikat anhand der Leistungserbringer-Attribute

Terminal	SEARCH(CPI,CAR),$(CVC.CA\_ORG\_HPC_m)_m$ → Select(CVC.CA_ORG_HPC)						
Load and Parse #1	Tag	L	Data	Parse	Tag	L	Data
$(CVC.CA\_ORG\_HPC_m)_m$	5F29h	01h	CPI=02	$CVC.CA\_ORG\_HPC_m$	5F20h	10h	CHR
Action	Recall_2	Check CPI, Compare: CHR		Select$(CVC.CA_ORG_HPC)_m$ or goto #1			

6.2.13 Anwählen und Beschreiben des dedizierten Containers mit dem Leistungserbringerorganisationszertifikat

Entfällt: Nach heutigem Stand der Technik nicht notwendig

6.2.14 Übertragung und Speicherung des Leistungserbringerorganisationszertifikats

Entfällt: Nach heutigem Stand der Technik nicht notwendig

6.2.15 Bereitstellung des Leistungserbringerzertifikats zur Verifikation

Entfällt: Nach heutigem Stand der Technik nicht notwendig

6.2.16 Setzen des öffentlichen Root-Schlüssels der neutralen CA_ROOT_VK

PDC					**MSE SET <PuK.CA_ROOT_VK> ← (CVC.CA_ROOT_VK)**	
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	22h	81h	B6h	12	{'83 ' ‖ '10' ‖ CHR(CVC.CA_ROOT_VK }	-

6.2.17 Prüfen des CVC-Zertifikats der entsprechenden Leistungserbringer-Herausgeberorganisation

PDC (9)				**PSO VERIFY CERTIFICATE (CVC.CA_ORG_HPC$_m$)**		
CLA	INS	P1	P2	Lc	Data	Le
00h	2Ah	00h	BEh	026Bh	7F4E{Certificate Body} ‖ 5F37{Signature}	-

6.2.18 Setzen des CA-Schlüssels der entsprechenden Leistungserbringer-Herausgeberorganisation

PDC					**MSE SET <PuK.CA_ORG_HPC$_m$← (CVC.CA_ORG_HPC$_m$)>**	
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	22h	81h	B6h	12h	{'83 ' ‖ '10' ‖ CHR(CVC.CA_ORG_HPC$_m$ }	-

6.2.19 Prüfen des CVC-Leistungserbringerzertifikats

PDC (10)				**PSO VERIFY CERTIFICATE (CVC.HPC)**		
CLA	INS	P1	P2	Lc	Data	Le
00h	2Ah	00h	BEh	026Dh	7F4E{Certificate Body} ‖ 5F37{Signature}	-

6.2.20 Setzen des Schlüssels des Leistungserbringerzertifikats

PDC					**MSE SET <PuK.HPC← (CVC.HPC)>**	
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	22h	81h	A4h	14h	{'83 ' ‖ '12' ‖ CHR(CVC.HPC }	-

6.2.21 Zufallszahl erzeugen

Terminal		**GET CHALLENGE (R_B)**				
CLA	INS	P1	P2	Lc	Data	Le
00h	84h	00h	00h	08h	-	-

6.2.22 Zufallszahl übermitteln

PDC → Terminal (11a)	**Datenübertragung**
Response Data	SW12
R_B (8 Byte Zufallszahl)	90 00h

6.2.23 Zufallszahl signieren

Hinweis: Im Standard eCH-0064 wurde darauf geachtet, dass beim Card2Card-Verfahren ohne Einverständnis des Leistungserbringers und ohne seine Kontrolle keine logisch nachweisbaren Verknüpfungen zwischen der Versichertenkarte und der Leistungserbringerkarte erfolgen können. In diesem Kontext müssen sämtliche Daten, welche mittels INTERNAL AUTHENTICATE durch die Leistungserbringerkarte signiert werden, rein zufällig sein, so dass damit nie ein nichtabstreitbarer Bezug zur Versichertenkarte hergestellt werden kann. Damit ist sichergestellt, dass die Leistungserbringerkarte während der Card2Card-Authentifizierung nie eine Signatur über einen Merkmalswert der Versichertenkarte leistet (Kennnummer der Versichertenkarte, Referenznummer, usw.).

6.2.23.1 Direkt mit der Leistungserbringerkarte [HPC]

Terminal → HPC(11b)	**INTERNAL AUTHETICATE (R_B): $sS_A(R_B)$**					
CLA	INS	P1	P2	Lc	Data	Le
00h	88h	00h	00h	000008h	R_B	000000h

6.2.23.2 Indirekt mit der Leistungserbringerkarte: Hashing und Padding in Middleware

Für Chipkartenbetriebssysteme bei HPCs, welche INTERNAL AUTHENTICATE mit RSA 2048, SHA256 und Padding nach PKCS#1 v1.5 nicht direkt implementiert haben, kann dieser APDU-Befehl auch als Wrapper, welcher im Lesegerät/Anwendungsmodul unter dessen Betriebssystem läuft, unter Anwendung der APDU-Befehle PSO COMPUTE DIGITAL SIGNATURE RSA2048 (Pure Mode) oder PSO ENCIPHER RSA2048 (Pure Mode) oder gleichwertige dedizierte Befehle umgesetzt werden. Für die kryptografischen Funktionen wie Padding PKCS#1 v1.5 und SHA256 stehen eine Vielzahl von Open Source Bibliotheken wie z.B. das OpenSSL Toolkit written by Eric Young u.a.m. zur Verfügung.

Terminal+HPC: Terminal → HPC(11b)		**INTERNAL AUTHETICATE (R_B): $sS_A(R_B)$**	
Get	Apply	Padding	
R_B	**$SHA256(R_B)$**	**$PAD_{PKCS\#1\_v1.5}[SHA256(R_B)]$**	**Bemerkungen**
Action	PSO ENC RSA2048_PURE: $eS_A(PAD_{PKCS\#1\_v1.5}[SHA256(R_B)])$		Encrypt with private Key
Response Data	$sS_A(R_B)$: HPC → Terminal		Signed Random Value

$PAD_{PKCS\#1\_v1.5}[SHA256(R_B)]$	**Padding nach EMSA-PKCS-v1_5**	
Wert	**Länge**	**Beschreibung**
´00 01 [FF ..] 00´	L – 51	Padding für Signatur
´30 31 30 0D 06 09 60 86 48 01 65 03 04 02 01 05 00 04 20´	19	
…	32	SHA-256-Hash über R_B[8 Byte]

6.2.24 Signierte Zufallszahl übermitteln

HPC → Terminal (12a)	**Datenübertragung**
Response Data	SW12
$sS_A(R_B)$	90 00h

6.2.25 Verifikation der signierten Zufallszahl

Terminal → PDC(12b)				**EXTERNAL AUTHETICATE (R_B): $sP_A(sS_A(R_B))$**		
CLA	INS	P1	P2	Lc	Data	Le
00h	82h	00h	00h	0100h	$sS_A(R_B)$	-

6.2.26 Der Kartenautorisierungsmerkmalswert CHA_n wird freigeschaltet

HPC → Terminal	**Datenübertragung**
Response Data	SW12
-	90 00h

6.3 Optionale Erweiterungen gegenüber eCH-0064

6.3.1 Zufallszahl erzeugen

Terminal	**GET CHALLENGE (R_T)**
Generate Random Value	
R_T (8 Byte)	

6.3.1.1 Zufallszahl signieren

Terminal → PDC				**INTERNAL AUTHETICATE (R_T): $sS_B(R_T)$**		
CLA	INS	P1	P2	Lc	Data	Le
00h	88h	00h	00h	000008h	R_T (8 Byte)	000000h

6.3.1.2 Signierte Zufallszahl übermitteln

PDC → Terminal	**Datenübertragung**
Response Data	SW12
$sS_B(R_T)$	90 00h

6.3.1.3 Verifikation der signierten Zufallszahl

Terminal	**EXTERNAL AUTHENTICATE (R_T): $sP_B(sS_B(R_T))$**						
Load	Parse	Tag	L	Data	Load	Decipher	Data
CVC.PDC	CVC.PDC	81h	0100h	PuK.CVC.PDC	$sS_B(R_T)$	$dP_B(sS_B(R_T))$	Store 1
Get	Apply	Padding				Data	
R_T	SHA256(R_T)	$PAD_{PKCS\#1\_v1.5}$[SHA256(R_T)]				Store 2	
Action	Compare (Store 1, Store 2, Result)			Result:= equal→ok; not-equal→ error			

$PAD_{PKCS\#1\_v1.5}$[SHA256(R_T)]	**Padding nach EMSA-PKCS-v1_5**	
Wert	**Länge**	**Beschreibung**
´00 01 [FF ..] 00´	L – 51	Padding für Signatur
´30 31 30 0D 06 09 60 86 48 01 65 03 04 02 01 05 00 04 20´	19	
…	32	SHA-256-Hash über R_T[8 Byte]

6.3.2 Prüfen des CVC-Organisationszertifikats der Versicherer-Organisation

Terminal	**MSE SET <PuK.CA_ROOT_VK ← (CVC.CA_ROOT_VK)**				
Load	Parse	Tag	L	Data	Action
CVC.CA_ROOT_VK	CVC.CA_ROOT_VK	81h	0100h	PuK.CA_ROOT_VK	Store_1

Terminal	**PSO VERIFY CERTIFICATE ($CVC.CA\_ORG\_PDC_m$)**			
Load	Parse	Tag	L	Data_#1
$CVC.CA\_ORG\_PDC_m$	$CVC.CA\_ORG\_PDC_m$	5F37h	0100h	Sig (CVC.PDC)
Load	Parse	Tag	L	Data_#2
$CVC.CA\_ORG\_PDC_m$	$CVC.CA\_ORG\_PDC_m$	7F4Eh	0161h	Certificate Body
Action	$PAD_{PKCS\#1\_v1.5}$[SHA256(Certificate Body)]	Recall_1	Verify Signature (Recall_1,#1,#2)	

$PAD_{PKCS\#1\_v1.5}$[SHA256(Certificate Body)]	**Padding nach EMSA-PKCS-v1_5**	
Wert	**Länge**	**Beschreibung**
´00 01 [FF ..] 00´	L – 51	Padding für Signatur
´30 31 30 0D 06 09 60 86 48 01 65 03 04 02 01 05 00 04 20´	19	
…	32	SHA-256-Hash über (Certificate Body)

7 PIN-Management nach eCH-0064 (Notfalldaten)

7.1 Befehlssatz

ZWINGEND: Für das PIN-Management wird folgender Befehlssatz gemäss ISO/IEC 7816-4 angewandt:

- VERIFY
- CHANGE REFERENCE DATA
- RESET RETRY COUNTER
- ENABLE VERIFICATION REQUIREMENT
- DISABLE VERIFICATION REQUIREMENT
- TERMINATE CARD USAGE

7.1.1 VERIFY

Dieses Kommando ist kodiert nach [ISO/IEC 7816-4] 7.5.6.

VERIFY						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	20h	00h	PIN1: 01h PIN2: 02h PUK: 04h	08h 08h 08h	PIN PIN PUK	-

Mit VERIFY werden die PINs und die PUK vom Chip verifiziert und die Sicherheitszustände entsprechend geändert.

7.1.2 CHANGE REFERENCE DATA

Dieses Kommando ist kodiert nach [ISO/IEC 7816-4] 7.5.7.

CHANGE REFERENCE DATA						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	24h	01h	PIN1: 01h PIN2: 02h	08h	PIN	-

7.1.3 RESET RETRY COUNTER

Dieses Kommando ist kodiert nach [ISO/IEC 7816-4] 7.5.10.

RESET RETRY COUNTER						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	2Ch	03h	PIN1: 11h PIN2: 12h	(08h)	(PIN)	-

Mit diesem Kommando kann der Fehlbedienungszähler zurückgesetzt und optional gleichzeitig ein neuer PIN gesetzt werden (Data).

7.1.4 ENABLE VERIFICATION REQUIREMENT-G (Global)

Dieses Kommando ist kodiert nach [ISO/IEC 7816-4] 7.5.8.

ENABLE VERIFICATION REQUIREMENT-G						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	28h	00h	00h	08h	PIN1	-

Mit diesem Kommando kann der PIN-Mechanismus für die Daten nach Artikel 42a Absatz 4 KVG global aktiviert werden. Allfällige PIN-Schutzzustände von Notfall-Datenkategorien, welche durch den Versicherten ausgewählt wurden, werden dabei wieder aktiviert. Bei der Auslieferung der Versichertenkarte durch die Versicherer ist der PIN-Mechanismus deaktiviert und sämtliche Notfall-Datenkategorien sind ohne PIN-Schutz belegt.

7.1.5 DISABLE VERIFICATION REQUIREMENT-G (Global)

Dieses Kommando ist kodiert nach [ISO/IEC 7816-4] 7.5.9.

DISABLE VERIFICATION REQUIREMENT-G						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	26h	00h	00h	08h	PIN1	-

Mit diesem Kommando kann der PIN-Mechanismus für die Daten nach Artikel 42a Absatz 4 KVG global deaktiviert werden. Allfällige PIN-Schutzzustände von Notfall-Datenkategorien, welche durch den Versicherten ausgewählt wurden, werden dabei deaktiviert.

7.1.6 ENABLE VERIFICATION REQUIREMENT-D (Datenkategorie)

Dieses Kommando ist kodiert nach [ISO/IEC 7816-4] 7.5.8.

ENABLE VERIFICATION REQUIREMENT-D						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	28h	00h	01h	08h	PIN1	-

Mit diesem Kommando können die Notfall-Datenkategorien der Daten nach Artikel 42a Absatz 4 KVG und nach eCH-0064 selektiv, einzeln und sequentiell mit einem PIN-Schutz belegt werden. Bei der Auslieferung der Versichertenkarte durch die Versicherer sind sämtliche Notfall-Datenkategorien ohne PIN-Schutz belegt.

7.1.7 DISABLE VERIFICATION REQUIREMENT-D (Datenkategorie)

Dieses Kommando ist kodiert nach [ISO/IEC 7816-4] 7.5.9.

DISABLE VERIFICATION REQUIREMENT-D						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	26h	00h	01h	08h	PIN1	-

Mit diesem Kommando können die Notfall-Datenkategorien der Daten nach Artikel 42a Absatz 4 KVG und nach eCH-0064, welche vom Versicherten selektiv mit einem PIN-Schutz belegt worden sind und einen entsprechenden PIN-Schutzzustand aufweisen, einzeln und sequentiell wieder zurückgesetzt werden.

7.1.8 TERMINATE CARD USAGE

Dieses Kommando ist kodiert nach [ISO/IEC 7816-9] 6.6

TERMINATE CARD USAGE						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	FEh	00h	00h	-	-	-

Mit diesem Kommando kann der Versicherte nach Verifizierung der PUK die Karte z. B. für den Rückversand unbrauchbar machen. Die Karte reagiert dann auf kein Kommando mehr. Sie zeigt dies durch das LCS-Byte im ATR an, das dann den Wert 0x0F hat. Vgl. [7816-4] 5.3.3.2.

7.2 PIN-Schutzzustände

ZWINGEND: Sämtliche Prozesse zur Festlegung der PIN-Schutzzustände erfolgen einzeln, sequentiell und können nicht kombiniert werden.

7.2.1 Versichertenkarte nach Auslieferung durch den Versicherer

ZWINGEND: Der PIN- Mechanismus ist deaktiviert. Damit ist grundsätzlich ein freier, geregelter Zugriff auf alle Notfalldaten möglich, sofern Card-to-Card-Authentisierung und Autorisierung erfolgreich durchgeführt worden sind.

7.2.2 Aktivierung des Pin Mechanismus

ZWINGEND: Folgende Prozessschritte müssen durchgeführt werden:

1. Aktivierung des PIN-Mechanismus
2. Auswahl der Kategorien der Notfalldaten (EF.Dateien), welche mit einem PIN-Schutz belegt werden sollen, mittels Tastatur durch den Versicherten.
3. Eingabe des neuen PIN-Kodes mittels Tastatur durch den Versicherten

Nr.	Prozessschritt	Lesegerät/ Anwendungsmodul [IFD]	PDC
1a	Aktivierung des PIN-Mechanismus	SELECT EF.PIN1 [3F00:0011]	⇐ FCI + Statuswort
1b	Aktivierung des PIN-Mechanismus	ENABLE VERIFICATION REQUIREMENT-G [DATA= 00000000, ISO 8859-1]	Globaler PIN-Schutz mittels PIN1 eingeschaltet ⇐ Statuswort
2a	Auswahl der Notfall-Datenkategorie, welche mit einem PIN-Schutz belegt werden soll, mittels Tastatur durch den Versicherten	SELECT EF.NOT [DF01:1F01 .. 1F07]	⇐ FCI + Statuswort
2b	Belegen der ausgewählten Notfall-Datenkategorie mit PIN-Schutz	ENABLE VERIFICATION REQUIREMENT-D [DATA= 00000000, ISO 8859-1]	⇐ Statuswort
2n	Auswahl weiterer Notfall-Datenkategorien, welche mit PIN-Schutz belegt werden sollen: Prozessschritte einzeln, sequentiell auf den ausgewählten Notfall-Datenkategorien durchführen	Prozessschritte 2a + 2b	Die neutralen PIN1 - Daten können bei Bedarf im Lesegerät zwischengespeichert werden
3	Eingabe des neuen PIN-Kodes mittels Tastatur durch den Versicherten	CHANGE REFERENCE DATA [P1=01h, P2=01h, DATA= PIN1(8)] (PIN: ISO 8859- 1)	⇐ Statuswort

7.2.3 Änderung des PIN-Schutzes auf Kategorien der Notfalldaten

ZWINGEND: Folgende Prozessschritte müssen durchgeführt werden:

1. Eingabe des PIN-Kodes mittels Tastatur durch den Versicherten
2. Auswahl der Kategorien der Notfalldaten (EF.Dateien), welche mit einem PIN-Schutz belegt oder wieder freigegeben werden sollen, mittels Tastatur durch den Versicherten
3. Eingabe des PIN-Kodes mittels Tastatur durch den Versicherten

Nr.	Prozessschritt	Lesegerät/ Anwendungsmodul [IFD]	PDC
1	Eingabe des PIN-Kodes mittels Tastatur durch den Versicherten	VERIFY [P2=01h] [DATA= PIN1(8), ISO 8859-1]	⇐ Statuswort
2a	Auswahl der Notfall - Datenkategorie, welche mit einem PIN-Schutz belegt werden soll, mittels Tastatur durch den Versicherten	SELECT EF.NOT [DF01: 1F01 .. 1F07]	⇐ FCI + Statuswort
3a	Belegen der ausgewählten Datenkategorie mit PIN - Schutzmechanismus und Eingabe des PIN - Kodes mittels Tastatur durch den Versicherten	ENABLE VERIFICATION REQUIREMENT-D [DATA= PIN1(8), ISO 8859-1]	⇐ Statuswort
2b	Auswahl der Notfall- Datenkategorie, welche wieder frei gegeben werden soll, mittels Tastatur durch den Versicherten	SELECT EF.NOT [DF01:1F01 .. 1F07]	⇐ FCI + Statuswort
3b	Deaktivierung des PIN - Schutzmechanismus bezogen auf die ausgewählte Notfall-Datenkategorie durch Eingabe des PIN -Kodes mittels Tastatur durch den Versicherten	DISABLE VERIFICATION REQUIREMENT-D [DATA= PIN1(8), ISO 8859-1]	⇐ Statuswort
2n 3n	Auswahl weiterer Notfall- Datenkategorien, welche mit PIN -Schutz belegt werden oder wieder freigegeben werden sollen: Prozessschritte einzeln, sequentiell auf den ausgewählten Notfall - Datenkategorien durchführen.	Prozessschritte 2a + 3a bzw. 2b + 3b	Die PIN1-Daten können bei Bedarf im Lesegerät gesichert zwischengespeichert werden

7.2.4 Änderung des PIN-Kodes (PIN-Mechanismus aktiviert)

ZWINGEND: Folgende Prozessschritte müssen durchgeführt werden:

1. Eingabe des PIN-Kodes mittels Tastatur durch den Versicherten
2. Eingabe des Änderungsbegehrens mittels Tastatur durch den Versicherten
3. Eingabe des geänderten PIN-Kodes mittels Tastatur durch den Versicherten
4. Nochmalige Eingabe des geänderten PIN-Kodes mittels Tastatur durch den Versicherten

Nr.	Prozessschritt	Lesegerät/ Anwendungsmodul [IFD]	PDC
1	Eingabe des PIN-Kodes mittels Tastatur durch den Versicherten	VERIFY [P2=01h] [DATA= PIN1(8), ISO 8859-1, PIN1(alt)]	⇐ Statuswort
2	Eingabe des Änderungsbegehrens mittels Tastatur durch den Versicherten	CHANGE REFERENCE DATA [P2=01h]	warten
3	Eingabe des geänderten PIN-Kodes mittels Tastatur durch den Versicherten	[DATA= PIN1(8), ISO 8859-1, PIN1(neu)]	⇐ Statuswort
4	Nochmalige Eingabe des geänderten PIN-Kodes mittels Tastatur durch den Versicherten	VERIFY [P2=01h] [DATA= PIN1(8), ISO 8859-1, PIN1(neu)]	⇐ Statuswort, überprüfen des Statuswortes. Damit kann geprüft werden, ob der Versicherte den neuen geänderten PIN sich richtig gemerkt hat
3-4	Prozesssicherheit	VERIFY [P2=01h] DATA= PIN1(8), ISO 8859-1, PIN1(alt)] CHANGE REFERENCE DATA [P2=01] [DATA= PIN1(8), ISO 8859-1, PIN1(neu)] VERIFY [P2=01h] DATA= PIN1(8), ISO 8859-1, PIN1(neu)]	Falls der Versicherte sich den geänderte PIN-Kode falsch gemerkt hat, was bei der finalen PIN-Verifikation in Prozessschritt 4 durch ein entsprechendes Statuswort detektiert wird, könnte das Lesegerät bei fehlgeschlagener PIN-Verifikation durch den Versicherten die Prozessschritte 1-4 mit dem gesichert temporär zwischengespeicherten geänderten PIN-Kode noch einmal wiederholen und einen nochmals geänderten und diesmal den bei der Überprüfung korrekt eingegebenen neuen PIN-Kode abgespeichert belassen.

7.2.5 Deaktivierung des PIN-Mechanismus

ZWINGEND: Folgende Prozessschritte müssen durchgeführt werden:

1. Eingabe des Deaktivierungsbegehrens mittels Tastatur durch den Versicherten
2. Eingabe des PIN-Kodes mittels Tastatur durch den Versicherten

Nr.	Prozessschritt	Lesegerät/ Anwendungsmodul [IFD]	PDC
1	Eingabe des Deaktivierungsbegehrens mittels Tastatur durch den Versicherten	SELECT EF.PIN1 [3F00 0001]	⇐ FCI + Statuswort
2a	Eingabe des Deaktivierungsbegehrens mittels Tastatur durch den Versicherten	DISABLE VERIFICATION REQUIREMENT-G	⇐ Statuswort
2b	Eingabe des PIN-Kodes mittels Tastatur durch den Versicherten	[DATA= PIN1(8), ISO 8859-1]	Genereller PIN-Schutz mittels PIN1 ausgeschaltet ⇐ Statuswort

Bei einer allfälligen Reaktivierung werden die letzten PIN-Schutzbelegungen auf den gewählten Notfall-Datenkategorien wieder wirksam.

7.2.6 PIN-Sperrmechanismen

ZWINGEND: Nach 5 fehlerhaften Versuchen wird die PIN-Eingabe gesperrt. Diese kann mit einem PUK von 8 Ziffern und einer Zulassung von maximal 10 Fehlversuchen wieder entsperrt werden. Werden die 10 Fehlversuche bei der PUK-Eingabe überschritten, bleibt die Versichertenkarte gesperrt und kann nicht mehr reaktiviert werden. Der zufällige PUK-Kode durch die Versicherer bei der Kartenausgabe personalisiert und ist auf der Karte enthalten.

Nr.	Prozessschritt	Lesegerät/ Anwendungsmodul [IFD]	PDC
1	Eingabe des PIN-Kodes mittels Tastatur durch den Versicherten	VERIFY [P2=01h] DATA= PIN1(8), ISO 8859-1]	⇐ Statuswort maximal 5 Fehlversuchen
2a	Entsperrung durch PUK	VERIFY [P2=04h] DATA= PUK(8), ISO 8859-1]	⇐ Statuswort 10 Fehlversuche überschritten, Sperrung bleibt und kann nicht mehr rückgängig gemacht werden.
2b	Entsperrung durch PUK	RESET RETRY COUNTER [P2=11h]	⇐ Statuswort

8 Kantonale Modellversuche

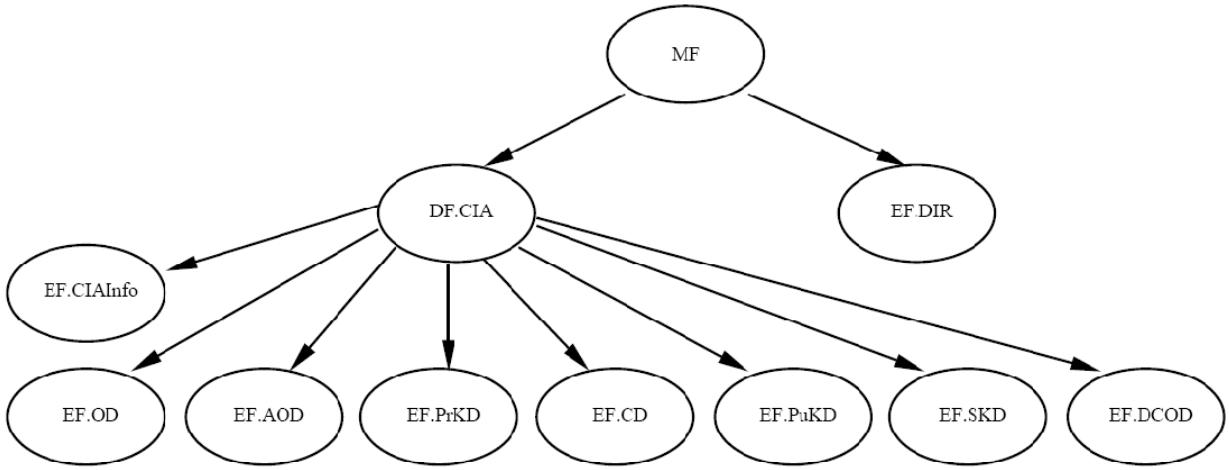
8.1 Anwendung

Für die kantonalen Modellversuche ist eine universelle PKCS#15 bzw. ISO/IEC 7816-15 Dateistruktur im entsprechend genormten Verzeichnis DF.PKCS#15 angelegt worden. Diese international genormte PKCS#15 bzw. ISO/IEC 7816-15 Dateistruktur soll allen Anbietern von Middleware und Anwendungen offen zur Verfügung stehen - es bestehen keine Einschränkungen bezüglich Lese- oder Schreibzugriff auf diese PKCS#15 bzw. ISO/IEC 7816-15 Dateistruktur. Ebenfalls besteht auf dieser international genormten, öffentlichen PKCS#15 bzw. ISO/IEC 7816-15-Dateistruktur kein Anspruch auf ein geistiges Eigentum und es kann auch keines erhoben werden. Die Middleware- und Anwendungsanbieter tragen dafür Sorge, dass das Schreiben von Datenobjekten in die PKCS#15 bzw. ISO/IEC 7816-15 Dateistruktur normengerecht erfolgt und bereits vorhandene Datenobjekte nicht ohne Einwilligung des Karteninhabers verändert oder gelöscht werden.

8.2 PKCS#15 bzw. ISO/IEC 7816-15 Spezifikation

Die PKCS #15 Spezifikation ist auch in kompatibler Form von der internationalen Chipkartennorm ISO/IEC 7816-15 abgedeckt. Bei der Erarbeitung der PKCS #15 Spezifikation wurden verschiedene Anforderungskriterien beachtet. Die Spezifikation sollte Plattform-, Hersteller- und anwendungsneutral sein und sich auch konform zu den üblichen Normen verhalten. Ausserdem musste sie modular und erweiterbar aufgebaut sein. Auch war gefordert, dass alle Daten der Anwendung mit einer ausreichenden Beschreibung versehen sind, so dass diese beim Zugriff als Referenz verwendet werden kann. Als Grundlage für die Darstellung von Dateien wurde die ISO/IEC 7816-Normenreihe, für die Formatbeschreibung wurde ASN.1 gewählt. Die für die jeweilige Abarbeitung der Daten notwendigen Kommandos sind nicht Gegenstand der PKCS #15 bzw. ISO/IEC 7816-15 Spezifikation. Aus diesen vorgenannten Gründen lässt sich PKCS #15 bzw. ISO/IEC 7816-15 verhältnismässig einfach auf Chipkarten mit typischen Multiapplikations- Betriebssystemen realisieren. Seitens des Konzepts kann jedoch jedes Sicherheitstoken für die PKCS #15- bzw. ISO/IEC 7816-15 - Funktionen benutzt werden.

8.3 Dateistruktur



ISO/IEC 7816-15 Dateistruktur

Auf der SASIS-Versichertenkarte nicht vorhanden ist EF.SKD, weil keine symmetrische Schlüssel für kantonale Modellversuche vorgesehen sind.

8.3.1 Verzeichnis PKCS#15 aka DF.CIA

Objekt	DF.PKCS#15	Verzeichnis für Notfalldaten
Objekttyp	Dedicated File	
FID	'DF02'	
AID	'A0 00 00 00 63 50 4B 43 53 2D 31 35'	
Grösse	10423 Bytes	
Zugriffsrechte	Zugriffsart [AM]	Sicherheitsbedingungen [SC]
	SELECT	Always

8.3.2 EF.CIAInfo aka EF.Tokeninfo

Objekt	EF.CIAInfo	Token Informationsdatei
Objekttyp	Elementary File	
Dateistruktur	Transparent	
Dateikategorie	Working	
FID	'5032'	
SFID	'12'	
Zugriffsrechte	Zugriffsart [AM]	Sicherheitsbedingungen [SC]
	SELECT	Always
	READ BINARY	Always

Daten im EF.CIAInfo

Data-Object				DO.CIAInfo	Meaning
T	T	T	L	30-L-{02-L-V-04-L-V-0C-L-V-03-L-V}	
30				Variable [V]	
	02		1	00h	Comp to PKCS#15
	04		10	xx xx xx xx xx xx xx xx xx xx	ICCSN: [BCD]; Kennnummer der Versichertenkarte
	0C		16	UTF8BasicLatin: 'SASIS-Intercard'	Manufacturer ID: oder SASIS-Trueb
	A0				
		0C	20	3x 3x 3x 3x 3x 3x 3x 3x … 3x 3x	ICCSN: ASCII; Kennnummer der Versichertenkarte
	03		2	04 60h	Flag: Usage

8.3.3 EF.OD aka EF.ODF

Objekt	EF.OD	Verzeichnis der Verzeichnisdateien
Objekttyp	Elementary File	
Dateistruktur	Transparent	
Dateikategorie	Working	
FID	'5031'	
SFID	'11'	
Zugriffsrechte	Zugriffsart [AM]	Sicherheitsbedingungen [SC]
	SELECT	Always
	READ BINARY	Always

Datenobjekte in EF.OD

Data-Object				DO.OD	Meaning
T	T	T	L	An-L-{30-L-04-L-V}	
A0				Variable [V]	Pointer: EF.PrKDF
	30				
		04		1F01	FID
A1					Pointer: EF.PuKDF
	30				
		04		1F02	FID
A4					Pointer: EF.CDF
	30				
		04		1F03	FID
A7					Pointer: EF.DCOD
	30				
		04		1F04	FID
A8					Pointer: EF.AODF
	30				
		04		1F05	FID

8.3.4 EF.PrKD aka EF.PrKDF

Objekt	EF.PrKD	Verzeichnisdatei für private Schlüssel
Objekttyp	Elementary File	
Dateistruktur	Transparent	
Dateikategorie	Working	
FID	'1F01'	
SFID	'01'	
Zugriffsrechte	Zugriffsart [AM]	Sicherheitsbedingungen [SC]
	SELECT	Always
	READ BINARY	Always

Datenobjekte in EF.PrKD

Data-Object						DO.PrKD	
Tag	Tag	Tag	Tag	Tag	L	30-L-{30-L-{0C-L-V-03-L-V-04-L-V}-30-{04-L-V-03-L-V-02-L-V}-A1-L-{30-L-{04-L-V}}-02-L-V}}	Meaning
30						Variable [V]	PKCS-15: Signature.Key X.509
	30						Key Information
		0C			13	'Signature Key'	Label
		03			2	06 80	Flag: Private
		04			1	02	AuthID: PIN2
	30						Common Key Attributes
		04			1	46	ID
		03			3	06 30 40	PKCS#15: Key Usage
		02			1	97	Key reference key 17
	A1						Private Key Attributes
		30					
			30				
				04	4	3F00DF02	Key path
			02		2	08 00	Length modulus
30						Variable [V]	PKCS-15: Decipher.Key
	30						Key Information
		0C			17	'Authorization Key'	Label

Data-Object						DO.PrKD	
Tag	Tag	Tag	Tag	Tag	L	30-L-{30-L-{0C-L-V-03-L-V-04-L-V}-30-{04-L-V-03-L-V-02-L-V}-A1-L-{30-L-{04-L-V}}-02-L-V}}	Meaning
		03			2	06 80	Flag: Private
		04			1	02	AuthID: PIN2
	30						Common Key Attributes
		04			1	47	ID
		03			3	06 74 40	PKCS#15: Key Usage
		02				96	Key reference key 16h
	A1						Private Key Attributes
		30					
			30				
				04	4	3F00DF02	Key path
			02		2	08 00	Length modulus

8.3.5 EF.PuKD aka EF.PuKDF

Objekt	EF.PuKD	Verzeichnisdatei für öffentliche Schlüssel
Objekttyp	Elementary File	
Dateistruktur	Transparent	
Dateikategorie	Working	
FID	'1F02'	
SFID	'02'	
Zugriffsrechte	Zugriffsart [AM]	Sicherheitsbedingungen [SC]
	SELECT	Always
	READ BINARY	Always

Datenobjekte in EF.PuKD

Data-Object						DO.PuKD	
Tag	Tag	Tag	Tag	Tag	L	30-L-{30-L{0C-L-V}-30-L-{04-L-V-03-L-V-02-L-V}-A1-L-{30-L-{30-L-{04-L-V}}-02-L-V}}}	Meaning
30							Signature Public Key X509
	30						
		0C			13	'Signature Key'	Label
	30						Common Key Attributes
		04				46h	ID
		03				06 03 00h	PKCS#15: Key Usage
		02			1	97h	Key reference: key 17h
	A1						Public RSA Key Attributes
		30					
			30				Container Public RSA Key
				04	2	'1F08'	Path to Public Key File
			02		2	08 00h	Length modulus
30						Variable [V]	PKCS#15: Authorization Key
	30						
		0C			17	'Authorization Key'	Label
	30						Common Key Attributes
		04			1	47h	ID
		03			3	06 8B 00h	PKCS#15: Key Usage
		02			1	96h	Key reference key 16h
	A1						Public RSA Key Attributes
		30					

Data-Object						DO.PuKD	
Tag	Tag	Tag	Tag	Tag	L	30-L-{30-L{0C-L-V}-30-L-{04-L-V-03-L-V-02-L-V}-A1-L-{30-L-{30-L-{04-L-V}}-02-L-V}}}	Meaning
			30				Path
				04	2	'1F07'	Path to Public Key File
			02		2	08 00h	Length modulus

8.3.6 EF.CD aka EF.CDF

EF.CD ist eine Verzeichnisdatei, in welcher auf die in der Zertifikatscontainerdatei EF.CERT enthaltenen X.509 Zertifikate referenziert wird. Bei jedem Abspeichern, Aktualisieren und Löschen eines X.509 Zertifikats im der Zertifikatscontainerdatei EF.CERT, muss auch das entsprechende Datenobjekt DO.CD in der Verzeichnisdatei EF.CD nachgeführt werden. Dies kann mit dem Kommando UPDATE BINARY mit einer Verifikation mittels der PIN2-Eingabe durch den Versicherten durchgeführt werden.

Objekt	EF.CD	Certificate Directory File
Objekttyp	Elementary File	
Dateistruktur	Transparent	
Dateikategorie	Working	
FID	'1F03'	
SFID	'03'	
Grösse	252 Bytes	
Zugriffsrechte	Zugriffsart [AM]	Sicherheitsbedingungen [SC]
	SELECT	Always
	READ BINARY	Always
	UPDATE BINARY	Authentifiziert mit PIN2

Datenobjekt in EF.CD

Data-Object						DO.CD	
Tag	Tag	Tag	Tag	Tag	L	30-L-{30-L-{0C-L-V-03-L-V}-30-L-{04-L-V}-A1-L-{30-L-{30-L-{04-L-V}}}}	Meaning
30						Variable [V]	x509Certificate
	30						CommonObjectAttributes
		0C			36	Cert1_KtMV_StGallen_SN_1234567890	Label: UTF8String
		03			1	'00'	Flags
	30						CommonCertificateAttributes
		04			1	'46'	iD Identifier: z.B. '46'
	A1						typeAttributes
		30					X509CertificateAttributes
			30				Path
				04	2	'1F06'	efidOrPath

8.3.7 EF.DCOD

Datencontainer zur Speicherung kleiner Anwendungstoken, "Interindustry Data Object Templates" (IDOs), usw. Das Schreiben in diesen Container kann mit dem Kommando UPDATE BINARY mit einer Verifikation mittels der PIN2-Eingabe durch den Versicherten durchgeführt werden.

Objekt	**EF.DCOD**	Data Container Object Directory
Objekttyp	Elementary File	
Dateistruktur	Transparent	
Dateikategorie	Working	
FID	'1F04'	
SFID	'04'	
Grösse	100 Bytes	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ BINARY	Always
	UPDATE BINARY	Authentifiziert mit PIN2

8.3.8 EF.AOD aka EF.AODF

Objekt	**EF.AOD**	**Authentication Object Directory**
Objekttyp	Elementary File	
Dateistruktur	Transparent	
Dateikategorie	Working	
FID	'1F05'	
SFID	'05'	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ BINARY	Always

Datenobjekt in EF.AOD

Data-Object					**DO.AOD**	
Tag	**Tag**	**Tag**	**Tag**	L	30-L-{30-L-{0C-L-V-03-L-V-04-L-V}-30-L-{04-L-V}-A1-L-{30-L-V-{03-L-V-0A-L-V-02-L-V-02-L-V-02-L-V-80-L-V-04-L-V}}}	**Meaning**
30					Variable [V]	PIN2: Object Information
	30					Common Object Attributes
		0C		4	'PIN2'	Label
		03		2	06 C0	Flag: Private, modifiable
		04		1	04	AuthID: PUK
	30					Common Auth. Object Attr.
		04		1	02h	AuthID: PIN2
	A1					Password Attributes
		30				
			03	3	04 0C 00	Flags, Padding
			0A	1	01	ASCII encodes Digits
			02	1	06	Min Length
			02	1	08	Stored PIN Length
			02	1	08	Max Length

Data-Object					DO.AOD	
Tag	**Tag**	**Tag**	**Tag**	L	30-L-{30-L-{0C-L-V-03-L-V-04-L-V}-30-L-{04-L-V}-A1-L-{30-L-V-{03-L-V-0A-L-V-02-L-V-02-L-V-02-L-V-80-L-V-04-L-V}}}	**Meaning**
			80	1	02	PIN2 Reference-Verify
			04	1	00	Pad Char
30					30-L-{30-L-{0C-L-V-03-L-V}-30-L-{04-L-V}-A1-L-{30-L-V-{03-L-V-0A-L-V-02-L-V-02-L-V-02-L-V-80-L-V}}}	PUK: Object Information
	30					Common Object Attributes
		0C		3	'PUK'	Label
		03		2	06 80	Flag: Private
	30					Common Auth. Object Attr.
		04		1	04	AuthID: PUK
	A1					Password Attributes
		30				
			03	3	04 3E 00	Flags
			0A	1	01	ASCII encodes Digits
			02	1	08	Min Length
			02	1	08	Stored PUK Length
			02	1	08	Max Length
			80	1	04	PUK Reference-Verify

8.3.9 EF.CERT

Diese Datei ist ein Datencontainer mit einer Grösse von 5120 Bytes, in dem gut 3 Stück X.509-Zertifikate abgespeichert werden können. Die Abspeicherung erfolgt PIN2-geschützt. Die Zertifikate können immer ausgelesen werden. Damit können bei kantonalen Modellversuchen unter der Einwilligung des Versicherten jederzeit entsprechende X.509-Zertifikate nachgeladen werden. Bei jedem Abspeichern, Aktualisieren und Löschen eines X.509 Zertifikats im der Zertifikatscontainerdatei EF.CERT, muss auch das entsprechende Datenobjekt DO.CD in der Verzeichnisdatei EF.CD nachgeführt werden.

Objekt	**EF.CERT**	Datencontainer für X.509 Zertifikate
Objekttyp	Elementary File	
Dateistruktur	Transparent	
Dateikategorie	Working	
FID	'1F06'	
SFID	'06'	
Grösse	5120 Bytes	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ BINARY	Always
	UPDATE BINARY	Authentifiziert mit PIN2

8.3.10 EF.PuK.X509

In diesem Datencontainer sind der Modulus sowie der Exponent des öffentlichen Schlüssels für die Signaturanwendung (z.B. Authentifizierung an einem Webportal, fortgeschrittene Signatur) als DER-kodiertes TLV-Datenobjekt einfach auslesbar abgespeichert. Dieser Schlüssel kann problemlos für einen CSR [Certificate Signing Request] bei einer X.509-PKI mittels einer entsprechenden Applikationen verwendet werden.

Objekt	EF.PuK.X509	Container mit öffentlichem Signaturschlüssel
Objekttyp	Elementary File	
Dateistruktur	Transparent	
Dateikategorie	Working	
FID	'1F08'	
SFID	'08'	
Grösse	270 Bytes	
Schlüssellänge	RSA - 2048 Bit	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ BINARY	Always

Datenobjekt in EF.PuK.X509

Data-Object			DO.PuK.X509	Meaning
T	T	L	30-L-{02-L-V-02-L-V}	
30			Variable [V]	
	02	0101h	00 xx xx xx xx .. .. xxh	modulus n, 00: signed integer
	02	03h	01 00 01h	exponent e

8.3.11 EF.PuK.DEC

In diesem Datencontainer sind der Modulus sowie der Exponent des öffentlichen Schlüssels für die Entschlüsselungsanwendung (z.B. Autorisierung, eRezept) als DER-kodiertes TLV-Datenobjekt einfach auslesbar abgespeichert. Dieser Schlüssel kann problemlos für einen CSR [Certificate Signing Request] bei einer X.509-PKI mittels einer entsprechenden Applikationen verwendet werden.

Objekt	EF.PuK.DEC	Container mit öffentlichem Autorisierungsschlüssel
Objekttyp	Elementary File	
Dateistruktur	Transparent	
Dateikategorie	Working	
FID	'1F07'	
SFID	'07'	
Grösse	270 Bytes	
Schlüssellänge	RSA - 2048 Bit	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	READ BINARY	Always

Datenobjekt in EF.PuK.DEC

Data-Object			DO.PuK.DEC	Meaning
T	T	L	30-L-{02-L-V-02-L-V}	
30			Variable [V]	
	02	0101h	00 xx xx xx xx .. .. xxh	modulus n, 00: signed integer
	02	03h	01 00 01h	exponent e

8.4 iEF.PrK.X509

Dieser nichtauslesbare private Schlüssel ist für Signaturanwendungen bestimmt. Die Signaturfunktion kann nur durch vorangehende korrekte PIN2-Verifikation angewandt werden.

Objekt	iEF.PrK.X509	Private Key für X.509 Signaturanwendungen
Objekttyp	Linear variable	
Dateikategorie	Internal	
FID	'0017'	
SFID	'17'	
Grösse	1224 Bytes	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	PSO COMPUTE DIGITAL SIGNATURE	Authentifiziert mit PIN2

8.4.1.1 Anwendung der Signaturfunktion

PSO COMPUTE DIGITAL SIGNATURE (RSA2048_PURE_SIG)						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	2Ah	9Eh	9Ah	000100h	tbsObject	0000h

8.4.1.2 tbsObject

tbsObject	To be signed object			
tbsObject	**Algorithmus**	**Hashfunktion**	**Paddingverfahren**	**Lc**
RSA-v1-5-SHA-1	RSA 2048	SHA-1	PKCS#1 v1.5 (SSL/TLS)	000100h
RSA-v1-5-SHA-256	RSA 2048	SHA-256	PKCS#1 v1.5 (SSL/TLS)	000100h
RSA-PSS-SHA-1	RSA 2048	SHA-1	PKCS#1 v2.1	000100h
RSA-PSS-SHA-256	RSA 2048	SHA-256	PKCS#1 v2.1	000100h

8.5 iEF.PrK.DEC

Dieser nichtauslesbare private Schlüssel ist für Entschlüsselungsanwendung (Autorisierung, eRezept usw.) bestimmt. Die Entschlüsselungsfunktion kann nur durch vorangehende korrekte PIN2-Verifikation angewandt werden.

Objekt	iEF.PrK.DEC	Private Key für X.509 Entschlüsselungsanwendungen
Objekttyp	Linear variable	
Dateikategorie	Internal	
FID	'0016'	
SFID	'16'	
Grösse	1224 Bytes	
Zugriffsrechte	**Zugriffsart [AM]**	**Sicherheitsbedingungen [SC]**
	SELECT	Always
	PSO DECIPHER	Authentifiziert mit PIN2

8.5.1.1 Anwendung der Entschlüsselungsfunktion

PSO DECIPHER (RSA2048_PURE)						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	2Ah	80h	86h	000100h	tbdObject	0000h

8.5.1.2 tbdObject

tbdObject	To be deciphered object			
tbdObject	**Algorithmus**	**Hashfunktion**	**Paddingverfahren**	**Lc**
RSA-v1-5-SHA-1	RSA 2048	SHA-1	PKCS#1 v1.5 (SSL/TLS)	000100h
RSA-v1-5-SHA-256	RSA 2048	SHA-256	PKCS#1 v1.5 (SSL/TLS)	000100h
RSA-PSS-SHA-1	RSA 2048	SHA-1	PKCS#1 v2.1	000100h
RSA-PSS-SHA-256	RSA 2048	SHA-256	PKCS#1 v2.1	000100h

8.6 PIN-Management für kantonale Modellversuche

8.6.1 Eingabe eines neuen PIN-Kodes

8.6.1.1 Eingabe des PUK

VERIFY (PUK)						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	20h	00h	04h	08h	PUK	-

8.6.1.2 Eingabe des neuen PINs

CHANGE REFERENCE DATA (PIN2)						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	24h	01h	02h	08h	neuer PIN	-

Neuer PIN (ISO 8859-1): nn nn nn nn nn nn nn nn (8 Stellen)

8.6.2 Zurücksetzen des Fehlbedienungszählers

8.6.2.1 Eingabe des PUK

VERIFY (PUK)						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	20h	00h	04h	08h	PUK	-

8.6.2.2 Fehlbedienungszähler Zurücksetzen

RESET RETRY COUNTER (PIN2)						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	2Ch	03h	12h	-	-	-

8.6.3 Anwendung der Signaturfunktion

8.6.3.1 Eingabe des PINs

VERIFY (PIN2)						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	20h	00h	02h	08h	PIN2	-

8.6.3.2 Signaturfunktion anwenden

PSO COMPUTE DIGITAL SIGNATURE (RSA2048_PURE_SIG)						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	2Ah	9Eh	9Ah	000100h	tbsObject	0000h

8.6.4 Abspeichern eines X.509-Zertifikats für kantonale Modellversuche

8.6.4.1 Datencontainer für X.509 Zertifikate selektieren

Terminal	SELECT (EF.CERT)					
CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	08h	00h	04h	DF021F06	-
Response Data						SW12
-						90 00h

8.6.4.2 Eingabe des PINs

VERIFY (PIN2)						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	20h	00h	02h	08h	PIN2	-
Response Data					SW12	
-					90 00h	

8.6.4.3 Abspeichern des X.509-Zerifikats

UPDATE BINARY (DO.CERT, X.509-Zertifikat)						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	D6h	Offset	Offset	Lc(DO.CERT)	(DO.CERT)	-
Response Data				SW12		
-				90 00h		

8.6.4.4 Selektieren der Certificate Directory File

Terminal	SELECT (EF.CD)					
CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	08h	00h	04h	DF021F03	-
Response Data						SW12
-						90 00h

8.6.4.5 Abspeichern des Zertifikatsreferenzdatenobjekts

UPDATE BINARY (DO.CD)						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data**	**Le**
00h	D6h	Offset	Offset	Lc(DO.CD)	(DO.CD)	-
Response Data				SW12		
-				90 00h		

8.7 PKCS#11 Middleware

Mit einer PKCS#11-Testlibrary für die SASIS-Versichertenkarte wurden verschiedene Tests basierend auf der vorhandenen PKCS#15 bzw. ISO/IEC 7816-15 Dateistruktur durchgeführt. Es wurden in einem grösseren Netzwerk Active-Directory-Authentifizierungen sowie Authentifizierungen auf Webportaldienste mittels X.509-Zertifikate und SSL/TLS erfolgreich geprüft. Als Plattformen sind Windowsbetriebssysteme (Win2000, XP, Win7.0), MAC OS X- und LINUX-Betriebssysteme geeignet.

9 Statuswörter

Die zurückgegebenen Statuswörter entsprechen [7816-4] Table 6:

SW1	SW2	Bedeutung
62h	81h	Ein Teil der zurückgegebenen Daten kann korrupt sein.
	82h	Dateiende erreicht bevor Le-Bytes gelesen werden konnten.
	83h	Angewählte Datei deaktiviert.
	84h	FCI-Daten nicht korrekt formatiert.
	85h	Angewählte Datei terminiert.
63h	00h	Prüfung fehlgeschlagen.
	Cxh	Prüfung fehlgeschlagen. Wert x gibt die noch erlaubten Versuche an.
64h	00h	Ausführungsfehler (z. B. konnte eine erzeugte Datei nicht aktiviert werden).
65h	00h	Ausführungsfehler (keine weitere Informationen).
	81h	Speicherfehler (fehlerhafte Änderung).
66h	00h	Fehler bei sicherheitsrelevanten Operationen, z. B. kryptographische Berechnungen.
67h	00h	Falsche Länge.
68h	81h	Logische Kanäle nicht unterstützt.
	82h	Gesicherte Datenübertragung nicht unterstützt.
	84h	Kommandoverknüpfung nicht unterstützt.
69h	81h	Kommando nicht kompatibel zur Dateistruktur.
	82h	Notwendiger Sicherheitsstatus nicht erfüllt.
	83h	Authentifizierungsmethode blockiert.
	84h	Referenzierten Daten nicht benutzbar.
	85h	Anwendungsbedingungen nicht erfüllt.
	86h	Kommando nicht erlaubt, z. B. kein angewähltes EF.
	87h	Erwartete Objekte bei gesicherter Datenübertragung nicht vorhanden.
	88h	Falsche Objekte bei gesicherter Datenübertragung.
6Ah	00h	Falsche Parameter P1-P2.
	80h	Falsche Parameter im Kommandodatenfeld, z. B. Fehler im Kryptogramm.
	81h	Funktion wird nicht unterstützt.
	82h	Datei oder Applikation nicht gefunden.
	83h	Eintrag nicht gefunden.
	84h	Nicht genügend Speicherplatz.
	85h	Lc nicht konsistent mit TLV-Stuktur.
	86h	Falsche Parameter P1-P2.
	87h	Lc nicht konsistent mit P1-P2.
	88h	Referenzierte Daten nicht gefunden.
	89h	Datei bereits vorhanden.
	8Ah	Dateiname (AID) bereits vorhanden.
6Bh	00h	Falsche Parameter (Offset ausserhalb der Datei).
6Dh	00h	INS-Byte nicht unterstützt oder ungültig.
6Eh	00h	CLA-Byte nicht unterstützt.
6Fh	00h	Allgemeiner Fehler (keine weitere Informationen).