

Secure Interoperable ChipCard Terminal

SICCT

V 1.3.0

2025

Impressum

Herausgeber:

Bundesverband IT-Sicherheit e.V. (TeleTrust)
Chausseestraße 17
10115 Berlin
Tel.: +49 30 4005 4310
E-Mail: info@teletrust.de
<https://www.teletrust.de>

© 2025 TeleTrust

V 1.3.0

Autoren

Holger Ahl	gematik GmbH
Jürgen Atrott	TÜV Informationstechnik GmbH
Thomas Bründl	HID Global GmbH
Maximilian Bukow	gematik GmbH
Volker Czmok	CCV Deutschland GmbH
Pascal Dörjes	achelos GmbH
Klaus Erichsen	Ingenico Healthcare GmbH
Dr. Alfred Fiedler	gematik GmbH
Christian Hartung	TÜV Informationstechnik GmbH
Erik Fischelschweiger	gematik GmbH
Stefan Frenzel	HID Global GmbH
Uwe Harasko	Cherry (vormals ZF Electronics) GmbH
Roland Kothe	Verifone GmbH
Jörg Kühnl	ZF Electronics GmbH
Andreas Kilger	Cherry (vormals ZF Electronics) GmbH
Lothar Mawick	Verifone GmbH
Tomas Müller	CCV Deutschland GmbH
Dr. Klaus Leistner	CCV Deutschland GmbH
Torsten Maykranz	Identiv (vormals SCM Microsystems) GmbH
Frank Neumann	Gemalto GmbH
Frank Osthoff	Worldline Healthcare (vormals Ingenico Healthcare) GmbH
Uwe Schnabel	HID Global GmbH
Sebastian Schraml	Cherry Digital Health (vormals Cherry vormals ZF Electronics) GmbH
Thore Simonides	Worldline Healthcare (vormals Ingenico Healthcare) GmbH
Dr. Holger Mühlbauer	TeleTrust – Bundesverband IT-Sicherheit e.V.

Contents

Page

1	Anwendungsbereich	9
2	Normative Verweisungen	9
3	Anforderungen und Systemvoraussetzungen	9
3.1	Architekturmodell	10
3.2	Kommunikationsmodell	11
3.3	Allgemeine Applikationsbeschreibung	11
3.4	Sicherheitsanforderungen	13
3.5	Benutzerrollen und CT Session	13
4	Beschreibung der Schnittstelle zur Karte	15
4.1	Mechanische Anforderungen	15
4.1.1	Geeignete kontaktbehaftete Kartentypen	15
4.1.2	Anforderungen an die Kartenkontaktierungen für Größe ID-1	15
4.1.3	Anforderungen an die Kartenkontaktierungen für Größe ID-000	16
4.1.4	Anforderungen bei Verwendung von kontaktlosen Karten	16
4.2	Elektrische Anforderungen	17
4.2.1	Elektrische Anforderungen für kontaktbehaftete Karten	17
4.2.2	Reset-Verhalten und ATR-Bearbeitung	17
4.2.3	Unterstützung von synchronen Karten	18
4.2.4	Elektrische Anforderungen für kontaktlose Karten	18
4.3	Protokolle	18
4.3.1	Protokolle für kontaktbehaftete Karten	18
4.3.2	Protokolle für kontaktlose Karten	19
5	Command Set	20
5.1	SICCT Command Structure	20
5.2	Command Header	21
5.2.1	CLASS - Byte	21
5.2.2	INSTRUCTION - Byte	22
5.2.3	Parameter P1	22
5.2.4	Parameter P2	22
5.3	Command Body	23
5.3.1	Data Length Lc und Expected Response Length Le	23
5.3.2	Command Data Field	23
5.4	SICCT Response Structure	24
5.4.1	Response Trailer	24
5.4.2	Checking Error / Prüfungsfehler	26
5.4.3	Response Body	26
5.5	Command Set Overview and General Return Codes	27
5.5.1	Adressierung und Transport von Kommandos	27
5.5.2	Ausführung und Wirkweise von Kommandos	28
5.5.3	Abbruch der Kommandoausführung	28
5.5.4	Nebenläufige Kommandobearbeitung	29
5.5.5	Betriebsmodi	29
5.5.6	Betriebsmodus 'BCS' und Command Set	29
5.5.7	Betriebsmodus 'SICCT' und Command Set	31
5.5.8	Unterstützte Chipkarten	33
5.5.9	Functional Units	34
5.5.10	Data Objects	39
5.6	General Handling Instructions für den SICCT Betriebsmode	77
5.6.1	Handling von Display Messages im SICCT-Mode	77
5.7	Handling von Tastatureingaben (Keypad-Entries) im SICCT-Mode	79
5.8	Command SICCT SELECT CT MODE	80
5.8.1	Funktion	80
5.8.2	Anwendungsbedingungen	80

5.8.3	Command Structure	81
5.8.4	Data Objects	83
5.8.5	Response Structure	83
5.8.6	Status-Codes SW1SW2	84
5.9	Command SICCT CONTROL COMMAND	85
5.9.1	Funktion	85
5.9.2	Anwendungsbedingungen	85
5.9.3	Command Structure	86
5.9.4	Data Objects	87
5.9.5	Response Structure	87
5.9.6	Status-Codes SW1SW2	89
5.10	Command SICCT INIT CT SESSION	91
5.10.1	Funktion	91
5.10.2	Anwendungsbedingungen	91
5.10.3	Command Structure	92
5.10.4	Data Objects	93
5.10.5	Response Structure	94
5.10.6	Status-Codes SW1SW2	95
5.11	Command SICCT CLOSE CT SESSION	96
5.11.1	Funktion	96
5.11.2	Anwendungsbedingungen	96
5.11.3	Command Structure	97
5.11.4	Data Objects	98
5.11.5	Response Structure	98
5.11.6	Status-Codes SW1SW2	99
5.12	Command SICCT RESET CT / ICC	100
5.12.1	Funktion	100
5.12.2	Anwendungsbedingungen	102
5.12.3	Command Structure	103
5.12.4	Data Objects	105
5.12.5	Response Structure	105
5.12.6	Status-Codes SW1SW2	106
5.13	Command SICCT REQUEST ICC	108
5.13.1	Funktion	108
5.13.2	Anwendungsbedingungen	109
5.13.3	Command Structure	109
5.13.4	Data Objects	112
5.13.5	Response Structure	112
5.13.6	Status-Codes SW1SW2	112
5.14	Command SICCT EJECT ICC	115
5.14.1	Funktion	115
5.14.2	Anwendungsbedingungen	115
5.14.3	Command Structure	116
5.14.4	Data Objects	117
5.14.5	Response Structure	118
5.14.6	Status-Codes SW1SW2	118
5.15	Command SICCT GET STATUS	121
5.15.1	Funktion	121
5.15.2	Anwendungsbedingungen	122
5.15.3	Command Structure	122
5.15.4	Data Objects	125
5.15.5	Response Structure	125
5.15.6	Status-Codes SW1SW2	126
5.16	Command SICCT SET STATUS	128
5.16.1	Funktion	128
5.16.2	Anwendungsbedingungen	129
5.16.3	Command Structure	129
5.16.4	Data Objects	131
5.16.5	Response Structure	131
5.16.6	Status-Codes SW1SW2	131
5.17	Command SICCT INPUT	134
5.17.1	Funktion	134
5.17.2	Anwendungsbedingungen	136

5.17.3	Command Structure	137
5.17.4	Data Objects	139
5.17.5	Response Structure	140
5.17.6	Status-Codes SW1SW2	140
5.18	Command SICCT OUTPUT	143
5.18.1	Funktion	143
5.18.2	Anwendungsbedingungen	143
5.18.3	Command Structure	144
5.18.4	Data Objects	146
5.18.5	Response Structure	146
5.18.6	Status-Codes SW1SW2	146
5.19	Command SICCT PERFORM VERIFICATION	149
5.19.1	Funktion	149
5.19.2	Anwendungsbedingungen	149
5.19.3	Command Structure	152
5.19.4	Data Objects	155
5.19.5	Response Structure	156
5.19.6	Status-Codes SW1SW2	156
5.20	Command SICCT MODIFY VERIFICATION DATA	159
5.20.1	Funktion	159
5.20.2	Anwendungsbedingungen	159
5.20.3	Command Structure	163
5.20.4	Data Objects	165
5.20.5	Response Structure	166
5.20.6	Status-Codes SW1SW2	166
5.21	Command SICCT COMFORT AUTHENTICATION	169
5.21.1	Funktion	169
5.21.2	Anwendungsbedingungen	169
5.21.3	Command Structure	170
5.22	Command SICCT COMFORT ENROLL	171
5.22.1	Funktion	171
5.22.2	Anwendungsbedingungen	171
5.22.3	Command Structure	172
5.23	Command SICCT DOWNLOAD INIT	173
5.23.1	Funktion	173
5.23.2	Anwendungsbedingungen	176
5.23.3	Command Structure	176
5.23.4	Data Objects	178
5.23.5	Response Structure	178
5.23.6	Status-Codes Sw1SW2	179
5.24	Command SICCT DOWNLOAD DATA	179
5.24.1	Funktion	179
5.24.2	Anwendungsbedingungen	180
5.24.3	Command Structure	180
5.24.4	Data Objects	181
5.24.5	Response Structure	182
5.24.6	Status-Codes Sw1SW2	183
5.25	Command SICCT DOWNLOAD FINISH	184
5.25.1	Funktion	184
5.25.2	Anwendungsbedingungen	184
5.25.3	Command Structure	184
5.25.4	Data Objects	186
5.25.5	Response Structure	186
5.25.6	Status-Codes Sw1SW2	187
6	Schnittstellenbeschreibung zum Host	188
6.1	Allgemeines	188
6.2	Anschlussmöglichkeit Ethernet	188
6.2.1	Netzwerk Protokoll Festlegungen	188
6.2.2	Adress- und Namensvergabe	189
6.2.3	Service Discovery/Announcement	190
6.2.4	Kommandotransport und Namensvergabe	196
6.2.5	Auslieferungszustand	206

6.3	Terminal Managementverfahren	207
6.3.1	Allgemeine Anforderungen an den Ablauf des Terminalmanagement	207
6.3.2	Managementschnittstellen.....	207
6.3.3	Firmware Download.....	209
6.4	Sicherer Kanal.....	209
6.4.1	Sicherheitsprotokolle	209
6.4.2	Zertifikate	210
6.4.3	Sicherung der netzwerkbasieren Verbindungen.....	211
7	Schnittstellenbeschreibung zum Benutzer	212
7.1	Tastatur	212
7.2	Display/Anzeige	213
7.2.1	Transparente Displayansteuerung.....	213
7.2.2	Leuchtdioden	214
7.3	Tongeber.....	214
7.4	Biometrische Einheiten	215
7.5	Benutzerauthentisierung bei der Komfortsignatur	215
7.5.1	Anlernfunktion	215
7.6	Sicherer Modus	215
8	Sicherheit.....	216
8.1	Allgemeine Betrachtungen	216
8.2	Sicherheitsziele.....	217
8.3	Bedrohungen.....	217
8.4	Generische Maßnahmen	217
8.5	Annahmen	218
8.6	Sicherheitsmechanismen	219
9	Anmerkungen zum Dokumentenstand.....	221
9.1	Dokumentenstand und Haftungsausschluss	221
9.2	Errata-Beschreibung	221
9.3	Zu bearbeitende Punkte	221
10	Abkürzungen	221
11	Referenzen.....	223
Anhang A (Normativ) ISO / EMV Vergleichstabelle		226
Anhang B (Informativ) eCard Strategie des Bundes / ECard Framework.....		234
1	Architektur.....	234
2	Application Layer.....	235
3	eCard Layer	235
4	Reader Layer	236
5	Umsetzung.....	236

Vorwort

Diese Spezifikation wurde von der Arbeitsgruppe "SICCT" des Bundesverbandes IT-Sicherheit e.V. (TeleTrusT) erarbeitet.

Einleitung

Kartenterminals stellen die Betriebsumgebung für Chipkarten und deren Anwendung dar.

Die Anforderungen an diese Geräte orientieren sich zumeist an technisch-funktionalen und sicherheitstechnischen Anforderungen, welche durch spezifische Einsatzumfelder vorgegeben werden. Neben proprietären bzw. geschlossenen Branchenlösungen existieren eine Reihe von neutralen Spezifikationen für universelle oder applikationsunabhängige Terminalplattformen.

Bedingt durch unterschiedliche technische wie sicherheitstechnische Anforderungen, die diverse Einsatzumgebungen teils aus organisatorisch-rechtlichen Vorgaben bedingen, kann die Existenz eines allumfassenden Standards, der alle Eventualitäten erfüllt, nur fraglich sein. Im Resultat ist davon auszugehen, dass stets Produkte nach unterschiedlichen Ausprägungen und in einem angemessenen Kostenrahmen vorhanden sein müssen.

Vor dem Hintergrund eines stetig wachsenden Anwendungsumfelds für Chipkarten in diversen Industrieanwendungen benötigen insbesondere Systemintegratoren und Betreiber von Chipkartenanwendungen Kartenterminalplattformen, welche zunehmend auf offenen und applikations-unabhängigen Standardplattformen basieren und damit einen hohen Standardisierungsgrad aufweisen sollen. Es besteht hier die Notwendigkeit, die Aufwände für die Integration und den Betrieb gering zu halten, aber aus einer fixierten Standardisierung, erforderliche Gerätevarianten ableiten zu können.

Zweck dieser Spezifikation ist die Definition und Darstellung eines neuen Basiskonzepts SICCT (Secure Interoperable Chip Card Terminal) für applikationsunabhängige Chipkartenterminals anhand von bestehenden Normen und Industriestandards zu Kartenterminals und Chipkarten. Die technischen Grundlagen hierfür stellen ausschließlich anerkannte und bewährte Chipkarten- und Kartenterminal-Standards dar, welche bei Systemintegratoren und Anwendern besondere Verbreitung gefunden haben.

Als Resultat eines Harmonisierungsprozesses stellt dieses Dokument eine technisch-funktionale Schnittmenge über bereits bestehende und bewährte Technologien sowie deren Sicherheitsmechanismen und Integrationsmöglichkeiten dar. Des Weiteren sollte diese Spezifikation auch Ergänzungen für neue Anwendungsfälle bieten, welche bisher in den bestehenden Standards fehlen oder nur bedingt dargestellt sind.

Der Prozess und die Pflege dieser offenen Spezifikation wird von der Arbeitsgruppe "SICCT", im IT-Sicherheitsverband TeleTrusT vorgenommen. Im Ergebnis stellt das SICCT-Basiskonzept eine interoperable und sichere Umgebung für diverse Anwendungsgebiete mit schutzwürdigen Daten dar, und wird derart generisch sein, dass unterschiedliche Ausprägungen und Anschlussarten von Kartenterminalprodukten möglich werden, und Anwender unter vergleichbaren und austauschbaren Kartenterminalplattformen wählen können.

Eine definierte Funktionalität ermöglicht prinzipiell den Betrieb aller derzeit verwendeten und standardisierten Arten von Chipkarten

- Speicherchipkarten (Memory Card);
- Prozessorchipkarte (Smart Card)

über kontaktbehaftete und kontaktlose Übertragungswege.

Vor dem Hintergrund eines als nachweisbar sicher einzustufenden Systems, ist es das Ziel dieser Spezifikation, insbesondere eine technische Harmonisierung in Form einer Eingrenzung von technischer Funktionalität, Variabilität und Sicherheitsanforderungen, um Systemintegratoren und Betreibern

- die Integration,
- die Konfiguration und
- den (sicheren) Betrieb von Kartenterminals sowie Chipkartenanwendungen

zu erleichtern.

Bekannte und gängige Sicherheitsanforderungen an die Terminalkomponente müssen erfüllt werden, so dass diese Geräte für sensitive Anwendungsfälle wie digitale Signaturanwendungen im Gesundheitswesen gleichermaßen einsetzbar sind. Für identifizierte Anwendungsbereiche kann ein jeweiliges Erweiterungsdokument eine Konkretisierung der Terminalausprägung vom Basiskonzept ableiten und fixieren. Die TeleTrusT-AG "SICCT"

hat das Anliegen, spezifische Ableitungen in Kooperation mit Anwendergruppen zu erarbeiten, und diese als separate Dokumente sukzessive zu veröffentlichen. Eine erste Ausprägung eines SICCT-Terminals findet sich in der Standardisierung [gemSpec_KT] einer Kartenterminalplattform entsprechend den Anforderungen des Deutschen Gesundheitswesens nach Vorgaben der nationalen Vorgabeinstanz gematik zur Einführung einer elektronischen Gesundheitskarte.

Unter Berücksichtigung dieser Basisspezifikation konzentrieren sich Kartenterminalhersteller und Anwenderindustrien von SICCT-Terminals auf 'best practice' - Lösungen zur Steigerung von Interoperabilität bei der Anwendung und Integration von Chipkarten in IT-Lösungen.

Applikationsentwickler profitieren von einer erweiterten Funktionalität sowie von besonderen Abfragemöglichkeiten hinsichtlich implementierter Gerätere Ressourcen und Funktionen.

Anwender und Betreiber sollen erkennbar gleichartige Produkte unter nachweisbaren Sicherheitskriterien identifizieren und vergleichen können, um für den individuellen Anwendungsfall ein interoperables Gerät einsetzen zu können.

Insgesamt soll dieses Basiskonzept helfen, die Integration von Kartenterminals so weit zu vereinfachen, dass eine verbesserte Interoperabilität unter diversen Lieferanten, Modellen und Ausbaustufen erzielt wird, wie es bei anderen standardisierten IT-Komponenten selbstverständlich geworden ist.

Hinsichtlich der funktionalen Betrachtung wird von einem applikationsunabhängigen und multifunktionalen Kartenterminal ausgegangen, welches eine einheitliche logische Sichtweise auf diverse Chipkartenarten realisiert. Multifunktionalität bedeutet hierbei die technische Fähigkeit, diverse Typen von Chipkarten verarbeiten zu können. Applikationsunabhängigkeit bedeutet hierbei den Verzicht auf proprietäre und möglicherweise branchenspezifische Vorgaben zugunsten einer abstrakten und universell einsetzbaren Funktionalität.

Mittels einer separaten Betrachtung sowie einer Analyse von Anforderungen einzelner Einsatzumgebungen können Zuordnungen zu den Eigenschaften des Basiskonzepts und Restriktionen erfolgen. Generell können spezifische Kartenterminalausprägungen aus dem Basiskonzept abgeleitet und definiert werden, ohne das Prinzip des interoperablen Basiskonzepts zu verlassen.

Im Resultat lässt diese Vorgehensweise eine vereinfachte Systemarchitektur zu, welche unterschiedliche Kartenterminalausprägungen und deren Integration für den Einsatz in diversen Einsatzumgebungen erlaubt.

Berücksichtigung findet dabei auch die eCard-Strategie zur Unterstützung der flächendeckenden Einführung von Chipkarten im Bereich der Bundesverwaltung welche das Bundeskabinett am 09.03.2005 beschlossen hat. Diese Strategie ist eine nationale Konkretisierung internationaler Bestrebungen auf Grundlage der ISO/IEC 24727 Normenreihe.

Wesentliche Stützpfeiler dieser Strategie sind die elektronische Authentisierung und die qualifizierte elektronische Signatur, die auf Chipkarten unterschiedlicher Ausprägung zum Einsatz kommen sollen. Für die eCard-Strategie sind insbesondere die folgenden Projekte von Bedeutung:

- die elektronische Gesundheitskarte (eGK)
- der neue Personalausweis (nPA)
- der elektronische Reisepass (ePass)
- die elektronische Steuererklärung (Elster)
- das Jobcard-Verfahren.

Durch das in Anhang B dargestellte eCard-Framework wird ein einheitlicher technischer Rahmen zur Umsetzung der eCard-Strategie geschaffen. Insbesondere wird durch das eCard-Framework die interoperable Umsetzung von Signatur- und Authentisierungsanwendungen basierend auf unterschiedlichen Chipkarten ermöglicht.

1 Anwendungsbereich

Diese Spezifikation beschreibt primär ein generisches Basiskonzept für eine Chipkartenterminalplattform, dessen generische SICCT-Systemarchitektur Plattformunabhängigkeit und Interoperabilität bei Einhaltung allgemeiner Sicherheitsstandards bieten soll. Im Ergebnis entsteht ein interoperabler und sicherer Kontext für den Betrieb von diversen Chipkartenarten.

Das "Secure Interoperable Chip Card Terminal" (SICCT) ist wegen einer generischen Systemarchitektur ein universell einsetzbares Chipkartenterminal, mit der Möglichkeit, Kartenterminals über diverse Anschlussmöglichkeiten, wie. z.B. RS232, USB, Ethernet 802.3 etc. zu betreiben.

2 Normative Verweisungen

Die Anforderungen dieser Spezifikation ergeben sich aus einer beabsichtigen (partiellen) Konformität und Harmonisierung der folgenden Normen und Standards

- für kontaktbehaftete Chipkarten ISO 7816
- für kontaktlose Chipkarten ISO 14443 (Proximity Cards, PICC, RFID token)
- der Industriespezifikation MKT
- der Industriespezifikation PC/SC
- der Industriespezifikation EMV

3 Anforderungen und Systemvoraussetzungen

In dem folgenden Text werden die zu erfüllenden funktionalen und nicht funktionalen Anforderungen an die SICCT-Basisplattform aufgelistet sowie Voraussetzungen an die beteiligten Systemkomponenten beschrieben. Tabelle 1 listet generelle Anforderungen.

Tabelle 1: Generelle Anforderungen

Generelle Anforderungen an ein SICCT - Kartenterminal		
GA_1	Sicherer Betrieb der Chipkarte	Die wichtigste Anforderung ist der fehlerfreie und unversehrte Betrieb der Chipkarte(n). Das Kartenterminal muss die Chipkarte(n) elektrisch und mechanisch schützen und einen stabilen Betrieb gewährleisten.
GA_2	Normgerechte elektrische Eigenschaften des Kartenterminals	Die vorrangige Anforderung ist die Bereitstellung einer normgerechten Funktionalität an allen elektrischen Anschlüssen (Interfaces) zur Außenwelt. Dies muss die Chipkarten-Interfaces (Kontaktiereinheiten) sowie Anschlüsse und Kommunikationsports (z. B. RS-232, USB, Ethernet 802.3) zu anderen IT-Systemen mit einbeziehen.
GA_3	Normgerechte Kommunikations-Protokolle	Die Kommunikation zu anderen IT - Komponenten (u. a. zur Chipkarte, Steuerrechner) muss auf der Basis standardisierter Protokollverfahren erfolgen.
GA_4	Interoperabilität	Der Betrieb eines Kartenterminals soll unabhängig von Plattform und Betriebssystem des Steuerrechners (HOST) sein. Diese Interoperabilität kann durch die Wahl eines Kartenterminal-APIs oder durch die Definition eines plattformunabhängigen und einheitlichen Kartenterminal-Kommandosatzes erreicht werden.
GA_5	Multifunktionalität	Die Multifunktionalität soll sich im Wesentlichen auf die Unterstützung verschiedener Chipkartentypen beziehen. Dieses umfasst die Verpflichtung, Prozessorchipkarten nach den üblichen spezifischen Protokollen betreiben zu können. Die Unterstützung von Speicherchipkarten und sowie von RFID-Token (PICC) kann erfolgen.

Generelle Anforderungen an ein SICCT - Kartenterminal		
GA_6	Applikationsunabhängigkeit.	Die Applikationsunabhängigkeit bedeutet eine technische Universalität zur Realisierung beliebiger Verarbeitungsprozesse und Szenarien im Zusammenhang mit Chipkartenapplikationen.
GA_7	IT-Sicherheit	Das Kartenterminal muss eine sichere Umgebung für Chipkarten bereitstellen und soll dem Kartenanwender einen erkennbar sicheren und vertrauenswürdigen Betriebszustand signalisieren.
GA_8	Anwenderauthentisierung	Das Kartenterminal muss technische Funktionen zur Anwenderauthentifizierung (Keypad, biometrische Sensorik, Kontaktlos Leser) bieten. Authentisierungsdaten dürfen vom Kartenterminal nicht an das Host-System weitergegeben, und intern fallbezogen angewendet aber nicht gespeichert werden.
GA_9	Basisfunktionalität	Das Kartenterminal muss einem angeschlossenen IT-System eine definierte Basisfunktionalität zum Umgang mit Chipkartenapplikationen zur Verfügung stellen.
GA_10	Betriebssicherheit	Das Kartenterminal muss für Dauerbetrieb geeignet sein, sowie einem Betreiber / Nutzer eine hohe Verfügbarkeit (z. B. geringe Ausfallquote, zugesicherte MTBF) bieten.
GA_11	Validierungsvoraussetzungen	Das Kartenterminal soll hinsichtlich der genannten Anforderungen und entsprechend dem Einsatzumfeld und der Einsatzumgebung durch akkreditierte Prüf- oder Zulassungsstellen validierbar sein.
GA_12	Updateverfahren für Kartenterminal -Software	Die Geräte - Firmware (FW) kann über die Lebensdauer des Kartenterminals gesichert aktualisiert werden, um • die Einführung neuer Funktionen, • die Beseitigung von erkannten Sicherheitsrisiken oder • eine generelle Fehlerbehebung vornehmen zu können.

3.1 Architekturmodell

Die Systemarchitektur muss generell vorsehen, dass über logische Funktionseinheiten, sog. Functional Units, alle verfügbaren Kontaktiereinheiten und Ressourcen des Kartenterminals an die externe Welt abgebildet werden. Den Zugriff auf jegliche Ressourcen muss das Terminal über eine definierte Kommandoschnittstelle kontrollieren und gewähren. Der Zustand und die Verfügbarkeit von Ressourcen müssen durch das Kartenterminal intern überwacht, kontrolliert und in Form von Zustandsvariablen an die externe Welt gemeldet werden können. Einer bestimmten Anzahl diskreter Kartenterminal- und Kartenzustände werden hierfür einheitliche Werte bzw. Statuscodes zugeordnet.

Im Zentrum der Systemarchitektur muss eine interne Interpreterlogik stehen, welche über ein Kommunikationsmodul Kartenterminal- (Basisdienste) sowie Kartenkommandos empfangen und zur Ausführung bringen kann.

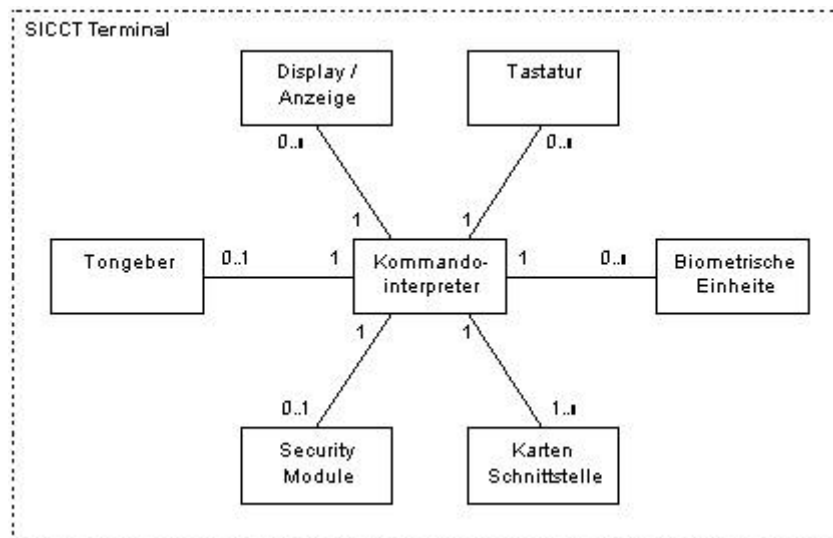


Abbildung 1: Modell SICCT-Terminal

3.2 Kommunikationsmodell

Das SICCT-Kommunikationsmodell stellt aus Sicht des Netzwerks ein Client-Server-Kommunikationsmodell dar, wobei das Kartenterminal in der Rolle des Servers Dienste anbieten soll. Aus Sicht der Applikation bzw. Ansteuerung des Kartenterminals und der Chipkarten liegt ein Master-Slave-Kommunikationsmodell vor. Die Ansteuerung von Ressourcen und Chipkarten bzw. Applikationen auf Chipkarten muss ausschließlich über die verfügbaren Basisdienste erfolgen. Eine korrespondierende Applikation außerhalb der Chipkarte kann entweder im Kontext einer externen Entität (HOST, z.B. PC) oder als interne Applikation im Kartenterminal ablaufen. Als Alternative kann auch eine Mischform aus externem und internem Applikationsteil bestehen.

Das Grundprinzip der Kommunikation stellt allein der geregelte Austausch interoperabler und einheitlicher Kartenterminal- und Chipkartenkommandos dar.

Dank der zentralen und interoperablen Interpreterlogik kann das Kommunikationsmodul unabhängig von Kommunikationsbus und Transportprotokoll gehalten werden.

Zum Transport von Karten- wie Chipkartenkommandos stehen somit diverse mögliche Transportprotokolle zur Verfügung, die eine gesicherte Übertragung von Nachrichten, d.h. Kommando- wie Antwortsequenzen, erlauben.

Hinsichtlich der Möglichkeiten, ein SICCT an bzw. in ein steuerndes System (HOST) zu integrieren, bestehen keine Vorgaben oder Restriktionen, sofern definierte SICCT-Kommandos und Antworten nach dem Request / Response-Verfahren ausgetauscht werden können.

Da dieses Prinzip von nahezu allen bekannten Transportprotokollen geleistet und durch Smartcard-Frameworks oder deren Treibertopologien eingehalten wird, kann eine explizite Vorgabe respektive des Transportprotokolls oder eines Frameworks entfallen.

3.3 Allgemeine Applikationsbeschreibung

Es wird angenommen, dass eine steuernde Instanz die korrespondierende Applikationsstruktur auf einer angesteuerten Chipkarte kennt und einen separaten Applikationskontext unterhält, der den Zustand des Kartenterminals, der Chipkarten sowie die Zustände der selektierten Chipkartenapplikation(en) im Umfeld einer konkreten Einsatzumgebung abbildet.

Durch dieses Prinzip kann das Kartenterminal als transparente Komponente wirken und benötigt prinzipiell keine Kenntnis des exakten Applikationskontextes.

Das Kartenterminal überwacht dabei die in Tabelle 2 gezeigten Zustände:

Tabelle 2: Zustände eines SICCT

Vom SICCT verwaltete Zustände		
ZUE_1	den eigenen logischen Gerätezustand	1) Versionsstand 2) Betriebsart 3) Powermanagement 4) sicherer Betriebsmode 5) transparenter Betriebsmode
ZUE_2	den logischen Zustand der Chipkartenkontaktiereinheiten	1) leer 2) Chipkarte steckt 3) Chipkarte bereit 4) Chipkarte aktiviert 5) Chipkarte ausgeworfen
ZUE_3	den elektrischen Zustand der Chipkarten bzw. Kontaktiereinheiten	1) POWERED 2) UNPOWERED
ZUE_4	den logischen Zustand der Chipkarten	1) Typinformation 2) Betriebsart: specific / negotiable 3) Transportprotokoll
ZUE_5	die Verwaltung und Synchronisation beim Zugriff auf Kartenterminalressourcen wie z.B. Display, Keypad, Biometriesensor	1) Eingabe von Authentifikationsdaten
ZUE_6	die Kommunikation zwischen steuernder Entität zum Kartenterminal und / oder Chipkarten.	1) Protokoll 2) Eventing 3) CT-Session 4) Monitoring

Für alle SICCT-Terminals sollen die in Tabelle 3 gezeigten Vorgaben verpflichtend gelten.

Tabelle 3: Generelle Vorgaben für SICCT

APP_1	SICCT-Kartenterminals können als autarke LAN-gekoppelte Geräte mit eigener TCP/IP-Kommunikation und Kommandointerpreter, und sowohl als einfacheres Kartenterminal an einem Peer-To-Peer-Bus (USB, RS232, ...), welches über eine Proxy-SW-Komponente dieselben Funktionen abbildet, erstellt sein.	Topologie
APP_2	Netzwerkfähige SICCT-Terminals müssen ein Kommunikationsmodul mit einheitlichen Funktionen zur Adressierung und zum Transport über das Standard - TCP/IP – Kommunikationsprotokoll besitzen.	Adressierung
APP_3	Für Kartenterminals, die an ein LAN angeschlossen sind, und prinzipiell netzwerkweit sichtbar sind, können über die Kombination aus IP-Adresse und Portnummer angesprochen werden.	Adressierung
APP_4	SICCT-Kartenterminals können einen oder mehrere Betriebsmodi anbieten, welche jeweils einem Kommandointerpreter oder Kommandosatz an Basisfunktionen zugeordnet sein können.	Kommandosatz
APP_5	SICCT-Kartenterminals müssen zur Ansteuerung des Kartenterminals sowie der Chipkartenslots eine Kommandoschnittstelle anbieten.	Kommandoschnittstelle
APP_6	Der SICCT-Kommandosatz kann auf dem CT-BCS Befehlssatz der MKT und der PC/SC Spezifikationen [MKT_10] basieren. Die Struktur der Kommandos muss derjenigen für ISO 7816-4 APDUs erfolgen.	Kommandostruktur
APP_7	SICCT-Kartenterminals können eine Schnittstelle zur zentralen Administration des Terminals unterhalten.	Administrationsschnittstelle
APP_8	Spezifische SICCT-Terminals für gleiche Anwendungsdomänen sollten eine "Drop in"-Ersetzbarkeit bei Produkten von unterschiedlichen Herstellern aufweisen.	Interoperabilität

ANMERKUNG: Eine konkrete Ausprägung eines auf SICCT-basierten Kartenterminals für eine oder mehrere Anwendungsdomänen kann nicht universell vorgegeben werden, da nicht immer dieselbe Komplexität oder Ausstattung benötigt wird. Das SICCT-Basiskonzept zeigt sich an dieser Stelle offen und benennt skalier- und kombinierbare Erweiterungen, die aber ebenso nur als standardisierte Optionen verfügbar sein sollen.

Tabelle 4: Optionale Vorgaben an ein SICCT

OPT_1	Lokale Kartenterminals an Peer-To-Peer-Bussystemen können ebenfalls Funktionen nach SICCT-Discovery anbieten, sofern die Gerätepräsenz durch eine Proxy-Instanz abgebildet wird.	UPnP - Proxy
OPT_2	Zur Authentifizierung kann das Kartenterminal eine Identität in Form eines gespeicherten SSL-Zertifikats in X509-Notation aufweisen.	Komponentenidentifikation
OPT_3	- SICCTs können über Ressourcen zur Authentifikation des Kartenanwenders verfügen. - Keypad - Biometrische Sensoren - Kontaktlos Leser	Anwenderauthentifikation
OPT_4	Autarke netzwerkfähige Kartenterminals können Funktionen zur automatischen Erkennung anbieten, wie in Kap.6 beschrieben	SICCT Discovery
OPT_5	Optional kann eine kryptografisch abgesicherte Kommunikation für entfernt im Netzwerk sichtbare Kartenterminals eingesetzt werden, sofern ein spezifischer Applikationskontext dieses erfordern sollte.	Kryptografische Sicherung des Kommunikationspfades
OPT_6	Unterstützung eines als Komfortsignatur benannten Mechanismus	Komfortsignatur

3.4 Sicherheitsanforderungen

Sicherheitstechnische Anforderungen an die SICCT-Basiskonzeption bestehen hinsichtlich

- des Zugriffs auf das Kartenterminal,
- der Verwaltung der Kartenterminalressourcen und
- in der Geheimhaltung von intern gehaltenen Daten.

Insbesondere Authentifikationsdaten, welche vom Kartenanwender über Interaktionspunkte am Kartenterminal eingegeben wurden, müssen der externen Welt verborgen bleiben und dürfen nur gerichtet an eine Chipkarte gesendet werden. Zu schützende Authentifikationsdaten muss das Kartenterminal aus einem korrespondierenden Kommandokontext erkennen, wie z.B. der Schutz der PIN-Daten nach und während einer PIN-Eingabe via Keypad.

Das Terminal darf Zugriffe nur über gerichtete Kommandos erlauben und muss jeglichen weiteren Zugriff auf Ressourcen unterbinden. Für netzwerkfähige Kartenterminals kann eine kryptografische Absicherung des Kommunikationsweges zur steuernden Instanz notwendig sein.

Die detaillierten Sicherheitsanforderungen werden im Kapitel 8 analysiert und beschrieben.

3.5 Benutzerrollen und CT Session

Ein Zugriff auf das Kartenterminal muss stets auf Basis einer Benutzerrolle im Rahmen einer Kartenterminalsitzung (CT Session) erfolgen. Das SICCT-Kartenterminalkonzept unterscheidet die in Tabelle 5 gezeigten Akteure entsprechend ihrer Zugriffsmöglichkeiten auf Funktionen und Betriebsparameter des Kartenterminals in drei Benutzerrollen.

Tabelle 5: Benutzerrollen eines SICCT

User role	Access	Function
Kartenterminaladministrator (CT Admin)	Privilegierter und uneingeschränkter Zugriff nach Authentifizierung durch das Kartenterminal	Administration, Konfiguration und programmatische Steuerung des Kartenterminalbetriebs: - Anzeige und Modifikation von Betriebsparametern, - Uneingeschränkte Nutzung bzw. Aufruf aller Kartenterminalkommandos.
Kartenterminalsteuerung (CT Control)	Unprivilegierter und beschränkter Zugriff ohne Authentifizierung durch das Kartenterminal (anonymous control)	Programmatische Steuerung des Kartenterminalbetriebs: - Keine direkte Modifikation von Betriebsparametern, - Abfrage bzw. Anzeige bestimmter Betriebsparameter, - Eingeschränkte Nutzung bestimmter Kartenterminalkommandos.
Karten- bzw. Kartenterminalanwender (CC / CT User)	Unprivilegierter und stark beschränkter Bedienerzugriff ohne Authentifizierung durch das Kartenterminal	Normale Anwendung bzw. Nutzung des Kartenterminals mit und ohne Chipkarte: - Bedienerrolle im Rahmen eines Benutzerdialogs sowie der Chipkartenapplikation (Benutzerinteraktionen), - Anzeige von Betriebsparametern im Rahmen des Benutzerdialogs (User Dialog) und der Möglichkeiten der Mensch-Maschine-Schnittstelle (User Interface) - Kein administrativer Zugriff, - Kein steuernder (programmatrischer) Zugriff.

Ein steuernder (programmatrischer) Zugriff auf das Kartenterminal erfolgt durch die Benutzerrolle

- CT Admin oder
- CT Control.

Tabelle 6: CT Session, Authorisation, Authentication und Session Handling

CT Session	Authorisation	Authentication	Session Handling
CT ADMIN Session	CT Admin Role	Explizite Logininformation - Benutzername (Username) - Default Username = "admin" - Benutzeridentifikation (Useridentifikation) durch Passwort.	Explizites Öffnen und Schließen einer CT Session.
CT CONTROL Session	CT Control Role	Implizite Logininformation - Benutzername (Username) - Default Username = "user" - Benutzeridentifikation (Useridentifikation) durch Passwort. - Ggf. anonymer Zugriff, wenn ein leeres Passwort übergeben wird, und die Terminaleinstellung dieses zulässt.	Explizites Öffnen und Schließen einer CT Session.
(none)	CC / CT User Role	(keine)	- Bedienerzugriff im Rahmen einer bestehenden CT Session. - CC / CT User kann keine CT Session aufbauen.

4 Beschreibung der Schnittstelle zur Karte

Für kontaktbehaftete Karten existieren zwei grundlegende Standards, die beide für die SICCT-Terminals relevant sind:

- ISO/IEC: [ISO/IEC 7810], [ISO/IEC 7816-2], [ISO/IEC 7816-3], [ISO/IEC 7816-10]
- EMV: [EMV_41]

Die Normreihe ISO/IEC 7816 und hier speziell [ISO/IEC 7816-3], enthält zumeist nur die Vorgaben für die Karten und ist deshalb nur indirekt als Basis für die Terminalanforderungen geeignet. Dagegen enthält die EMV-Norm neben den Kartenanforderungen (die denen der Normenreihe ISO/IEC 7816 stark ähneln) separat im Kapitel 5.5 die Terminalanforderungen.

Die SICCT-Spezifikation basiert auf den folgenden drei Regeln:

- 1) Terminalanforderungen der EMV-Norm werden vorzugsweise übernommen.
- 2) Besteht eine zusätzliche oder weitergehende Anforderung der ISO/IEC 7816 Normenreihe (direkt oder indirekt über die Anforderungen an die Karte), so ist diese maßgeblich.
- 3) Widersprechen sich die Anforderungen der beiden Normen, so muss das Terminal-Verhalten umgeschaltet werden. Die ISO-Betriebsart ist obligatorisch für ein SICCT, optional kann eine EMV-Betriebsart vorgesehen werden.

Der Anhang A als normativer Teil der SICCT-Spezifikation ist eine detaillierte Gegenüberstellung der Anforderungen aus den ISO/IEC- und EMV-Normen (letzte in der Variante "Neue Terminals ab Juli 2009").

Die Spalten "ISO 7816-3" und "EMV 4.1" der Tabelle in Anhang A definieren die Anforderungen der Normen an das Terminal (direkt oder indirekt) zu dem in der Spalte "Description" gelisteten Sachverhalt. Die letzte Spalte gibt Auskunft über die für ein SICCT-Terminal maßgebliche Anforderung gemäß den drei zuvor gelisteten Regeln. Nur, wenn dieses Tabellenelement geteilt ist, muss sich das Terminal unterschiedlich für den eingestellten Modus "ISO-Mode" oder "EMV-Mode" verhalten.

Aus diesen Regeln ergibt sich die uneingeschränkte Einhaltung der zutreffenden Norm, jeweils für die Terminal-Einstellung "ISO" und "EMV", sowie eine Minimierung der einstellungsabhängigen Unterschiede.

Für SICCT-Terminals, die auch die EMV-Betriebsart unterstützen, muss der Auslieferungszustand die Einstellung "ISO" sein.

Als kontaktlose Karten (RFID Token) dürfen nur "Proximity Cards" (PICC) berücksichtigt werden, die sich nach der Normenreihe ISO/IEC 14443 richten, konkret: [ISO/IEC 14443-1], [ISO/IEC 14443-2], [ISO/IEC 14443-3] und [ISO/IEC 14443-4].

4.1 Mechanische Anforderungen

4.1.1 Geeignete kontaktbehaftete Kartentypen

Das Terminal ist zur Aufnahme von einer oder von mehreren kontaktbehafteten Karten der Größe ID-1 (siehe [ISO/IEC 7810]) vorgesehen.

Das Terminal kann optional eine oder mehrere kontaktbehaftete Karten der Größe ID-000 aufnehmen (Plug-In).

Für alle diese Karten müssen die Normen [ISO/IEC 7816-10] und [ISO/IEC 7816-2] oder ID-000 gelten.

4.1.2 Anforderungen an die Kartenkontaktierungen für Größe ID-1

Alle Anforderungen aus Kapitel 5.4 der EMV-Norm [EMV_41] müssen eingehalten werden. Die Kartenkontakte C4, C6 und C8 können vorhanden sein.

Für die Karten ID-1 wird eine hohe Zuverlässigkeit und Funktionssicherheit auch bei hoher Steckzyklenzahl für Kontaktierung und Karte verlangt. Hierfür muss sichergestellt sein:

- Mindestzyklenzahl von 200.000 Zyklen
- absenkende Kontakte mit einer kurzen translatorischen Bewegung auf den Kartenkontaktflächen (ca. 0,5 mm)

Ferner muss ein Endkontaktschalter (Kontakt bei vollständig eingeschobener Karte) vorgesehen werden.

Je nach Anforderungen der Applikation kann für bestimmte oder alle Kartenkontaktierungen die Bauart vorgeschrieben werden:

- Einsteckleser ohne Entnahmeschutz (Karte immer im vollen Anwenderzugriff)
- Auswurfleser mit eingeschränktem Entnahmeschutz (Karte im eingesteckten Zustand nur 1-3mm aus Terminal herausragend, dadurch Notentnahme jederzeit möglich)
- Auswurfleser mit sicherem Entnahmeschutz durch Verriegelung (Karte ist vollständig dem Anwenderzugriff entzogen und wird ausschließlich von der Applikation freigegeben).

Bei der Bauart Auswurfleser können weitere Varianten unterschieden werden:

- mit mechanischem Auswurf, z.B. Auswurfaste
- mit elektromechanischem Auswurf auf Anforderung der Applikation
- mit manuellem Einstecken der Karte
- mit motorischem Einzug der Karte
- mit motorischem Einzug und der Möglichkeit, die Karte nicht auszuwerfen, sondern einzubehalten.

Bei Kartenlesern mit Entnahmeschutz muss die Möglichkeit einer Notentnahme der Karte bei Fehlfunktionen des Terminals oder bei Stromausfall sichergestellt werden.

4.1.3 Anforderungen an die Kartenkontaktierungen für Größe ID-000

Applikationen können eine Mindestanzahl von Plug-In-Kartenaufnahmen verlangen.

Die Abmessungen der Kartenaufnahme müssen der Norm CEN ENV 1375-1 [STD16] (kartenseitige Bemaßung) entsprechen. Alle Anforderungen aus Kapitel 5.4 der EMV-Norm [EMV_41] müssen eingehalten werden. Die Kartenkontakte C4, C6 und C8 können optional vorhanden sein.

Der Zugang der Plug-In-Karten kann applikationsgemäß wie folgt gefordert sein:

- Karte im ungesicherten Kartenfach frei zugänglich,
- Karte im versiegelten Kartenfach bedingt zugänglich (Zugangsberechtigung muss organisatorisch geregelt sein),
- Karte nicht von außen zugänglich,
- bei mehreren Kartenaufnahmen eine Mischung der Varianten.

Optional kann die Applikation die Existenzprüfung dieser Karten fordern, das bedingt einen entsprechenden zusätzlichen Kartenkontakt. Nur dadurch wäre bei Entnahme einer aktivierten Karte eine normgemäße Deaktivierung realisierbar sowie eine Interrupt-gesteuerte Meldung beim Einsetzen bzw. Entnahme der Plug-In-Karte. Dies betrifft in Folge auch die Statusauskunft und das Eventing zum Host.

4.1.4 Anforderungen bei Verwendung von kontaktlosen Karten

Optional kann das Terminal Funktionalitäten zur Verwendung kontaktloser Karten (PICC, Proximity Cards nach der Normenreihe Iso/IEC 14443) aufweisen. Hierfür ist am Terminal die Position der Antenne gekennzeichnet. Bei Annäherung der Karte ist sicherzustellen, dass die Karte zuverlässig erkannt und bearbeitet wird.

4.2 Elektrische Anforderungen

4.2.1 Elektrische Anforderungen für kontaktbehaftete Karten

Die Terminal-Anforderungen ergeben sich aus den Normen

- [ISO/IEC 7816-3] und
- EMV Version 4.1 [EMV_41]

ANMERKUNG: Die Vorgaben für das Verhalten eines SICCT-Terminals sind in diesem Kapitel und in der Tabelle Anhang A detailliert aufgeführt.

Ergänzungen und Erläuterungen zu den Vorgaben der Tabelle in Anhang A:

Anforderungen an Kontakt C6: Dieser Kontakt wurde früher für die Programmierspannung VPP (nur bei Class A) benutzt. Diese wird bei modernen Karten nicht mehr benötigt. Deshalb fordern ISO- wie EMV-Norm, dass C6 nicht angeschlossen werden sollte. Dies gilt auch für ein SICCT-Terminal.

Anforderungen an Kontakte C4 und C8: Diese Kontakte müssen laut EMV-Norm nicht vorhanden sein. Dies gilt auch für ein SICCT-Terminal. Kann das Terminal optional USB-Karten nach [ISO/IEC 7816-12] und/oder synchrone Karten vom Typ 2 nach [ISO/IEC 7816-10] unterstützen, muss C4 und evtl. C8 entsprechend ansteuerbar sein. In diesem Fall müssen C4 und C8 im deaktivierten Zustand (Low-Zustand) gehalten werden, bis eine entsprechende Karte erkannt ist.

Wie aus der Tabelle (Teil A1) ersichtlich, soll das Terminal folgende Spannungsversorgungen für die Karten bereitstellen:

- 5,0 Volt (Class A), garantierter Strom 60 mA
- 3,0 Volt (Class B), garantierter Strom 55 mA
- optional: 1,8 Volt (Class C), garantierter Strom 35 mA

Das Terminal garantiert diese Versorgungsspannungen/ströme für jede der vorhandenen Kontaktiereinheiten, auch gleichzeitig und unabhängig von den jeweils anderen Einheiten.

Generell müssen alle vorhandenen Karten unabhängig voneinander ansteuerbar sein. Das bezieht sich auf alle vom Terminal unterstützten Kartenkontakte.

4.2.2 Reset-Verhalten und ATR-Bearbeitung

Die Vorgaben für das Verhalten eines SICCT-Terminals bezüglich der Normenreihe ISO/IEC 7816 ([ISO/IEC 7816-2], [ISO/IEC 7816-3] und [ISO/IEC 7816-10]) und EMV [EMV_41] sind in Abschnitt 4 und in der Tabelle Anhang A detailliert aufgeführt.

Ergänzungen und Erläuterungen zu den Vorgaben der Tabelle in Anhang A:

Im Rahmen der ATR-Prozedur wird ergänzend zu [EMV_41] die Versorgungsspannung für die folgende Chipkarten-Session vereinbart. Dazu muss das in [ISO/IEC 7816-3] Kapitel 6.2.4 definierte Verfahren angewendet werden. In Ergänzung hierzu muss folgende Reihenfolge der Aktivierungen eingehalten werden: zunächst 5V, dann 3V, optional schließlich 1,8V.

Als abschließenden Teil der Einschaltprozedur soll das Terminal ergänzend zu [EMV_41] das PPS-Verfahren (Protocol Parameter Selection) durchführen. Das Terminal muss folgende Mindestanforderungen erfüllen:

- Taktfrequenz f_{CLK} 1,0 MHz bis 5,0 MHz
- Parameter F_n 372 und 512
- Parameter D_n 1, 2 und 4

Größere D_n -Werte sind dringend empfohlen und können von der Anwendung gefordert werden. Bezüglich der Parameter F_n und D_n siehe [ISO/IEC 7816-3] Tabelle 7.

Terminals, die einzelne Baudraten-Faktoren (z.B. $D_n=64$) nicht unterstützen, können mittels PPS-Verfahren einen kleineren, vom Terminal unterstützten Faktor (z.B. $D_n=32$) aushandeln, falls von der Karte der Negotiable Mode angezeigt wird. Alternativ dazu ist die Reduzierung der Taktfrequenz zulässig.

Gebrochen rationale Quotienten F_n/D_n sind nicht zwingend zu unterstützen.

Das PPS-Verfahren ist erfolgreich, wenn die Bedingungen von [ISO/IEC 7816-3] 9.3 erfüllt sind, zusätzlich die PPSS0_Response = 'FF' ist und die PCK_Response vorhanden und korrekt ist. Im Fehlerfall muss die Karte deaktiviert werden.

Höhere Werte für f_{CLK} und F_n sind für moderne Karten nicht mehr notwendig, dürfen aber optional unterstützt werden.

Die Taktfrequenz soll gemäß Kapitel 5.5.4 von [EMV_41] während der ATR-Sequenz und der nachfolgenden Card Session eine Stabilität von +/- 1% haben. Falls eine Frequenzumschaltung nach dem ATR erfolgt, gilt diese Forderung nicht für den Umschaltzeitpunkt.

Der Clock-Stop-Mode aus [ISO/IEC 7816-3] 6.3.2 kann ergänzend zu [EMV_41] unterstützt werden.

Die mit dem ATR-Parameter TC1 definierte ExtraGuardTime (EGT) bezieht sich nur auf die vom Terminal gesendeten Zeichen. Die EGT ist ebenfalls nach dem letzten Zeichen der Karte bis zum ersten Zeichen eines neuen Terminal-Befehls einzuhalten. Die EGT wird immer aufgrund des aktuellen F/D-Verhältnisses berechnet, unabhängig davon, ob T=15 im ATR vorhanden ist.

Das Global Interface Byte TB_i (first TB for T=15) darf nicht zur Ablehnung des ATR führen. Es ist aber solange inhaltlich zu ignorieren, wie Kontakt C6 nicht mit einer SPU-Funktion belegt ist, siehe [ISO/IEC 7816-3] 5.2.4 und 8.3.

4.2.3 Unterstützung von synchronen Karten

Zusätzlich zu asynchronen Prozessorkarten können auch Speicherkarten mit synchroner Datenübertragung unterstützt werden, siehe [ISO/IEC 7816-10]. Die Verfahren und Protokolle sind der MKT-Spezifikation Teil 2, Abschnitt 7 zu entnehmen [MKT_10].

Die Kartenaktivierung ist bei Unterstützung synchroner Karten wie folgt zu erweitern: Sendet die Karte beim Kalt-Reset keine asynchronen Antwortzeichen, soll die Karte deaktiviert und das ATR-Verfahren nach [ISO/IEC 7816-10] zur Erkennung einer synchronen Karte angewendet werden.

Wird auch nach diesem Verfahren kein ATR von der Karte geliefert, kann das Protokoll S=8 (I²C mit 1-Byte-Adressierung) verwendet werden. Erst bei negativem Ergebnis aller ATR-Versuche darf die Karte abgewiesen werden.

Die optionale Unterstützung synchroner Karten darf sich auf Karten der Größe ID-1 beschränken, da Karten der Größe ID-000 in der Regel als Sicherheitsmodule verwendet werden, deren Anforderungen nicht von synchronen Karten erfüllt werden können.

4.2.4 Elektrische Anforderungen für kontaktlose Karten

Werden kontaktlose Karten (RFID token) unterstützt, so müssen sich die Terminal-Anforderungen nach der Normenreihe ISO/IEC 14443 für Typ A und Typ B (PICC, Proximity Cards) richten, siehe [ISO/IEC 14443-1], [ISO/IEC 14443-2], [ISO/IEC 14443-3] und [ISO/IEC 14443-4]. Darüberhinausgehend gelten die Anforderungen aus Kapitel 4.2.1 aus [PCSC2].

4.3 Protokolle

4.3.1 Protokolle für kontaktbehaftete Karten

Vorgeschriebene Protokolle für asynchrone Karten sind T=0 und T=1 gemäß [ISO/IEC 7816-3] oder [EMV_41]. Weitere Protokolle können optional unterstützt werden.

Die Vorgaben für das Verhalten eines SICCT-Terminals bezüglich der Standards [ISO/IEC 7816-3] und EMV [EMV_41] sind in Abschnitt 4.0 und in der Tabelle Anhang A (Teile A5 und A6) detailliert aufgeführt.

Für das in [ISO/IEC 7816-3] 12.2 für das Protokoll T=0 beschriebene APDU-Mapping werden unter Umständen der Befehl und/oder die Antwort in mehreren Teilen von/zur Karte übertragen. Dies wird über spezielle Returncodes (z.B. '61XX') von der Karte gesteuert. Diese Abläufe müssen komplett vom Terminal umgesetzt werden (zum Beispiel Generierung eines GET RESPONSE) und verlaufen für die Anwendung transparent, das heißt die speziellen Returncodes dürfen nicht der Applikation übergeben werden, sondern nur die komplette Kartenantwort. Dies gilt sowohl für den ISO-Mode als auch für den optionalen EMV-Mode.

Das in [ISO/IEC 7816-3] 12.2 für das Protokoll T=0 beschriebene APDU-Mapping im Falle Extended Length (L_e und L_c größer als ein Byte) darf vom SICCT-Terminal nicht unterstützt werden. Im Fehlerfall muss das Terminal den Fehlercode '6F00' erzeugen.

Die Protokolle der synchronen Karten sind meist in Industrie-Publikationen veröffentlicht:

- S=8 I²C bzw. SDAP, siehe [IND1]
- S=9 3-Draht bzw. 3WBP, siehe [IND2]
- S=10 2-Draht bzw. 2WBP, siehe Typ 1 in [ISO/IEC 7816-10], Teil 6 von [MKT_10] und [IND3]
- S=11 FCBP, siehe Typ 2 in [ISO/IEC 7816-10]

4.3.2 Protokolle für kontaktlose Karten

Werden kontaktlose Karten unterstützt, gilt das Protokoll T=CL gemäß [ISO/IEC 14443-4]. Weitere Protokolle sind optional.

Darüberhinausgehend gelten die Anforderungen aus Kapitel 3.1.3 aus [PCSC3].

5 Command Set

Dieser Teil beschreibt die verfügbaren Kartenterminalkommandos zur Steuerung von SICCT-Kartenterminals unterschiedlicher Ausprägung und Bauart.

Das Funktionsprinzip des SICCT-Terminals basiert auf einem Befehlssatz (SICCT-Command Set), welcher zur Steuerung des Kartenterminals sowie zur Kontrolle, Aktivierung, Deaktivierung und Statusabfrage des elektrischen Zustands von Chipkartenkontaktiereinheiten einen definierten Satz an Kartenterminalkommandos bereitstellt.

Eine steuernde Entität tauscht mit dem SICCT-Kartenterminal Nachrichten nach dem Request / Response-Verfahren aus. Die Nachrichtentelegramme beinhalten Kommandosequenzen, deren Interpretation und Ausführung durch einen Kommandointerpreter des SICCT-Terminals erfolgen. Die CT-Kommandos sind anwendungsneutral und können von beliebigen Chipkarten-basierten Anwendungen verwendet werden.

Der Transport der Nachrichten darf prinzipiell über diverse Bussysteme (z.B. RS-232, USB, Ethernet 802.3, ...) geschehen und muss transparent und unabhängig zu den Kommandostrukturen und Inhalten gestaltet sein. Das Kapitel 'Schnittstellenbeschreibung zum Host' beschreibt die Abläufe sowie das zugrunde gelegte netzwerkbasierte Protokoll.

Das Verfahren des Austauschs von Kommandonachrichten zwischen einer steuernden Entität (HOST) und einem Kartenterminal (CT) muss dem Sessionprinzip folgen, wobei immer eine steuernde Entität einer Terminalinstanz zugeordnet wird. Während einer Session muss ein SICCT-Terminal exakt einer steuernden Instanz zugeordnet sein.

Der Umfang des SICCT-Kommandosatzes (SICCT-CS) definiert das Verhalten des Kommandointerpreters. Jede erfolgreiche Abarbeitung eines Kommandos soll einen definierten Gerätezustand oder Betriebsmodus erzeugen. Betriebsmodi und Zustände sollen durch den Einsatz des Kommandosatzes selektierbar und abfragbar sein. Eine ansteuernde Entität kann somit Gerätezustände erzeugen oder setzen bzw. diese eindeutig abfragen und unterscheiden. Es wird hierbei angenommen und vorausgesetzt, dass die steuernde Entität aus diesen Informationen einen separaten anwendungsfallbezogenen Kontext generiert, den das Kartenterminal nicht kennen muss.

5.1 SICCT Command Structure

Die Struktur der Kartenterminalkommandos muss der bytesequenziellen Kommandostruktur für Chipkarten als sog. APDU (Application Protocol Data Unit, Byte-Folge) nach [ISO/IEC 7816-4] folgen, bzw. derjenigen des MKT-Basic Command Sets (CT-BCS) [MKT_10], dessen Befehlsstruktur demselben Prinzip unterliegt.

SICCT-Terminal-APDUs werden generell in zwei Gruppen unterschieden

- Command-APDU (C-APDU),
- Response-APDU (R-APDU).

In Ergänzung zu diesen sieht das Transportprotokoll noch eine separate Ereignisnachricht vor (siehe 6.2.4.4 Ereignisbenachrichtigung).

Zu jedem Kommando C-APDU, welches eine Entität an das Kartenterminal sendet, muss der Sender eine Antwort (R-APDU) als Quittung erhalten. Response-APDUs dürfen nicht gesondert quittiert werden.

Kommando C-APDUs stellen eine Bytesequenz dar. Diese besteht aus den zwei Teilen Header und Body.

<C-APDU>									
Length: 4 ... ((0...3) + Lc + (0...3))									
[Header] mandatory				[Body] [optional]					
Length: 4				Length: 0 ... ((0 ... 3) + Nc + (0 ... 3))					
1	2	3	4	Coding Rules for Nc and Ne in accordance to ISO 7816-4					
				Max. Value indicated by Lc: Nc <= 65535		Max. Value indicated by Le Ne <= 65536			
				Lc absent	empty	Nc = 0	Le absent	empty	Ne = 0
				Lc short ¹	1 Byte	'01' <= Lc <= 'FF' 1 <= Nc <= 255	Le short	1 Byte	'01' <= Le <= 'FF' 1 <= Ne <= 255 Le = '00' Ne = 256
				Lc extended ¹	3 Byte	'000001 <= Lc <= '00FFFF' 1 <= Nc <= 65535	Le extended	3 Byte	Condition: Lc absent '000001' <= Le <= '00FFFF' 1 <= Ne <= 65535 Le = '000000' Ne = 65536
								2 Byte	Condition: Lc extended '0001' <= Le <= 'FFFF' 1 <= Ne <= 65535 Le = '0000' Ne = 65536
CLA	INS	P1	P2	[Lc]	[Data Field] <Nc Bytes of Data>	[Le] < requesting Ne Bytes of response Data >			

Aufbau und Länge eines Command Application Protocol Data Units (C-APDU)

Nur der Header, d.h. die ersten vier Bytes (CLA, INS, P1, P2) der C-APDU sind verpflichtend. Die nachfolgenden Felder des Bodys können bzw. müssen konditional vorhanden sein und dürfen ein Datenfeld und / oder einen Längenqualifizierer für gesendete wie erwartete Antwortdaten beinhalten. Ihre Präsenz ist abhängig vom Anwendungsfall, Kommando und Anwendungskontext.

5.2 Command Header

5.2.1 CLASS - Byte

Das CLASS-Byte zeigt einen Kommandosatz durch einen eindeutigen, individuellen Wert an.

¹ According to [ISO/IEC 7816-4] either both Le and Lc are short or extended.

5.2.2 INSTRUCTION - Byte

Das Instruction Byte muss eine auszuführende Funktion des Kartenterminals durch einen eindeutigen Wert in einem Wertebereich eines Kommandosatzes kodieren.

5.2.3 Parameter P1

Der Parameter P1 muss einen Ausführungskontext für einen Befehl am Kartenterminal beinhalten. Dieser gibt an, welche Kartenterminalressourcen (Functional Units) bei der Befehlsbearbeitung hinzugezogen werden.

Prinzipiell richtet sich ein Kommando an das Kartenterminal oder eine Functional Unit des Kartenterminals. Klassisch kodiert P1 eine einzelne Functional Unit in einem Byte (Direct Coding). Zur Erweiterung des Wertebereichs kann P1 durch einen eindeutigen Wert ("Escape-Zeichen") eine Referenz auf ein TLV-Datenobjekt im Datenteil des Kommandos (Command Data) signalisieren (Referenced Coding). Das TLV-Datenobjekt darf eine oder mehrere Functional Units referenzieren.

5.2.4 Parameter P2

Der Parameter P2 stellt generell einen Command Qualifier dar, der ein Kommando oder einen Ausführungskontext noch näher qualifizieren lässt. Aufgrund der unterschiedlichen Kommandofunktionen (INS) erfolgt keine stereotype Ausprägung von P2, so dass die Bedeutung des Parameters P2 vom jeweiligen Befehlskontext abhängt. Zur Erweiterung des Wertebereichs kann P2, und sofern das Kommando dieses unterstützt, durch einen eindeutigen Wert ("Escape-Zeichen") auf eine Referenz eines TLV-Datenobjekts im Datenteil des entsprechenden Kommandos (Command Data) hinweisen (Referenced Coding).

5.3 Command Body

5.3.1 Data Length Lc und Expected Response Length Le

Die Längenkodierer Lc und Le müssen den Kodierungsregeln nach ISO 7816-4 [STD8] folgen und sollten das Extended Length Format unterstützen.

Üblicherweise besitzen Lc und Le eine Länge eines Bytes. Dabei kann Lc maximal 255 und Le max. 256 Byte kodieren.

Optional ist ebenso möglich, ein Lc und Le-Feld von je maximal drei Byte zu führen. Der Maximalwert für Lc kann 65535 und der von Le 65536 kodieren. Der Wert für Lc (Length of command data field) kodiert die Anzahl der Daten, welche mit dem Kommando (C-APDU) an den Empfänger gesendet werden.

Die Angabe für Le (Length of expected data) kodiert, wie viele Datenbytes in der Antwort (R-APDU) maximal zurückgegeben werden dürfen. Der Aufrufer reserviert für den Empfang der Daten einen korrespondierend großen Empfangspuffer und kann somit byteweise ggf. Teilinformationen von Datenobjekten anfordern.

- Sonderfall 0 – Le fehlt, und die Anzahl der Daten wird zu Null (Ne = 0) bestimmt, d.h ein SICCT-Kommando liefert keine Daten und nur den Status (SW1SW2) zurück.
- Sonderfall 1 - Die Anzahl zurückgegebener Daten unterschreitet oder ist gleich dem (erwarteten) Wert von Ne. Das bedeutet, dass alle verfügbaren Daten, zurückgeliefert wurden. Der korrespondierende Statuscode (SW1SW2, siehe 5.4.1) für SICCT-Kommandos muss dann je nach Kartentyp '9000' bzw. '9001' oder darf '6282' sein.
- Sonderfall 2 - Wird der Wert für Le auf '00' bzw. '000000' gesetzt, bedeutet dieses alle verfügbaren Daten, maximal 256 bzw. 65536 Bytes, zurückzuliefern. Der korrespondierende Statuscode (SW1SW2, siehe 5.4.1) für SICCT-Kommandos muss dann je nach Kartentyp '9000' bzw. '9001' oder darf '6282' sein.
- Sonderfall 3 - Existieren Le und Lc müssen diese gleich kodiert entweder in short oder extended Form vorliegen.
- Sonderfall 4 - Le darf in nur Zwei-Byte <HB | LB> kodiert werden, wenn Lc in extended (Drei-Byte-Darstellung) vorliegt.

Für die Darstellung von Lc sowie Le muss es in allen SICCT-Kommandos ebenso zulässig sein, die erweiterte Längendarstellung nach den Regeln aus [ISO/IEC 7816-4] (siehe auch 5.1 SICCT Command Structure) für Lc und Le durchgängig zu verwenden, auch wenn diese Form in den Beschreibungen zu den SICCT-Kommandos nicht explizit erscheint.

5.3.2 Command Data Field

Das Command Data Field kann beliebig strukturierte Befehlsdaten beinhalten. Der Inhalt und die Datenstruktur sind befehlspezifisch. Die maximale Länge dieses Felds ergibt sich aus dem jeweiligen Maximalwert für Lc.

5.4 SICCT Response Structure

Eine Antwort zu einem C-APDU Kartenterminalkommando muss von einem SICCT generell in Form eines R-APDUs zurückgegeben werden. Optional und abhängig vom Anwendungsfall, Kommando und Anwendungskontext kann ein Datenfeld vorhanden sein. Verpflichtend ist die Präsenz eines zwei-Byte-Statusworts 'SW1SW2', dessen Statusbytes SW1 und SW2 getrennt bewertet werden müssen, um den Zustand des vorausgegangenen Kommandos oder den Gerätezustand zu erkennen.

<R-APDU>		
[Body] [optional]	Trailer mandatory	
Requested Information	Status Word	
Length: 0 ... [<Le> of C-APDU]	Length: 2 Byte	
Position: 1 ... (1 + [<Le> of C-APDU])	Pos: 1 ... (1 + [<Le> of C-APDU])	Pos: 2 ... (2 + [<Le> of C-APDU])
[Information Field] < up to Ne Bytes of Response Data >	SW1	SW2

Aufbau und Länge eines Response Application Protocol Data Units (R-APDU)

5.4.1 Response Trailer

Ein R-APDU mit einem Trailer muss stets auf ein C-APDU zurückgesendet werden. Der Trailer muss ein Zwei-Byte-Statuswort (SW1SW2) enthalten, dessen Kodierung in Anlehnung an ISO 7816-4 [STD8] erfolgen soll. Das erste Byte SW1 muss stets den Ausgang einer Operation kodieren.

SW1		Meaning		
'61'	'90'	Process completed	Normal Processing	normale Bearbeitung
'62'	'63'		Warning	Warnung
'64'	'65'	Process aborted	Execution Error	Ausführungsfehler
'67' ... '6F'			Checking Error	Prüfungsfehler
Note: Ranking - 1 st 'Checking Error', 2 nd 'Warning' and 'Normal Processing' , '3 rd Execution Error'				

Das Byte SW2 soll den Ausgang einer Operation genauer spezifizieren. Der Wert kann befehlspezifisch ausfallen.

Folgende Statusworte gelten allgemein, d.h. sofern diese nicht im Befehlskontext spezifischer dargestellt sind:

SW1 SW2		Meaning
General Execution Errors		
'64xx'	Execution error	Ausführungsfehler
'64A1'	No card present	Keine Karte vorhanden
'64A2	Card not activated	Karte nicht aktiviert
General Checking Errors		
'6700'	Wrong length	Falsche Länge
	Too less / many Data (Objects) given within command.	Zu viele oder zuwenig Daten (Objekte) im Kommando enthalten.
'6900'	Command not allowed	Kommando (ggf. z.Zt.) nicht zulässig
'6941'	Functional Unit busy / not available	Funktionseinheit belegt / nicht verfügbar
'6A00'	Wrong parameters P1, P2	Falsche Parameter P1, P2
'6A80'	Invalid Data Object	Unzulässiges Datenobjekt
	Incorrect parameters in the command data field	Unzulässige Parameter (Datenobjekt) im Datenfeld
'6A88'	Missing Data Object	Fehlendes Datenobjekt
	Referenced data or reference data (data object) not found (exact meaning depending on the command)	Referenzdaten (Datenobjekt) nicht in Datenteil des Kommandos enthalten.
'6C00'	Wrong length Le	Falsche Längenangabe Le
'6D00'	Wrong instruction	Falsches / unbekanntes Instruction-Byte
'6E00'	Class not supported	Falsches / unbekanntes Class-Byte
'6F00'	Communication with ICC not possible	Kommunikationsfehler zur Karte

5.4.2 Checking Error / Prüfungsfehler

Bevor ein Kommando abgearbeitet wird, muss das SICCT-Terminal eine komplette Überprüfung aller Parameter durchführen. Bei Terminals, die eine Queue verwenden, muss immer nur das aktuelle Kommando einer Parameterprüfung unterzogen werden.

Falls in einem Kommando mehrere fehlerhafte Parameter enthalten sind, muss derjenige Fehlercode zurückgegeben werden, der eine höhere Priorität aufweist.

Nachfolgende Tabelle beschreibt alle möglichen Fehlercodes für die Parameterüberprüfung mit fallender Priorität.

Priority	SW1 SW2	Meaning	Remark
Priorität		Bedeutung	Bemerkung
1	'6E00'	Class not supported	Falsches / unbekanntes Class-Byte
2	'6D00'	Wrong Instruction	Falsches / unbekanntes Instruction-Byte
3	'6700'	Wrong Lc	Lc Wert ist für dieses SICCT Kommando nicht erlaubt
4	'6C00'	Wrong length Le	Le Wert ist für dieses SICCT Kommando nicht erlaubt
5	'6A00'	Wrong Parameters P1/P2	Dieser Fehler muss auch bei Referenced Coding (P1 und/oder P2) zurückgegeben werden, falls im FUI DO (s. 5.5.10.9) oder FU CON DO (s. 5.5.10.10) eine nicht verfügbare FU angegeben ist.
6	'6A80'	Invalid Data Object	Das Datenobjekt ist bei dem SICCT Kommando nicht erlaubt. Das Datenobjekt hat eine laut Spezifikation unerlaubte Länge
7	'6A88'	Missing Data Object	Ein für das SICCT Kommando notwendiges Datenobjekt fehlt.
8	'6A89'	Too many Data Objects	Zu viele Datenobjekte gleichen Typs sind enthalten.
9	'6930'	Terminal does not support Timer Option	Das Terminal unterstützt (für diesen Befehl) keine Timer-Option.
10	'6940'	Command with Display not supported	Das Terminal unterstützt (für diesen Befehl) keine Display-Option.
11	'6942'	The addressed Display does not support the selected Character Set.	Das Display unterstützt den selektierten Zeichensatz nicht.

Tabelle 7: Fehlercodes nach Prioritäten

5.4.3 Response Body

Der Body muss aus einem Datenfeld bestehen, dessen maximale Länge durch den Wert von Le im C-APDU bestimmt ist. Entgegen der Angabe von Ne kann das Datenfeld fehlen oder kürzer ausfallen, sofern bei der Befehlsausführung weniger verfügbare Daten zurückgemeldet werden oder ein Fehlerzustand aufgetreten ist. Der Status muss dann im Response-Trailer entsprechend kodiert sein.

5.5 Command Set Overview and General Return Codes

Ein SICCT-Kartenterminal muss über einen zentralen Kommandointerpreter verfügen und kann über einen Kommandosatz gesteuert werden. Die Funktionalität eines SICCT-Kartenterminals richtet sich nach der technischen Ausprägung (Ausstattung) des Geräts sowie dem Umfang und der Mächtigkeit des Kommandosatzes. Jeweils ein Kommandosatz muss einem Betriebsmodus zugeordnet sein. Über die Selektion eines Betriebsmodus kann der Kommandosatz eingestellt bzw. gewechselt werden. Es muss ein Betriebsmodus zur Zeit aktiviert sein.

5.5.1 Adressierung und Transport von Kommandos

Am Transportendpunkt des Kartenterminals muss sich der Kommandointerpreter befinden, der genereller Empfänger aller Kartenterminalkommandos sein muss.

Kartenterminal- wie Chipkartenkommandos werden auf APDU-Level mit einer vom Absender vorgegebenen eindeutigen Sequenznummer zur Ausführung an den Kommandointerpreter übergeben. Das Kartenterminal muss nach Ausführung des Kommandos unter Angabe der Sequenznummer antworten (Response).

Das Kapitel 'Schnittstellenbeschreibung zum Host' beschreibt das Protokoll sowie die Datenstruktur zum Transport von Kommandos an den Interpreter.

Die vorgesehene Datenstruktur dient ebenso zum Rücktransport von Response-Daten und Events vom Kartenterminal und beinhaltet zur Unterscheidung folgende wesentliche Informationen

- eine Nachrichtenkennzeichnung
(Messagetype: Command, Response oder Event),
- eine eindeutige Sequenznummer,
- eine Empfängerkennzeichnung für Kommandos,
- eine Absenderkennzeichnung für Response-Daten.
- eine Längenangabe des Messagetype abhängigen Datenteils.

5.5.2 Ausführung und Wirkweise von Kommandos

Der SICCT Kommandosatz ermöglicht primär die Aktivierung, Kommunikation und Deaktivierung mit Chipkarten. Zusätzlich müssen Kommandos zum Management des Kartenterminals sowie zur Realisierung einer Benutzerführung am Kartenterminal vorhanden sein.

Jedes Kommando adressiert Ressourcen des Kartenterminals und realisiert primär eine Hauptaktion (wie z.B. Aktivierung, Deaktivierung, Display-Ausgabe, Keypad-Eingabe). Über optionale Kommandoqualifizierer dürfen einige Kommandos die Kombination mit einer oder mehreren Zusatzaktionen wie Statusanzeige über die Mensch-Maschine-Schnittstelle, Benutzerdialoge oder den mechanischen Auswurf einer Chipkarte zulassen.

Vor dem Hintergrund gesehen, kann die Ausführung der hauptsächlichen Kommandos in einzelnen Phasen ablaufen:

Ausführungsphasen der SICCT Kommandos				
Phase		Stage	Option	Description
preprocessing phase	Vorbereitungsphase	1	✓	z.B. - Anzeige eines Benutzerdialogs, - Eingabeaufforderung - Chipkartenanforderung
				i.e. - Displaying of user dialogue - Prompt for input of user data - Request for chipcard insertion
processing phase	Ausführungsphase	2		z.B. Ausführen der Hauptaktion, z.B. Aktivieren einer Chipkarte nach ISO 7816
				i.e. Execution of main action e.g. chipcard activation accordingly ISO 7816
postprocessing phase	Nachbereitungsphase	3	✓	z.B. - Benutzerdialoganzeige - Chipkartenentnahme - Chipkartenauswurf
				i.e. - Displaying of user dialogue - Request for chipcard removal - Chipcard ejection

Die Mächtigkeit und Wirkweise einzelner Kommandos muss sich nach der Art des Kommandos, sowie den technischen Möglichkeiten und verfügbaren Ressourcen der jeweiligen Terminalplattform (LED, Display, Keypad, interner Timer, Kontaktloser Leser etc.) richten.

5.5.3 Abbruch der Kommandoausführung

Die Ausführung eines Kommandos kann

- zur Ausführungszeit am Terminal durch Benutzerinteraktionen über die Mensch-Maschine-Schnittstelle am Kartenterminal
- oder durch die steuernde Entität (HOST) mittels eines SICCT CONTROL COMMANDs unter Angabe der Kommandosequenznummer abgebrochen werden, sofern ein bestimmter Kommandotyp den Kommandoabbruch unterstützt.

Der Ausgang bzw. das Resultat eines Kommandoabbruchs ist kommandospezifisch und muss sich nach der Art Verarbeitungsphase, in der sich das Kommando zum Zeitpunkt des Abbruchs befindet, richten.

5.5.4 Nebenläufige Kommandobearbeitung

Empfänger von Kommandos muss stets der zentrale Kommandointerpreter des Kartenterminals sein, der auch alle Ressourcen verwaltet. Zur optionalen Unterstützung einer möglichen nebenläufigen Kommandobearbeitung durch das Kartenterminal werden folgende Ablaufregeln definiert

- 1) Die generelle Unterstützung nebenläufiger Kommandobearbeitung ist abhängig vom Kommunikationsprotokoll, dem Betriebsmode oder dem Betriebszustand des Kartenterminals und kann generell negiert werden.
- 2) Für den Fall, dass keine nebenläufige Kommandobearbeitung möglich ist, verarbeitet das Kartenterminal ein Kommando zurzeit,
- 3) Empfänger aller Kommandos muss stets der Kommandointerpreter sein. Der Kommandointerpreter kann mehrere Kommandos nebenläufig zur Ausführung bringen, sofern die Kommandos nicht gleichzeitig auf dieselben Kartenterminalressourcen (Functional Units) zugreifen,
- 4) Die Eignung zur nebenläufigen Abarbeitung hängt vom spezifischen Kommando ab und muss auf geeignete Kommandos (siehe Kommandobeschreibungen) beschränkt sein
- 5) Die Anzahl nebenläufiger Kommandos muss insofern begrenzt ausfallen, dass je referenzierter Functional Unit (z.B. Kartenkontaktiereinheit, Display, ...) ein Kommando zurzeit abgearbeitet werden kann,
- 6) Eine Ausnahme muss die Functional Unit des Kommandointerpreters 'Cardterminal' bilden, welche Kommandos zur Statusabfrage und zum Kommandoabbruch, stets beantworten muss (z.B. "BUSY"),
- 7) Zu allen Kommandos muss ein Aufrufer über den Protokollrahmen eine eindeutige Sequenznummer senden, die der Aufrufer selbst vergibt und verwaltet, und welche zum Abbruch eines Kommandos referenziert werden kann.
- 8) Die Kommandos zur Statusabfrage oder zum Kommandoabbruch müssen auf eine korrespondierende Sequenznummer referenzieren und müssen einen Kommandobezogenen Status zurückliefern,
- 9) Die Eignung zu generell abbrechbaren Kommandos muss abhängig vom spezifischen Kommando und auf geeignete Kommandos (siehe Kommandobeschreibungen) beschränkt sein.

5.5.5 Betriebsmodi

Ein SICCT-Terminal darf einen oder mehrere Betriebsmodi bieten, von denen immer einer als 'Aktiver Betriebsmodus' für das gesamte Gerät selektiert sein muss.

Ein erster optionaler Betriebsmodus 'BCS' darf auf dem Basisbefehlsatz der MKT CT-BCS Spezifikation [MKT_10] basieren. Dieser Modus muss ausschließlich im Rahmen Möglichkeiten der MKT -Spezifikation gehalten sein und darf nur die Verarbeitung kontaktbehafteter synchroner und asynchroner Chipkarten erlauben.

Ein weiterer Betriebsmodus 'SICCT' muss eine erweiterte Funktionalität oberhalb des BCS-Modes abbilden. Die korrespondierenden Kartenterminalkommandos wurden in Form neu definierter SICCT-Kommandos festgelegt. Der Modus 'SICCT' muss den Zustand kontaktbehafteter Kartenkontaktiereinheiten sowie asynchrone Chipkarten steuern und kontrollieren können und kann weitere Funktionen hinsichtlich einer komfortableren Terminalansteuerung bieten. Des Weiteren darf der SICCT-Modus kontaktlose und / oder synchrone Chipkarten sowie die Umschaltungsmöglichkeit zum Chipkartenbetriebsmodus EMV unterstützen.

Zwischen den Modi kann je nach Fähigkeit und Kenntnis einer steuernden Entität (HOST) gewählt und gewechselt werden. Per Gerätekonfiguration muss ein Betriebsmodus als 'Standard-Betriebsmodus' aktiviert sein. Diesen Modus muss das Gerät automatisch nach einem Reset des Kartenterminals wählen.

Der 'Aktive Betriebsmodus' kann durch die steuernde Entität in jedem Modus über ein entsprechendes Kommando (SICCT SELECT CT MODE) umgeschaltet werden.

Im Folgenden werden die Modi und deren Befehlsumfang wie Betriebsverhalten dargestellt.

5.5.6 Betriebsmodus 'BCS' und Command Set

Der optionale Betriebsmodus 'BCS' muss den Zustand kontaktbehafteter Chipkarten-Kontaktiereinheiten für synchrone und asynchrone Chipkarten nach ISO 7816 (siehe [STD2 - [STD9]) über den BCS-Kommandosatz steuern und kontrollieren können.

Der Umfang und die Ausgestaltung der BCS-Befehle muss wie in der MKT-Spezifikation V1.0 [MKT_10] erfolgen. Verwendet die steuernde Entität (Applikation) den BCS - Kommandosatz, muss sich das SICCT abwärtskompatibel zu einem MKT verhalten.

Zusätzlich muss der Betriebsmodus 'BCS' das folgende SICCT-Kommando zusätzlich unterstützen

- SICCT SELECT CT MODE (siehe 5.8).

Dieser Betriebsmodus darf generell keine nebenläufige Kommandobearbeitung und keinen Kommandoabbruch unterstützen.

Informativ soll an dieser Stelle ein Auszug des Cardterminal Basic Command Set (CT-BCS) dargestellt werden. Weitere Detailinformationen müssen der MKT-Spezifikation Teil 4 'Basic Command Set' [MKT_10] entnommen werden.

SICCT-supported BCS-Commands (C-APDU)			
CardTerminal Basic Command	CLA Code (hex)	INS Code (hex)	Description
RESET	20	10	B1-Command: Perform an explicit reset of the CT ICC1 or ICC2: This command is a subset of the RESET CT. It is recommended to use the standardised RESET CT command.
RESET CT	20	11	Perform an explicit reset of the CT or a selected ICC Reset CT: Set the CT and all ICC-slots to reset state. Reset ICC: Cold Reset. Optional receive the ATR or HB information.
REQUEST ICC	20	12	Request an ICC within given time. The ICC will be set to reset state. Optionally display a message on the MCT's display.
GET STATUS	20	13	Request CardTerminal status information. Query which ICC slots contain an inserted ICC ready for use. Retrieve the MCT's firmware version.
DEACTIVATE	20	14	B1-Command: Deactivate / eject addressed ICC. This command is a subset of the EJECT CT. It is recommended to use the standardised EJECT ICC command.
EJECT ICC	20	15	Eject an ICC within given time. Deactivate addressed ICC. Optionally display a given message on the MCT's display in order to instruct the user to remove the inserted card. Optical and acoustic signals are also selectable.
INPUT	20	16	Query keyboard input from the user within given time. Select the message the MCT will display on its Display (i.e. LCD). Determine the amount of queried information. Indicate input characters to appear as asterisks for secure typing.
OUTPUT	20	17	Send application data to a functional unit Indicate the application label the MCT will show on its display. At the moment the only supported functional unit is the MCT's display.
PERFORM VERIFICATION	20	18	Query PIN within given time and interact with an ICC. Issues the query of a user PIN and performs the interaction with the ChipCard. Select the message the MCT will display on its LCD. Control, which PIN-handling instructions will be performed.
MODIFY VERIFICATION DATA	20	19	Complete transaction of changing a user PIN. Queries the old and the new PIN - each within a defined time. Perform interaction with an ICC. Select the display message, control the PIN handling instruction and perform transaction with an ICC.

Tabelle 8: Overview of the SICCT supported Basic Command Set

5.5.7 Betriebsmodus 'SICCT' und Command Set

Der Mode 'SICCT kompatibel' muss den Zustand kontaktbehafteter und kontaktloser Kartenkontaktier-einheiten steuern und darf den Betrieb synchroner wie asynchroner Chipkarten nach ISO 7816 und 14443 ermöglichen. Hinsichtlich einer komfortableren Terminalansteuerung, welche z.B. detailliertere Zustände und Geräteattribute zurückzugeben vermag, werden weitere Kommandos ergänzt, die außerhalb des eingeschränkten BCS-Funktionsumfangs liegen.

Die Nutzungsmöglichkeit und die Aktivierung dieser Betriebsart muss auf einer expliziten Anwahl durch die Anwendungsumgebung basieren. Diese darf entweder durch eine statische Konfiguration am Terminal (gewählter 'Standard-Betriebsmodus') oder durch eine Selektion über das SICCT-Kommando SELECT CT MODE erfolgen. Damit verfügt die Anwendungsumgebung über die Kenntnis, dass dieser spezielle Modus selektiert wurde, und kann von allen Funktionserweiterungen Gebrauch machen.

Die Strukturen der SICCT-Terminalkommandos müssen derjenigen für BCS-Kommandos entsprechen. Bei der Wahl des CLASS / INSTRUCTION-Bytes ist darauf zu achten, dass dieses keine Kollision mit dem existierenden BCS Kommandosatz bedeutet. Da es überwiegend um erweiterte Funktionalitäten des schon bestehenden BCS-Kommandosatz handelt, werden die INSTRUCTION-Bytes (INS) weitgehend beibehalten und eine neue Klasse (CLA) eingeführt.

SICCT Command	CLA Code (hex)	INS Code (hex)	P1	P2	Lc	Cmd Data	Le	Brief Description	
SICCT RESET CT / ICC	80	11	FU	CQ	cond	cond	opt	man	Perform Cold / Warm Reset and optional PPS. Reset of the - cardterminal device - contactless interfaces (option) - Chipcards, RFID tokens
SICCT REQUEST ICC	80	12	FU	CQ	cond	cond	opt	man	Request for chipcard or RFID token (option) presentation and monitoring for a time period.
SICCT GET STATUS	80	13	FU	CQ	cond	cond	man	man	Request parameter for the - Cardterminal - functional units - ChipCard, RFID tokens
SICCT EJECT ICC	80	15	FU	CQ	cond	cond	absent	man	Disable chipcard or RFID token (option) logically and / or electrically. Optionally: mechanical operation.
SICCT INPUT	80	16	FU	CQ	cond	cond	opt	man	Query input data by the user interface functions of the cardterminal - Keypad (Biometrical) Sensors (option)
SICCT OUTPUT	80	17	FU	CQ	man	man	absent	man	Send output data to the Cardterminal or a functional unit of the cardterminal. - Display Message - Printer Data / Message
SICCT PERFORM VERIFICATION	80	18	FU	CQ	man	man	opt	man	Process Card Holder Verification. Perform a password / PIN entry operation, build a chipcard command and perform verification by an addressed chipcard or RFID token
SICCT MODIFY VERIFICATION DATA	80	19	FU	CQ	man	man	opt	man	Perform modify operation of Card Holder Verification Data Password / PIN
SICCT SELECT CT MODE	80	20	FU	CQ	cond	opt	opt	man	Select Operation / Command Set Mode - BCS Mode - SICCT Mode

SICCT Command	CLA Code (hex)	INS Code (hex)	P1	P2	Lc	Cmd Data	Le	Brief Description	
SICCT COMFORT_ AUTHENTICATION	80	21	FU	CQ	opt	opt	opt	opt	Support command for authentication. Learns and stores an authentication dataset with - authentication data - and Serialnumber (ICCSN).
SICCT COMFORT_ ENROLL	80	22	FU	CQ	opt	opt	opt	opt	Support command for authentication. Learns and stores an authentication dataset with - authentication data and - Serialnumber (ICCSN).
SICCT SET STATUS	80	23	FU	CQ	man	man	absent	man	Set parameter (Data Object) for the - cardterminal - functional units. - chipcards.or RFID tokens
SICCT DOWNLOAD INIT	80	24	FU	CQ	cond	cond	man	man	Start of FW download
SICCT DOWNLOAD DATA	80	25	FU	CQ	man	man	man	man	Data transportation during FW down- load
SICCT DOWNLOAD FINISH	80	26	FU	CQ	cond	cond	man	man	Manifestation and completion of a FW download
SICCT CONTROLCOMMAND	80	27	FU	CQ	man	man	absent	man	Abort or termination of SICCT command.
SICCT INIT CT SESSION	80	28	FU	CQ	man	man	man	man	Init and open a Cardterminal Session .
SICCT CLOSE CT SSESSION	80	29	FU	CQ	man	man	absent	man	Close a Cardterminal Session.
FU = Functional Unit									
CQ = Command Qualifier									
cond = conditional									
opt = optional									
man = mandatory									

Tabelle 9: Overview of the SICCT Command Set

Die BCS sowie SICCT-Kommandos müssen einen Rückgabewert produzieren, der gemäß den ISO 7816-4-Konventionen als Zwei-Byte Statuswort (SW1SW2) an den Aufrufer zurückgegeben werden muss.

Return Code SW1 SW2		Return of ...	Meaning
'90 00'	☺	all Commands	Synchronous ICC , CT / ICC or RFID token reset successful. Synchronous ICC presented, reset successful. Command successful. Change of verification data successful
'90 01'	☺	RESET EJECT	Asynchronous ICC or RFID token, reset successful. Asynchronous ICC presented, reset successful. Command successful, card or RFID token removed.
All other codes	☹	all Commands	ERROR or Warning.

Tabelle 10: CardTerminal Return Codes SW1SW2 for successful operation

5.5.8 Unterstützte Chipkarten

Je nach aktivem Betriebsmodus und Kommandosatz müssen die folgenden Chipkartenarten unterstützt werden.

Chipcard Type	Normative Reference(s)	Chipcard Protocol		BCS Mode	SICCT Mode	Remark
Asynchronous Smartcard	ISO7816-3	T=0	T0CP	✓	✓	Microprocessor Chipcard with character transmission protocol T=0
		T=1	T1BP	✓	✓	Microprocessor Chipcard with block transmission protocol T=1
Synchronous Chipcard	ISO7816-10 Type 1	S=8	SDAP	✓	opt	Memory Chipcard with I ² C-Bus Protocol
		S=9	3WBP	✓	opt	Memory Chipcard with 3 Wire Bus -Bus Protocol
		S=10	2WBP	✓	opt	Memory Chipcard with 2 Wire Bus -Bus Protocol
	ISO7816-10 Type 2	S=11	FCBP		opt	Function Control Bus Protocol
RFID-Token (PICC)	ISO14443-4	T=CL	T=CL		opt	Proximity Chipcard (RFID token) with block transmission protocol
Note: 'opt' = optional						

Tabelle 11: Unterstützte Chipkarten und Protokolle

5.5.8.1 Kommandos für synchrone Chipkarten

Zur vereinfachten Unterstützung von synchronen kontaktbehafteten Chipkarten nach ISO 7816-10 [STD9], kann das SICCT-Kartenterminal eine interne automatische Umsetzung eines definierten Satzes ISO7816-4 strukturierter Interindustry Commands auf Chipspezifische Busprotokolle und Aktionen bieten.

Bietet ein SICCT-Kartenterminal die optionale Unterstützung von synchronen Chipkarten, müssen die folgenden Interindustry Commands entsprechend [MKT_10] Teil 7 unterstützt werden.

Interindustry Command	Normative Reference	Condition	Description
SELECT FILE	ISO 7816-4	always to support	Selektieren einer Chipkartenanwendung
READ BINARY			Lesen eines selektierten Datenbereichs
UPDATE BINARY			Datenmodifikation eines selektierten Datenbereichs
VERIFY	ISO 7816-8	for chipcards with Verification Data (security function)	Vergleich von Verifikationsdaten mit den in der Karte gespeicherten Verifikationsdaten
CHANGE REFERENCE DATA			Änderung der in der Karte gespeicherten Verifikationsdaten

5.5.9 Functional Units

Functional Units (FU) bilden adressierbare Ressourcen des Kartenterminals an die externe Welt ab und können in einem Kommando (C-APDU) über den Parameter P1 innerhalb des in Tabelle 12 aufgeführten Wertebereichs angesprochen werden. Je nach technischer Ausprägung und Betriebsmode des SICCT-Terminals können unterschiedliche Funktionseinheiten zur Verfügung stehen.

Die Kodierung von Functional Units erfolgt über die Kommandoparameter P1 und ggf. P2 in der Direct Coding (s. 5.5.9.3) oder Referenced Coding (s. 5.5.9.4) Darstellung.

5.5.9.1 Functional Units und ihre Multiplizität

Folgende Abbildung zeigt die Multiplizität der Functional Units.

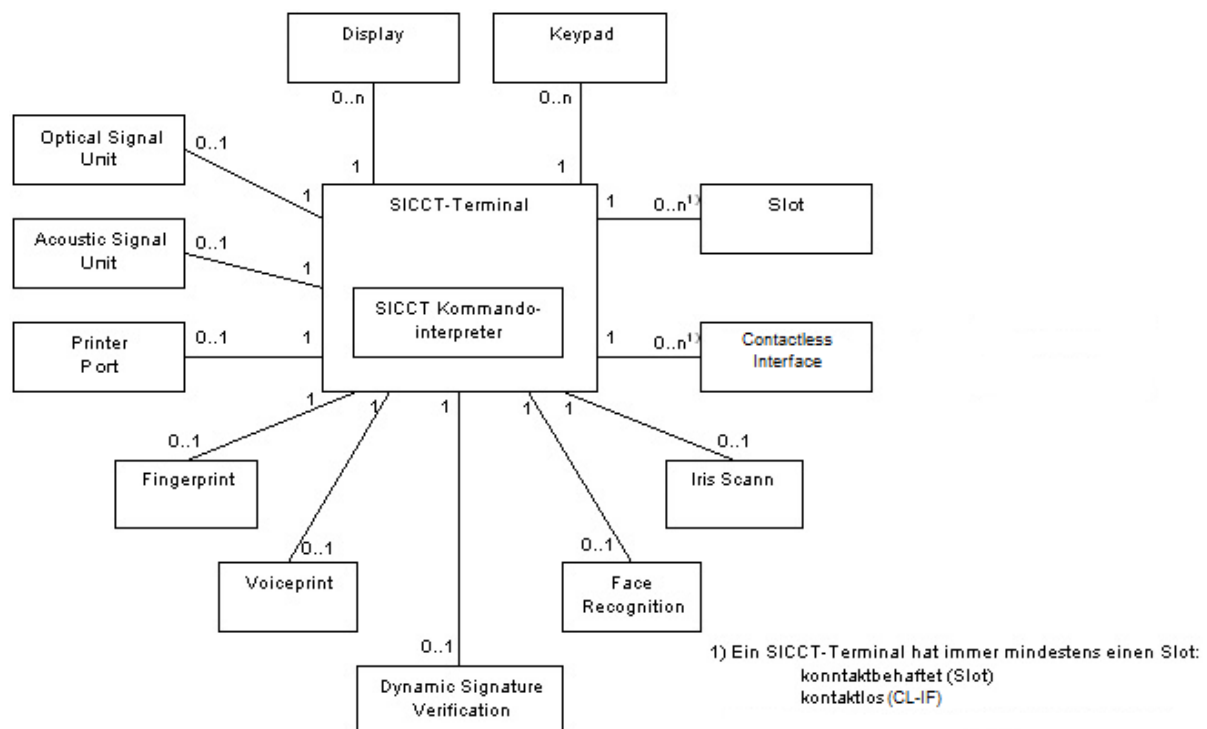


Abbildung 2: SICCT Functional Units

5.5.9.2 Ressourcentabelle

Das SICCT muss die Möglichkeit bieten, Functional Units mit einem Bezeichner (sog. Friendly Name) versehen zu können. Zu diesem Zweck muss das SICCT eine Ressourcentabelle mit maximal 256 möglichen Einträgen verwalten, die für jede verfügbare Functional Unit, einen Schlüsselwert, einen Typwert sowie einen Friendly Name, in Form eines ASCII-Strings, beinhalten muss.

Der Typwert muss als Ein-Byte-Kodierung erfolgen und klassifiziert eine Functional Unit nach Ihrer Funktionalität in folgende Typen.

SICCT Resource table for maximal 256 Functional Units				
Table index	FU Type (hex)	FU-Index-Value (hex)	Friendly Name	Remark
0	00	00	"Terminal"	Cardterminal
1	00	01	"ICC SLOT 1"	1 st Contact bound ChipCard Slot
:	:	:	:	:
0<n>	00	0E	"ICC SLOT x"	Contact bound ChipCard Slot x
0<n+1>	10	00	RFU	RFU
0<n+2>	10	01	"CL-IF 1"	1 st Contactless Interface
:	:	:	:	:
:	10	0E	"CL-IF 14"	14 th Contactless Interface
:	40	00	"Standard Display"	
:	40	01	"Display 1"	Additional Display 1
:	:	:	:	:
:	50	00	"Standard Keypad"	1 st Keypad
:	50	01	"Keypad 1"	Additional Keypad 1
:	:	:	:	:
:	60	00	"Printer Port"	(Optional) Printer Port
:	:	:	:	:
:	70	00	"Fingerprint Sensor"	
:	:	:	:	:
:	70	02	"Dynamic Signature Verification Unit"	
:	:	:	:	:

Tabelle 12: Typkennzeichnung von Functional Units in der Ressourcentabelle

Der FU-Index-Value adressiert die Functional Unit und gibt zusammen mit dem FU-Type den Wert an, der im Value-Teil eines FUI DO (s. 5.5.10.9) erscheinen kann, wenn eine Referenced Coding Darstellung von P1 erfolgt."

Der Wertebereich des Indexwerts muss abhängig von der maximalen Anzahl der Functional Units gewählt sein, und darf ein Byte ('00' bis 'FF') umfassen.

Friendly Names dürfen nur durch ASCII-Kodierte Zeichenketten mit einer Maximallänge von 32 Zeichen dargestellt sein.

5.5.9.3 Direct Coding

Folgende Tabelle zeigt die verfügbaren Werte in der Direct Coding-Darstellung für den Parameter P1. Einige Werte dürfen auch im Parameter P2 erscheinen, sofern ein Kommando dieses erfordert.

Functional Units (Direct Coding Representation)						
Functional Unit	Meaning	Direct Coding (1 Byte)	Mode			Description
			BCS	SICCT		
Address the Cardterminal (whole device)						
CT	CT-Kernel	'00'	✓	✓	man	Address the Cardterminal device with all attached / controlled resources / functional units.
Address contact bound ChipCard Interface						
ICCN	ICC-Interface n	'0x'	✓	✓		Address single ICC Interface.
ICC1	ICC-Interface 1	'01'			man	1 st contact bound ChipCard Interface Unit
:	:	:			:	:
ICC14	ICC-Interface 14	'0E'			opt	14 th ChipCard Interface Unit
Address Contactless ChipCard Interface						
CL-IFn	CL-IF n	'1x'	-	✓	opt	Address single Contactless Interface
CL-IF1	CL-IF 1	'11'			opt	1 st Contactless Interface
:	:	:			:	:
CL-IF14	CL-IF 14	'1E'			opt	14 th Contactless Interface
Address Human Interface Device Unit						
CT-Display (0)	Standard Display	'40'	✓	✓	opt	1 st Cardterminal controlled Display (0) (Standard Display)
CT-Display 1	Display1	'41'	-	✓	opt	Additional Cardterminal controlled Display 1
:	:	:	-		:	:
CT-Display 14	Display 14	'4E'	-		opt	Additional Cardterminal controlled Display 14
CT-Keypad	Standard Keypad	'50'	✓	✓	opt	1 st Cardterminal controlled Keypad
CT Keypad 1	Keypad 1	'51'	-		opt	Additional Cardterminal controlled Keypad 1
:	:	:	-		opt	
CT Keypad 14	Keypad 14	'5E'	-		opt	Additional Cardterminal controlled Keypad 14
CT-Printer	Printer port	'60'	✓	✓	opt	Cardterminal controlled Printer port
Address Human Interface Device Unit of Type 'Biometric Sensor'						
Biometrical Unit		'7x'	✓	✓	opt	2 nd nibble codes the unit type.
CT-FP	Finger Print Sensor Unit	'70'			opt	Cardterminal controlled Finger Print Sensor
CT-VP	Voiceprint Unit	'71'			opt	Cardterminal controlled Voiceprint Sensor
CT-DSV	Dynamic Signature Ver- ification Unit	'72'			opt	Cardterminal controlled Dynamic Signature Veri- fication Sensor / System
CT-FR	Face Recognition Unit	'73'			opt	Cardterminal Face Recognition Sensor / System
CT-I	Iris Unit	'74'			opt	Cardterminal Iris Unit Sensor / System
other Biometrical Units		'75' - '7F'	-	✓	opt	RFU
SICCT specific FUs		'80' - '9F'	-	✓	opt	RFU
Vendor specific FUs		'A0' - 'FE'	-	✓	opt	Vendor defined Functional Unit

Tabelle 13: Functional Units - Direct Coding Representation

5.5.9.4 Referenced Coding

Für den Fall, dass die Direct Coding Darstellung nicht ausreichen sollte oder nicht gewünscht wird, kann mittels der Referenced Coding Darstellung eine Erweiterung der durch P1 selektierten Functional Units

erfolgen. Der P1-Wert 'FF' ('Escape-Zeichen') signalisiert dem Kommandointerpreter, dass im Command Field über ein TLV-Datenobjekt eine oder mehrere Functional Units referenziert werden.

5.5.9.5 Mapping zwischen Direct Coding und Referenced Coding

Die Abbildung der "Direct Coding" Methode auf die "Reference Coding" und somit die Abbildung eines 1 Byte P1 Wertes auf einen 2 Byte Primärschlüssel der Ressourcentabelle muss wie folgt möglich sein: Das MSB-Nibble von P1 ("Direct Coding") wird zum MSB-Nibble im Type-Kennzeichen der Ressourcentabelle. Das LSB-Nibble von P1 wird zum LSB-Nibble im Index-Kennzeichen der Ressourcentabelle. Das LSB-Nibble des Type-Kennzeichens und das MSB-Nibble des Index-Kennzeichens werden zu '0' gesetzt. Nachfolgend ein Beispiel für das 1. CL-IF:

Abbildung "Direct Coding" auf Reference Coding für den Zugriff auf die Ressourcentabelle am Beispiel des 1. RFID Slots:

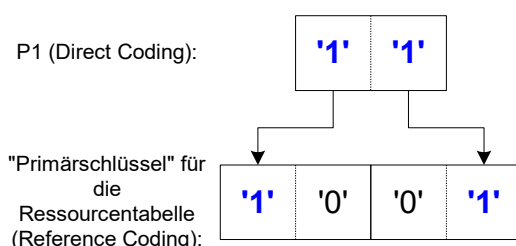


Abbildung 3: Beispielabbildung für Direct Coding auf Ressourcentabelle

Die Ein-Byte "Direct Coding" Methode darf für den Indexwert (LSB Nibble von P1) nur Werte zwischen '0' und 'E' (Hexadezimal) zulassen, so dass das Mapping konfliktfrei möglich ist.

Functional Units (Referenced Coding Representation)					
Functional Unit	Meaning	Referenced Coding (2 Byte)	Mode		Description
			SICCT		
Address the Cardterminal (whole device)					
✓	CT-Kernel	'0000'	✓	man	Address the Cardterminal device with all attached / controlled resources / functional units.
Address contact bound ChipCard Interface					
✓	ICC-Interface n	'00xx'	✓		Address single ICC Interface.
ICC1	ICC-Interface 1	'0001'		man	1 st contact bound ChipCard Interface Unit
:	:	:		opt	:
ICC255	ICC-Interface 255	'00FF'		opt	255 th ChipCard Interface Unit
Address Contactless ChipCard Interface					
CL-IFn	CL-IF n	'100x'	✓	opt	Address single Contactless Interface.
CL-IF1	CL-IF 1	'1001'		opt	1 st Contactless Interface
:	:	:		opt	:
CL-IF14	CL-IF 255	'10FF'		opt	255 th Contactless Interface
Address Human Interface Device Unit					
CT-Display (0)	Standard Display	'4000'	✓	opt	1 st Cardterminal controlled Display (0) (Standard Display)
CT-Display 1	Display1	'4001'	✓	opt	Additional Cardterminal controlled Display 1
:	:	:		opt	
CT-Display 14	Display 14	'400E'		opt	Additional Cardterminal controlled Display 14
CT-Keypad	Standard Keypad	'5000'	✓	opt	1 st Cardterminal controlled Keypad
CT Keypad 1	Keypad 1	'5001'		opt	Additional Cardterminal controlled Keypad 1
:	:	:		opt	
CT Keypad 14	Keypad 14	'500E'		opt	Additional Cardterminal controlled Keypad 14
CT-Printer	Printer port	'6000'	✓	opt	Cardterminal controlled Printer port
Address Human Interface Device Unit of Type 'Biometric Sensor'					
Biometrical Unit		'700x'	✓	opt	2 nd nibble codes the unit type.
CT-FP	Finger Print Sensor Unit	'7000'		opt	Cardterminal controlled Finger Print Sensor
CT-VP	Voiceprint Unit	'7001'		opt	Cardterminal controlled Voiceprint Sensor
CT-DSV	Dynamic Signature Verification Unit	'7002'		opt	Cardterminal controlled Dynamic Signature Verification Sensor / System
CT-FR	Face Recognition Unit	'7003'		opt	Cardterminal Face Recognition Sensor / System
CT-I	Iris Unit	'7004'		opt	Cardterminal Iris Unit Sensor / System
other Biometrical Units		'7005' - '70FF'	✓	opt	RFU
SICCT specific FUs		'8000' - '90FF'	✓	opt	RFU
Vendor specific FUs		'A000' - 'FOFE'	✓	opt	Vendor defined Functional Unit

Tabelle 14: Functional Units - Referenced Coding Representation

Die folgende Tabelle zeigt einige Beispiele für Referenced Coded Values.

Referenced Coded Value (hex)	FU Type (hex)	FU-Index-Value (hex)	Friendly Name	Remark
0000	00	00	"Terminal"	Cardterminal
0001	00	01	"ICC SLOT 1"	1 st Contact bound ChipCard Slot
:	:	:	:	:
000E	00	0E	"ICC SLOT 14"	Contact bound ChipCard Slot 14
000F	00	0F	"ICC SLOT 15"	Contact bound ChipCard Slot 15
0010	00	10	"ICC SLOT 16"	Contact bound ChipCard Slot 16
:	:	:	:	:
00FF	00	FF	"ICC SLOT 255"	Contact bound ChipCard Slot 255
:	:	:	:	:
1001	10	01	"CL-IF 1"	1 st CL-IF
:	:	:	:	:
100E	10	FF	"CL-IF 255"	255 th CL-IF
4000	40	00	"Standard Display"	1 st Display
4001	40	01	"Display 1"	Additional Display 1
:	:	:	:	:
5000	50	00	"Standard Keypad"	1 st Keypad
5001	50	01	"Keypad 1"	Additional Keypad 1
:	:	:	:	:
6000	60	00	"Printer Port"	One (optional) Printer Port
7000	70	00	"Fingerprint Sensor"	One (optional) Fingerprint Sensor
:	:	:	:	:
7002	70	02	"Dynamic Signature Verification Unit"	One (optional) Dynamic Signature Verification Unit
:	:	:	:	:
8000	80	00	"1 st SICCT Generic FU"	
:	:	:	:	:
A000	A0	00	"1 st Vendor defined generic FU"	

Tabelle 15: Typkennzeichnung von Functional Units in der Ressourcentabelle

5.5.10 Data Objects

Datenobjekte (DO) transportieren Statusinformationen über das Kartenterminal, adressierbare Functional Units und Chipkarten. In Anlehnung an bewährte Standards für Chipkarten und Kartenterminals muss die Kodierung von SICCT-Datenobjekten nach ASN.1 (Tag-Length-Value) erfolgen. Die Datenobjekte sind kontext- bzw. kommandospezifisch, so dass Mehrfachnennungen von definierten Tag-Werten zulässig sind. Die Präsenz, Bedeutung und Gültigkeit eines Datenobjekts muss vor dem Hintergrund eines Kontextes stets eindeutig sein.

Das Design der Datenobjekte muss dem Prinzip der MKT-Spezifikation [MKT_10] folgen, mit dem Ziel, die enthaltenen Datenobjekte des Basic Command Sets weitgehend unverändert erhalten zu können. Für ergänzte Funktionalitäten werden weitere Datenobjekte eingeführt, die im Betriebsmodus SICCT aber nicht im Modus BCS nutzbar sind.

5.5.10.1 Data Objects supported in BCS-Mode

Für die optionale Betriebsart 'BCS' muss der Umfang und die Ausgestaltung der Datenobjekte nach den Definitionen der MKT-Spezifikation V1.0 [MKT_10] erfolgen, damit sich ein SICCT – Terminal abwärts-kompatibel zu einem MKT – Terminal verhalten kann.

Data Object	Type	Tag	Mode			IN / OUT	Description
			BCS	SICCT			
ATR	Byte Sequence	-	✓		man	OUT	One single Answer To Reset byte sequence of a ChipCard
HB	Byte Sequence	-	✓		man	OUT	One single Historical Byte sequence of a ChipCard
CTM DO	TLV DO	'46'	✓	✓	man	OUT	CardTerminal Manufacturer Data Object (pre-issuing data)
CMD DO	TLV DO	'52'	✓	✓	opt	IN	Command-to-perform
ICCS DO	TLV DO	'80'	✓	✓	man	OUT	ICC Status DO
FU DO	TLV DO	'81'	✓	✓	man	OUT	Functional Unit Data Object
APPL DO	TLV DO	'50'	✓	✓	man	IN	Application Label Data Object codes a Display Message.
WT DO	TLV DO	'80'	✓	✓	man	IN	Waiting Time Data Object codes time periods for user interactions - card insertion interval - card removal intervaltime until 1st input by Keypad

Tabelle 16: Overview SICCT Data Objects in BCS-Mode

5.5.10.2 Data Objects supported in SICCT-Mode

Data Object	Type	Tag	Mode			IN / OUT	Description
			BCS	SICCT			
CTM DO	TLV DO	'46'	✓	✓	man	OUT	CardTerminal Manufacturer Data Object (pre-issuing data) Used within SICCT commands SICCT GET STATUS
APPL DO	TLV DO	'50'	✓	✓	man	IN	Application Label Data Object codes a Display Message. Used within SICCT commands - SICCT REQUEST ICC - SICCT EJECT ICC - SICCT INPUT - SICCT OUTPUT - SICCT PERFORM VERIFICATION - SICCT MODIFY VERIFICATION DATA
CMD DO	TLV DO	'52'	✓	✓	opt	IN	Command-to-perform Used within SICCT commands - SICCT PERFORM VERIFICATION - SICCT MODIFY VERIFICATION DATA
PPSR DO	TLV DO	'53'		✓	opt	IN	PPS - Request Byte Sequence
						OUT	PPS - Response Byte Sequence Used within SICCT commands SICCT RESET CT / ICC
ATR DO	TLV DO	'5F41'		✓	man	OUT	Answer To Reset information of a ChipCard or RFID-token (PICC) Used within SICCT commands - SICCT REQUEST ICC - SICCT RESET CT / ICC
HB DO	TLV DO	'5F52'		✓	man	OUT	Historical Bytes of an ATR of a ChipCard or RFID-token (PICC) Used within SICCT commands - SICCT REQUEST ICC - SICCT RESET CT / ICC

DLPARAM DO	TLV DO	'73'		✓	man	OUT	Download Parameter Data Object - Max. Block size - Timeout Value Used within SICCT commands - SICCT DOWNLOAD INIT - SICCT DOWNLOAD DATA
DLDATA DO	TLV DO	'73'		✓	man	IN	Download Data Object Used within SICCT commands SICCT DOWNLOAD DATA
DLTERM DO	TLV DO	'73'		✓	man	OUT	Download Termination Data Object Used within SICCT commands SICCT DOWNLOAD FINISH
WT DO	TLV DO	'80'	✓	✓	man	IN	Waiting Time Data Object codes time periods for user interactions - card insertion interval - card removal interval - RFID token (PICC) presentation interval - RFID token (PICC) removal interval - time until 1st input entry by Keypad - time until next input entry by Keypad. - time for one input sequence (i.e. entry of 12-digit-PIN) - overall time interval for a PIN-entry Used within SICCT commands - SICCT REQUEST ICC - SICCT EJECT ICC - SICCT INPUT - SICCT OUTPUT - SICCT PERFORM VERIFICATION - SICCT MODIFY VERIFICATION DATA
ICCS DO	TLV DO	'80'	✓	✓	man	OUT	ICC Status Data Object Used within SICCT commands - SICCT GET STATUS - SICCT RESET CT / ICC
FU DO	TLV DO	'81'	✓	✓	man	OUT	Functional Unit Data Object Used within SICCT commands - SICCT RESET CT / ICC - SICCT GET STATUS
CLIF DO	TLV DO	'83'		✓	opt	OUT	CL-IF Status Data Object Used within SICCT commands SICCT GET STATUS
FUI DO	TLV DO	'84'		✓	opt	IN	Functional Unit Index Data Object for Referenced Coding Representation of Functional Units. Used within SICCT commands - as equivalent P1 (reference coding) within all SICCT commands - within FU CON DO for SICCT commands which need an execution context of multiple FUs.
CS DO	TLV DO	'85'		✓	opt	IN OUT	Character Set Data Object indicates display type and application label specific character set. Used within SICCT commands SICCT GET STATUS
SMTBD DO	TLV DO	'A0'		✓	opt	IN	SICCT Message-To-Be-Displayed Data Object combines an application label with a Character Set Data Object. Used within SICCT commands - SICCT REQUEST ICC - SICCT EJECT ICC - SICCT INPUT - SICCT OUTPUT - SICCT PERFORM VERIFICATION - SICCT MODIFY VERIFICATION DATA

FU NAME DO	TLV DO	'A1'		✓	opt	IN	Functional Unit Name Data Object associating a friendly name for one FUI DO. Used within SICCT commands ▪ SICCT GET STATUS ▪ SICCT SET STATUS
						OUT	
FU CON DO	TLV DO	'A2'		✓	opt	IN	Functional Unit Context Data Object Used within SICCT commands which need a execution context of multiple FUs ▪ SICCT REQUEST ICC ▪ SICCT EJECT ICC ▪ SICCT INPUT ▪ SICCT OUTPUT ▪ SICCT PERFORM VERIFICATION ▪ SICCT MODIFY VERIFICATION DATA
CTS DO	TLV DO	'63'		✓	opt	OUT	Reduced / Cardterminal Status Information Used within SICCT commands ▪ SICCT RESET CT / ICC ▪ SICCT GET STATUS
INTFS DO	TLV DO	'65'		✓	man	OUT	ICC Interface Status Data Object Used within SICCT commands ▪ SICCT RESET CT / ICC
INTFC DO	TLV DO	'66'		✓	man	OUT	ICC Interface Capabilities Data Object Used within SICCT commands ▪ SICCT GET STATUS
DSPLC DO	TLV DO	'67'		✓	opt	OUT	Display Capabilities Data Object Used within SICCT commands ▪ SICCT GET STATUS
SEQNO DO	TLV DO	'68'		✓	man	IN OUT	Sequence Number Data Object Used within SICCT commands ▪ SICCT CONTROL COMMAND
CTSESS DO	TLV DO	'69'		✓	man	IN OUT	Cardterminal Session Data Object Used within SICCT commands ▪ SICCT INIT CT SESSION ▪ SICCT CLOSE CT SESSION
KPC DO	TLV DO	'6A'		✓	opt	OUT	Keypad Capabilities Data Object Used within SICCT commands SICCT GET STATUS

Tabelle 17: Overview SICCT Data Objects in SICCT Mode

5.5.10.3 TLV-Data Objects

Als Codierungstechnik für Tag-Length-Value - Datenobjekte (TLV-DO) müssen die 'Basic Encoding Rules (BER)' der ISO-Codierungskonvention 'Abstract Syntax Notation One (ASN.1)' verwendet werden.

Folgende Tabelle zeigt die generellen Feldkodierungen. Entsprechende Begrenzungen und Wertebereichsangaben finden sich in den Definitionen der entsprechenden Datenobjekte (s. ab 5.5.10.4) und Kommandobeschreibungen.

Field	Bytes	Coding				
Tag	one byte coding					
	1	b8 b7	00	Universal		
			01	Application Context		
			10	Context specific		
			11	Private		
		b6	0	primitive data object		
			1	constructed data object		
		b5...b1	00000 ... 11110	'00' ... '1E'	0 ... 30	Tag Number
	two bytes coding					
	1	b8 b7	see above			
		b6	0	primitive data object		
			1	constructed data object		
b5...b1		11111	Indicates two byte tag: tag number contained in the next (2 nd) byte			
2	b8	0	'1F' ... '7F'	31 ... 127	Tag Number	
	b7...b1	0011111 ... 1111111				
LEN	one byte coding					
	1	b8	0	'00' ... '7F'	Range <LEN>	
		b7...b1	0000000 ... 1111111		0 ... 127	
	two byte coding					
	1	b8	1	'81'	129	
		b7...b1	0000001			
	2	b8...b1	00000000 ... 11111111	'00' ... 255	Range <LEN>	
			0 ... 255			
			Note: It shall be allowed to start with zero.			
	three byte coding					
	1	b8	1	'82'	130	
		b7...b1	0000010			
	2	b8...b1	00000000 ... 11111111	High Byte '00' ... 'FF'	Range <LEN> < HighByte LowByte > Note: It shall be allowed to start with zero.	
3	b8...b1	00000000 ... 11111111	Low Byte '00' ... 'FF'	'0000' ... 'FFFF'	0 ... 65535	
Value	<LEN> == 0: absent - no value data - empty data object					
	1 : < LEN>	1 ... 65535 bytes of data				

Tabelle 18: TLV-Coding according to ASN.1-BER

5.5.10.4 Answer-To-Reset Data Object

Das Answer-To-Reset Datenobjekt (ATR DO) muss im SICCT – Mode unterstützt werden und muss die unveränderte Reset-Antwort einer Chipcard oder RFID Token (PICC) zurückliefern, wie diese von der Chipkarte an das Kartenterminal gesendet und vom Kartenterminal empfangen wurde.

Das DO muss in Anlehnung an ISO 7816-6 ASN.1 kodiert werden.

Answer-To-Reset Data Object (ATR DO)						
TAG	'5F41'		Two byte tag according to ISO 7816-6: Answer-To-Reset			
			Tag coding according to ASN.1 BER see 5.5.10.3			
			BER-Coding: Application context, primitive, Tag-Number = 65 ('41')			
LEN	LEN coding see 5.5.10.3					
	one or two bytes coding - LEN in the range of: 0 <= LEN <= 255					
	'00' ... '7F'		0 <= LEN <= 127	One byte coding		
	'81'	'80' ... 'FF'	128 <= LEN <= 255	Two byte coding		
VALUE	ICC	ICC Reset information max. 32 Bytes				
	ICC	7816-3 ICC	Asynchronous CC ATR	2 ... 32 Bytes	according to 7816-3	
		7816-10 ICC	Synchronous CC ATR	2 ... 4 Bytes	according to 7816-10 [STD9]	
	PICC	PICC Reset information - max. 255 Bytes				
		PICC-A	Answer-To-Select Information (ATS)		3 ... 255 Bytes	PICC-A ATS according to ISO 14443-4
		PICC-B	ATQB ATTRIB Rsp.		17 Bytes	Concatenation of ISO 14443-3 - Answer-To-Query B (ATQB) - ATTRIB Response
	Answer-To-Query (ATQB) ATTRIB Response					

Byte	1	2	3	4	5	6	7	8	9	10	11	12	13	14
ATQB	'50'	PUPU 4 bytes				Application Data 4 bytes				Protocol Info 3 bytes			CRC_B 2 bytes	
		Pseudo Unique PICC Identifier Type B				AFI	CRC_B (AID)	Number. of Applications		according to ISO 14443-3			Cyclic Redundancy Check Error Detection Code Type B	
		MS B			LSB								MSB	LSB

Application Family Identifier (AFI)				
family	sub-family	Meaning according to ISO 14443-3	Remark	
MSB	LSB	Type of requested (responding) PICC		
'0'	'0'	All families and sub-families		
X	'0'	All sub-families of family X	application preselection	
X	Y	All sub-families Y of family X		
'0'	Y	Proprietary sub-family Y only		
'1'	'0', 'Y'	Transport		
'2'	'0', 'Y'	Financial		
'3'	'0', 'Y'	Identification		
'4'	'0', 'Y'	Telecommunication		
'5'	'0', 'Y'	Medical		
'6'	'0', 'Y'	Multimedia		
'7'	'0', 'Y'	Gaming		
'8'	'0', 'Y'	Data Storage		
'9' ... 'F'	'0', 'Y'	RFU		

Byte	1		2	3	Meaning
ATTRIB Response	MBLI	CID	CRC_B		ATTRIB Response according to ISO 14443-3 given to an ATTRIB command without higher layer Information (command) field.
	Maximum Buffer Length Index Type B	Card Identifier	Cyclic Redundancy Check Error Detection Code Type B 2 bytes		
		Range: 0 ... 14			

5.5.10.5 Historical Byte Data Object

Das Historical Byte Data Object (HB DO) muss im SICCT – Mode unterstützt werden und muss eine Teilmenge des Answer-To-Reset Datenobjekts darstellen.

Das DO wird in Anlehnung an ISO 7816-6 ASN.1 kodiert.

Historical Byte Data Object (HB DO)				
TAG	'5F52'		Two byte tag according to ISO 7816-6: Historical Bytes	
			Tag coding according to ASN.1 BER see 5.5.10.3	
			BER-Coding: Application context, primitive, Tag-Number = 82 ('52')	
LEN	LEN coding see 5.5.10.3			
	one or two bytes coding - LEN in the range of: 0 <= LEN <= 252			
	'00' ... '7F'		0 <= LEN <= 127	One byte coding
	'81'	'80' ... 'FC'	128 <= LEN <= 252	Two byte coding
VALUE	Part of the CC / PICC Reset information providing the Historical or Application Information Bytes			
	CC Historical Bytes - LEN in the range of 2 ... 15			
	HB	Historical Bytes	0 ... 2 Bytes	Synchronous CC HB
			0 ... 15 Bytes	Asynchronous CC HB
	PICC Application Information Bytes - LEN in the range of 0 or 4 ... 252			
	AIB	Application Information Bytes	0 or 4 ... 252 Bytes	PICC AIB according to ISO 14443-4

5.5.10.6 CardTerminal Manufacturer Data Object

Das CardTerminal Manufacturer Data Object (CTM DO) muss im SICCT – Mode unterstützt werden und muss invariante Angaben zum Kartenterminal (pre-issuing data) beinhalten, die der Hersteller zum Zweck der Erkennung zum Zeitpunkt der Herstellung eingebracht hatte.

Diese Angaben müssen der Erkennung folgender Informationen dienen

- Kartenterminalhersteller (CardTerminal Manufacturer , CTM)
- Kartenterminaltyp (CardTerminal Type, CTT)
- Version der Kartenterminal – Software (CardTerminal Software Version, CTSV)
- SICCT Version, d.h. Angabe des kompatiblen SICCT- Spezifikationsstandards
- (optionale) herstellerspezifische Kenndaten (Discretionary Data, DD),
z. B. Seriennummer (i.e. Serial Number).

CardTerminalManufacturer Data Object (CTM DO)				
TAG	'46'	One byte tag according to MCT-Specifications: CardTerminalManufacturer Data Object		
		Tag coding according to ASN.1 BER see 5.5.10.3		
		BER-Coding: Application context, primitive, Tag-Number = 82 ('52')		
LEN	LEN coding see 5.5.10.3			
	one byte coding - LEN in the range of: 0 <= LEN <= 127			
	'0F' ... '7F'	15 <= LEN <= 127		One byte coding
VALUE	Cardterminal pre-issuing data			
	CTM	CTT	CTSV	[Discretionary Data]
	man	man	Man	opt
	5 Byte	5 Byte	5 Byte	0 <= LEN <= 112 Bytes
	Cardterminal Man- ufacturer	SICCT Version	Cardterminal Soft- ware Version	Discretionary Data
	RID provided	SICCT specification provided	Vendor provided	Vendor provided

Data		Len		Description				
CTM	Cardterminal Manufacturer	5	man	5 Byte ASCII String - padded with Space ('20')				
				Unique Manufacturer Coding value according to the RID German National Registration Authority				
				2 byte Country Code according to ISO 3166		3 byte Manufacturer-Acronym.		
				Germany				
				'44'	'45'	'20'	'20'	'20'
CTT	SICCT Cardterminal Version	5	man	5 Byte ASCII String - padded with Space ('20')				
				SICCT Specification Version				
				SICCT Major Number	SICCT Major Number	SICCT Minor Number	SICCT Minor Number	Release Number / Char- acter or Padding
				'30'	'31'	'32'	'30'	'20'
				CTSV	Cardterminal Software Ver- sion	5	man	5 Byte ASCII String - padded with Space ('20')
Major Number	Major Number	Minor Number	Minor Number					Release Number or Char- acter
'30'	'31'	'30'	'30'					'20'
DD	Discretionary Data	0 ... 112	opt	vendor proprietary or domain specific format				

5.5.10.7 ICC Status Data Object

Das ICC Status Data Object (ICCS DO) muss im SICCT – Mode unterstützt werden und muss ein Objekt darstellen, um den Betriebsstatus von kontaktbehafteten Chipkarten (nach ISO 7816-3 [STD6] / 7816-10 [STD9]) vom Kartenterminal mittels eines einzigen Datenobjekts abfragen zu können.

ICC Status Data Object (ICCS DO)				
TAG	'80'	One byte tag according to MCT-Specifications:: ICC Status Data Object		
		Tag coding according to ASN.1 BER see 5.5.10.3		
		BER-Coding: Context specific, primitive, Tag-Number = 0 ('00')		
LEN	LEN coding see 5.5.10.3			
	one or two bytes coding LEN in the range of: 1 <= LEN <= 255			
	'01' ... '7F'	1 <= LEN <= 127	One byte coding	
	'81'	'80' ... 'FF'	128 <= LEN <= 255	Two byte coding
VALUE	ICC Status Information			
	1 ..., 255 status bytes - each representing one cc or contact unit (slot)			
	Coding of ICC Status Byte see table below			
	ICCSB1	ICCSB2	...	ICCSB<n> '01' <= n <= 'FF'
	ICC Status Byte 1	ICC Status Byte 2	...	ICC Status Byte <n>

Data		Len bytes	Description			
ICCSB<n>	ICC Status Byte	1	ICC Status Value			
			One byte coding			
			b1	'0'	No CC inserted	
				'1'	CC (or some Card) inserted	
			b2	'1'	CC electrically not connected	
			b3	'1'	CC electrically connected / powered	
			b4	'1'	CC is in negotiable mode.	
			b5	'1'	CC is in specific mode.	
			b6	0	other values RFU	
			b7	0	other values RFU	
			b8	'1'	unknown CC state or general contact unit (slot) error	
			b4...b5	'0' '0'	no information given on the cc operation mode.	
			b2...b3	'0' '0'	no information given on the electrical interface	
			b8 ... b1	'00'	CC absent	No CC (Card) present in the addressed contact unit (slot).
				'01'	CC present	A CC is present in the addressed contact unit (slot), but that it has not been moved into position for use. The CC is not powered.
				'03'	CC swallowed	A CC in the contact unit (slot) in position for use, but the CC is not powered.
				'05'	CC powered	A CC in the contact unit (slot) in position for use, and the CC is powered.
				'0D'	CC negotiable	CC has been reset and is awaiting PTS negotiation.
				'15'	CC specific	CC has been reset and specific communication protocols have been established.
				'80'	CC unknown	The cardterminal is unaware of the current state of the contact unit (slot).

5.5.10.8 Functional Unit Data Object

Das Functional Unit Data Object (FU DO) muss im SICCT – Mode unterstützt werden und muss ein Objekt darstellen, um die Funktionalität und Präsenz von Funktionseinheiten des Kartenterminals mittels eines einzigen Datenobjekts abfragen zu können.

Der Aufbau des Objekts richtet sich nach den zurückgemeldeten Funktionseinheiten und variiert mit der technischen Ausprägung des SICCT-Terminals.

Functional Unit Data Object (FU DO)						
TAG	'81'	One byte tag according to MCT-Specifications: Functional Unit Data Object				
		Tag coding according to ASN.1 BER see 5.5.10.3				
		BER-Coding: Context specific, primitive, Tag-Number = 1 ('01')				
LEN	LEN coding see 5.5.10.3					
	one or two bytes coding - LEN in the range of: 0 <= LEN <= 512					
	'00' ... '7F'		0 <= LEN <= 127	One byte coding		
	'81'	'80' ... 'FF'	128 <= LEN <= 255	Two byte coding		
	'82'	'0100' ... '0200'	256 <= LEN <= 512	Three byte coding		
VALUE	Coding, when operating in BCS Mode					
	List of existent and addressable Functional Units					
	BCS-Mode - Direct Coding - At maximum 256 Functional Units per one SICCT-Terminal					
	The list is given by a set of one byte values according to the Functional Unit Index value - see 5.5.9.4					
	ICC Slots		[1 st Display]	[1 st Keypad]	[1 st Printer port]	[1 st Fingerprint] [...]
	'01'	[...]	['0E']	[40]	[50]	[60] [70] [...]
	Coding, when operating in SICCT Mode					
	Referenced Coding - At maximum 256 Functional Units per one SICCT-Terminal					
	The list is given by a set of two byte values according to the Functional Unit Index value - see 5.5.9.4					
	ICC Slots		CL-IF	Displays	Keypads	... Vendor specific FUs
	'0001' [... '00FF']		['1001'...'10FF']	['4000'...'40FF']	['5000'...'50FF']	... '[A000'...'F0FE']
	Note: The presence of FU = '0000' will not be indicated.					

5.5.10.9 Functional Unit Index Data Object

Das Functional Unit Index Data Object (FUI DO) darf im SICCT – Mode unterstützt werden und muss ein TLV-Objekt zur Kodierung von Functional Units in Referenced Coding Darstellung darstellen. Der Wert des Objekts muss einen Ein-Byte-Typqualifizierer der Functional Unit angeben. Nachfolgend muss der korrespondierende Indexwert erscheinen, den das SICCT-Terminal in der Ressourcentabelle zusammen mit einem Friendly Name verwalten muss.

Functional Unit Index Data Object (FUI DO)			
TAG	'84'	One byte tag according to SICCT-Specifications: Functional Unit Index Data Object	
		Tag coding according to ASN.1 BER see 5.5.10.3	
		BER-Coding: Context specific, primitive, Tag-Number = 4 ('04')	
LEN	LEN coding see 5.5.10.3		
	one byte coding - LEN in the range of: LEN = 2		
	'02'	LEN = 2	One byte coding
VALUE	Type and Index of available Functional Units		
	Functional Unit Type Qualifier (first Byte)		
	'00'... 'A0'	see 5.5.9.5	
	Functional Unit Index Number (second Byte)		
	'00' ... 'FF'	see 5.5.9.5	

5.5.10.10 Functional Unit Context Data Object

Das FU Context Data Object (FU CON DO) darf im SICCT – Mode unterstützt werden und muss dazu dienen, einen Ausführungskontext aus verschiedenen Functional Units für die Ausführung bestimmter SICCT-Kommandos definieren zu können.

Das FU Context Data Object muss ein zusammengesetztes TLV-Datenobjekt darstellen, welches minimal zwei und maximal drei Functional Unit Index Data Objects (FUI DO) enthalten darf. Die Anzahl, Kombination und Zuordnung der Functional Units ist kommandospezifisch und bestimmt, welche Terminalressourcen an der Ausführung des entsprechenden Kommandos zu beteiligen sind.

Für ein Terminal mit Display- und Keypad-Optionen kann ein Ausführungskontext bestehen aus:

- einem Karteninterface: ICC / CL-IF (mit PICC, RFID Token)
- einem Ausgabegerät: ein Display
- einem Eingabegerät: ein Keypad oder eine biometrische Sensoreinheit.

Functional Unit Context Data Object (FU CON DO)			
TAG	'A2'	One byte tag according to SICCT-Specifications: Functional Unit Context Data Object	
		Tag coding according to ASN.1 BER see 5.5.10.3	
		BER-Coding: Context specific, constructed, Tag-Number = '02'	
LEN	LEN coding see 5.5.10.3		
	one byte coding - LEN in the range of: 6 <= LEN <= 12		
	'08' ... '0C'	8 <= LEN <= 12	One byte coding
VALUE	Functional Unit Index Data Objects (FUI DO)		
	'84'	L = '02'	Functional Unit Index Value
	:	:	up to three Functional Unit Index Value Data Objects
	['84']	[L = '02']	[Functional Unit Index Value]

Das FU Context Data Object (FU CON DO) darf nur in SICCT-Kommandos, die einen Ausführungskontext aus mehreren Functional Units benötigen, und nur in Referenced Coding Darstellung und nur anstelle eines FUI Datenobjekts verwendet werden.

5.5.10.11 Functional Unit Name Data Object

Das Functional Unit Name Data Object (FU Name DO) darf im SICCT – Mode unterstützt werden und muss ein TLV-Objekt zur Kodierung des 'Friendly Names' einer FU darstellen. Der 'Friendly Name' darf maximal 32 (darstellbare) ASCII-Zeichen umfassen, und muss entsprechend einem im SICCT-Kommando angegebenen Index (FUI DO) zugeordnet und in der Ressourcentabelle des SICCT-Terminals verwaltet werden.

Functional Unit Name Data Object (FU NAME DO)				
TAG	'A1'	One byte tag according to SICCT-Specifications: Functional Unit Name Data Object		
		Tag coding according to ASN.1 BER see 5.5.10.3		
		BER-Coding: Context specific, constructed, Tag-Number = '01'		
LEN	LEN coding see 5.5.10.3			
	One byte coding - LEN in the range of: 0 <= LEN <= 34			
	'00' ... '22'	0 <= LEN <= 34	One byte coding	
VALUE	Friendly Name			
	'13'	'00' <= L <= '20'	Printable String, ASN.1 Coding	
		0 <= L <= 32	Friendly Name: Up to 32 characters	
			see 5.5.9.2	
			Note: LEN == '00' or '02' means an empty Friendly Name (with Tag == '13' and L='00') in order to delete / erase a Friendly Name.	

5.5.10.12 CL-IF Status Data Object

Das CL-IF Status Data Object (CLIF DO) darf im SICCT – Mode unterstützt werden und stellt eine optionale Erweiterung dar, um den Betriebsstatus aller vorhandenen CL-IF vom Kartenterminal mittels eines einzigen Datenobjekts abfragen zu können. Wie das ICC Status Datenobjekt muss das Kommando den Status aller vorhandener CL-IF zurückliefern.

CL-IF Status Data Object (CLIF DO)				
TAG	'83'	One byte tag according to SICCT-Specifications: CLIF Status Data Object		
		Tag coding according to ASN.1 BER see 5.5.10.3		
		BER-Coding: Context specific, primitive, Tag-Number = 3 ('03')		
LEN	LEN coding see 5.5.10.3			
	One or two bytes coding - LEN in the range of: 0 <= LEN <= 255			
	'00' ... '7F'		0 <= LEN <= 127	One byte coding
	'81'	'80' ...'FF'	128 <= LEN <= 255	Two byte coding
VALUE	CLIF Status for max. 14 CLIF			
	Coding of CLIF Status Byte see table below			
	CLIF_SB1	CLIF_SB2	...	CLIF_SB<n> '01' <= n <= 'FF'
	CLIF Status 1	CLIF Status 2	...	CLIF Status <n>

CLIF Status Byte Coding			
CLIF Status Byte	b8...b5	CL-IF Type Information	
		'0'	Unknown RFID token (PICC) / type
		'1'	PICC: ISO/IEC 14443 Type A
		'2'	PICC: ISO/IEC 14443 Type B
		others	RFU
	b4...b1	RFID Token (PICC) Presence	
		'0'	Not present
		'1'	Present, activated, but not selected
		'2'	Present, activated, selected but waiting
		'3'	Present, active
		Others	RFU

5.5.10.13 Reduced / Card Terminal Status Data Object Objects

Das Cardterminal Status Data Object (CTS DO) darf im SICCT – Mode unterstützt werden und stellt eine optionale Erweiterung dar, um Parameter des Kartenterminals mittels eines einzigen Datenobjekts abfragen zu können. Es muss sich um ein 'constructed' ASN.1-Datenobjekt handeln, welches weitere ASN.1-Datenobjekte aufnehmen kann.

Wird das optionale CTS DO unterstützt, müssen beide Darstellungsformen, d.h. die komplette (complete DO) sowie die reduzierte (reduced DO) Darstellungsform, unterstützt werden.

Reduced / Card Terminal Status Data Object (CTS DO)						
TAG	'63'	One byte tag according to SICCT-Specifications:				
		Tag coding according to ASN.1 BER see 5.5.10.3				
		BER-Coding: Application context, constructed, Tag-Number = 3 ('03')				
LEN	LEN coding see 5.5.10.3					
	One, two or three bytes coding - LEN in the range of: 0 <= LEN <= 65525					
	'00' ... '7F'		0 <= LEN <= 127	One byte coding		
	'81'	'80' ... 'FF'		128 <= LEN <= 255	Two byte coding	
	'82'	'0100' ... 'FFF5'		256 <= LEN <= 65525	Three byte coding	
VALUE	Cardterminal Status Information					
	Contained in complete DO	Contained in Reduced DO				
	✓	-	Functional Unit Data Object		see 5.5.10.8	
	✓	✓	ICC Status Data Object		see 5.5.10.7	
	[✓]	[✓]	optional - if present: CL-IF Status Data Object		see 5.5.10.12	
	✓	[✓]	Maximum Length of SICCT Commands – and Response APDUs ²			
			'04'	'01'	'xx'	ASN.1 BER Coding
				'02'	'yyxx'	OCTETSTRING (UNIVERSAL 4): <HB> <LB>

² The maximum size is vendor specific.

5.5.10.14 ICC Interface Status Data Object

Das ICC Interface Status Data Object (INTFS DO) muss im SICCT – Mode unterstützt werden und muss eine Erweiterung darstellen, um Kommunikationsparameter einer Chipkartenfunktionseinheit vom Kartenterminal mittels eines Datenobjekts abfragen bzw. setzen zu können. Es muss sich um ein 'constructed' ASN.1-Datenobjekt handeln, welches weitere ASN.1-Datenobjekte aufnehmen kann.

ICC Interface Status Data Object (INTFS DO)										
TAG	'65'		One byte tag according to SICCT-Specifications:							
			Tag coding according to ASN.1 BER see 5.5.10.3							
			BER-Coding: Application context, constructed, Tag-Number = 5 ('05')							
LEN	LEN coding see 5.5.10.3									
	one or two bytes coding - LEN in the range of: 0 <= LEN <= 255									
	'00' ... '7F'		0 <= LEN <= 127			One byte coding				
	'81'	'80' ...'FF'		128 <= LEN <= 255			Two byte coding			
VALUE	Interface Device Protocol Options									
	Tag	LEN max.	Remark: Object construction according to PCSC V2.0x, but different tag values and value definitions, because value coding might differ to PCSC 2.0x definitions. It shall be possible to map these values to the values of the PCSC framework. Note in difference to PCSC and since SICCT > V1.10: All ASN.1-BER coded Data Objects value field coded as big endian.							
	'80'	'04'	Current Protocol Type							
			ICC	Contact bound Chipcard Protocol Setting						
				b1	T=0	Asynchronous T=0 Protocol		According to ISO 7816-3		
				b2	T=1	Asynchronous T=1 Protocol				
				b3	S=8	SDAP	I ² C Bus Memory Chipcard	One Byte Address Mode		
				b4	S=8	SDAP		Two Byte Address Mode		
				b5	S=9	3WBP	3 Wire Bus Memory Chipcard	According to ISO 7810		
				b6	S=10	2WBP	2 Wire Bus Memory Chipcard			
				b7	S=11	FCBP	Function Control Bus Memory Chip-card			
			b8	'0'	Contact bound					
			PICC	Contactless Chipcard Protocol Setting						
				b1...b3		RFU				
				b4	T=CL	Proximity Smart Card (PICC, RFID Token)		According to ISO 14443-4		
				b5 ... b7		RFU				
				b8	'1'	Contactless				
			'81'	'04'	ICC	Current Clock				
						Current ICC CLK frequency in KHz encoded as big endian integer value. Example: 3.58 MHz is encoded as the integer value 3580.				
	PICC	Current RF frequency								
		Current RF frequency in kHz encoded as big endian integer value.								

	'82'	'04'	ICC	Current F (Clock Conversion Factor)		
				F encoded as big endian integer as read from the card's ATR (TA1). Note: The value may be modified by a subsequent PPS procedure.		
	'83'	'04'	ICC	Current D (Bit Rate Conversion Factor)		
				D encoded as big endian integer as read from the card's ATR (TA1). Note: The value may be modified by a subsequent PPS procedure.		
			PICC	Current D (Bit Rate Conversion Factor) D consists of DS and DR as follows.		
				- DS is encoded by the most significant 2 bytes as a big endian integer, which is bit rate factor for the direction PICC to PCD. - DR is encoded by the least significant 2 bytes as a big endian integer, which is bit rate factor for the direction IFD to PICC		
	'84'	'04'	ICC	Current N (Guard Time Factor)		
				N encoded as a big endian integer.		
	'85'	'04'	ICC	Current W (Work Waiting Time)		
				only T=0 Protocol	Current W	
	'86'	'04'	ICC	Current IFSC (Information Field Size Card)		
				only T=1 Protocol	Current IFSC	
			PICC	Current FSC (maximum Frame Size which the PICC is able to receive)		
				FSC encoded as a big endian integer, if it applies.		
	'87'	'04'	ICC	Current IFSD (Information Filed Size Device)		
				only T=1 Protocol	Current IFSD IFSD encoded as a big endian integer.	
			PICC	Current FSD (maximum frame size which the IFD is able to receive)		
				FSD encoded as a big endian integer, if it applies.		
	'88'	'04'	ICC	Current BWT (Block Waiting Time)		
				only T=1 Protocol	Current BWT BWT encoded as a big endian integer.	
PICC			Current FWT (Frame Waiting Time)			
			FWT encoded as a big endian integer, if it applies.			
'89'	'04'	ICC	Current CWT (Character Waiting Time)			
			only T=1 Protocol	Current CWT CWT encoded as a big endian integer.		
'8A'	'04'	ICC	Current EDC (Error Detection Code)			
			only T=1 Protocol	'00'	LRC	
				'01'	CRC	

5.5.10.15 ICC Interface Capabilities Data Object

Das ICC Interface Capabilities Data Objects (INTFC DO) muss im SICCT – Mode unterstützt werden. Mittels des INTFC DO müssen Eigenschaften einer physikalischen Chipkartenschnittstelle dargestellt werden. Es muss sich um ein 'constructed' ASN.1-Datenobjekt handeln, welches weitere ASN.1-Datenobjekte aufnehmen kann.

ICC Interface Capabilities Data Object (INTFC DO)						
TAG	'66'		One byte tag according to SICCT-Specifications:			
			Tag coding according to ASN.1 BER see 5.5.10.3			
			BER-Coding: Application context, constructed, Tag-Number = 6 ('06')			
LEN	LEN coding see 5.5.10.3					
	one or two bytes coding - LEN in the range of: 0 <= LEN <= 255					
	'00' ... '7F'		0 <= LEN <= 127		One byte coding	
	'81'	'80' ...'FF'	128 <= LEN <= 255		Two byte coding	
VALUE	Interface Device Protocol Options					
	Tag	LEN				
	'80'	'01'	Supported Card Size			
			b1	1	ID-000	
			b2	1	ID-1	
			b3...b8		RFU	
			'00'		unknown / contactless	
	'81'	'01'	Supported / available Mechanical Attributes			
			b1	1	swallowing mechanism	
			b2	1	ejection mechanism	
			b3	1	capture mechanism	
			b4	1	internal slot	
			b5	1	qualifier- mechanic / automatic feature	
			b6 ...b8		RFU	
			'00'		no special characteristics	
	'82'	'01'	Supported / available Card Contacts			
			bn	1	C<n> present / available	
			'00'		unknown / contactless	
	'83'	'01'	Supported Card Types			
			b1	1	ISO 7816-Asynchr.	
			b2	1	ISO 7816-Synchr (Type1)	
			b3	1	ISO 7816-Synchr (Type2)	
			b4	1	ISO 14443-A	
			b5	1	ISO 14443-B	
			b6	1	ISO 15693	
			b7	1	ISO 7816-12	
			b8	1	EMV Level 1	
			'00'		unknown	
	'84'	'01'	Supported Asynchr. Protocols			

			b1	1	T=0	
			b2	1	T=1	
			b3...b6		RFU	
			b7	1	qualifier - ISO 7816 Compatible	
			b8	1	qualifier - EMV Level 1 Compatible	
			'00'		unknown / none	
	'85'	'01'	Supported Synchr. ISO 7816-10 Protocols			
			b1	1	S=8	SDAP (IIC Bus), one Byte Address
			b2	1		SDAP (IIC Bus), two Byte Address
			b3	1	S=9	3 WB (Type 1)
			b4	1	S=10	2 WB (Type 1)
			b5	1	S=11	FCB (Type2)
			b6		RFU	
			b7	1	qualifier - ISO 7816 Compatible	
			b8	1	qualifier – supported in EMV Level 1 Mode	
			'00'		unknown / none	
	'86'	'01'	Supported Wireless ISO 14443-Protocols			
			b1	1	Type A	
			b2	1	Type B	
			b3	1	supported according to ISO 14443-1	
			b4	1	supported according to ISO 14443-2	
			b5	1	supported according to ISO 14443-3	
			b6	1	supported according to ISO 14443-4 (T=CL)	
			b7	1	qualifier – supported in ISO 7816 Mode	
			b8	1	qualifier – supported in EMV Level 1 Mode	
			'00'		unknown / none	
	'87'	'02'	Extended Length Indication			
			'20' ... 'FFF5'		32 ... 65525 Bytes (Maximum value: maximum value of CTS DO - 10 Byte for SICCT-Envelope)	

Für einen Client ergibt sich der Maximalwert einer längstmöglichen APDU (ohne den 10 Byte langen SICCT-Envelope) zu einer konkreten ICC/PICC aus dem kleineren Wert aus INTFC DO Tag '87' und CTS DO (Maximum Length of SICCT Commands – and Response APDUs).

Diese Werte sind insofern statisch, als dass sie sich nicht über eine Gerätekonfiguration am Kartenterminal, sondern nur über eine neue Firmwareversion anpassen lassen. Für einen Client bedeutet dies, dass diese Kartenterminal-individuellen Parameter nicht vor jeder Kartenkommunikation abgefragt werden müssen.

Für die Kommunikation zu einer konkreten gesteckten Karte sind für einen Client neben den Kartenterminal-individuellen Parametern insbesondere auch die Parameter der Karte zu berücksichtigen.

5.5.10.16 Display Capabilities Data Object

Das Display Capabilities Data Objects (DSPLC DO) darf im SICCT – Mode unterstützt werden. Mittels des DSPLC DO müssen die Eigenschaften einer physikalischen Anzeigeeinheit dargestellt werden. Es muss sich um ein 'constructed' ASN.1-Datenobjekt handeln, welches weitere ASN.1-Datenobjekte aufnehmen kann.

Display Capabilities Data Object (DSPLC DO)						
TAG	'67'		One byte tag according to SICCT-Specifications:			
			Tag coding according to ASN.1 BER see 5.5.10.3			
			BER-Coding: Application context, constructed, Tag-Number = 7 ('07')			
LEN	LEN coding see 5.5.10.3					
	one or two bytes coding - LEN in the range of: 0 <= LEN <= 255					
	'00' ... '7F'		0 <= LEN <= 127	One byte coding		
	'81'	'80' ... 'FF'	128 <= LEN <= 255	Two byte coding		
VALUE	Interface Device Protocol Options					
	Tag	LEN				
	'80'	'01'	Number of visible characters per line			
			b8...b1	Number of visible characters per line: '10' <= m <= 'FF'		
			'00'	none		
	'81'	'01'	Number of visible lines			
			b8 ...b1	Number of visible lines: '02' <= n <= 'FF'		
			'00'	none		
	'82'	'01'	Number of virtual characters per line (panning)			
			b8...b1	Number of virtual characters per line (panning): '10' <= p <= 'FF'		
			'00'	none		
	'83'	'01'	Number of virtual lines (scrolling):			
			b8...b1	Number of virtual lines (scrolling): '02' <= s <= 'FF'		
			'00'	None		
	'84'	'01'	Acoustical Indicator(Beeper): 0 ... 1			
			b1	1	Acoustical Indicator(Beeper) present: '0' <= t <= '1'	
			b2...b4		RFU	
			b5	1	Optical Indicator (LED) present	
			b6...b8		Number of LEDs: '0' <= u <= '7'	
			'00'		None	
	One or sequence of Character Set Data Objects					
	'85'	'01'	Character Set Data Object			
			b8...b1	1 st supported Character Set	see 5.5.10.19	
	'85'	'01'	Character Set Data Object			
			b8...b1	2 nd supported Character Set	see 5.5.10.19	
	:	:	:	:	:	
	'85'	'01'	Character Set Data Object			
			b8...b1	<n ^{nt} > supported Character Set	see 5.5.10.19	

Beispiel: SICCT GET STATUS an FU(Dispaly) zur Abfrage des DSPLC DO

```
CAPDU    = '80 13 40 67 00'
RAPDU    = [DSPLC DO] || [SW1SW2]
RAPDU    = '67 18      DSPLC DO
            80 01 17  number of visible characters per line = 23
            81 01 06  number of visible lines = 6
            82 01 17  number of virtual characters per line (panning) = 23
            83 01 06  number of virtual lines (scrolling) = 6
            84 01 91  acoustical indicator: 4 LED and a beeper
            85 01 00  display supports ISO 646 character set
            85 01 30  display supports ISO 18004 QR
            85 01 40  display supports ISO 16022 DataMatrix
9000'     SW1SW2
```

5.5.10.17 PPS - Request / Response Data Object

Das PPS Request / Response Data Object (PPSR DO) darf im SICCT – Mode unterstützt werden und muss eine Protocol Parameter -Select-Message (PPS, nach ISO 7816-3 [STD6]) der Applikationsschicht kodieren, welche das Kartenterminal während der Abarbeitung eines SICCT RESET ICC Kommandos unverändert an eine adressierte Chipkarte und unmittelbar nach dem Reset dieser Chipkarte sendet, sofern die Chipkarte die Bereitschaft zur Negotiation der Kommunikationsparameter im Answer-To-Reset signalisiert hatte.

Die Antwort der Chipkarte (PPS-Response) muss vom Kartenterminal in demselben Format als PPS-Response Data Object zurückgegeben werden, wenn die PPS-Kommunikation mit der Chipkarte stattgefunden hat. Die PPS-Response signalisiert der Applikationsschicht die aktuellen Kommunikationsparameter nach ISO7816-3 [STD6].

PPS - Request / Response Data Object (PPSR DO)								
TAG	'53'	One byte tag according to ISO 7816-6: Discretionary Data Objects						
		Tag coding according to ASN.1 BER see 5.5.10.3						
		BER-Coding: Application context, primitive, Tag-Number = 19 ('13')						
LEN	LEN coding see 5.5.10.3							
	one byte coding - LEN in the range of: 0 <= LEN <= 127							
	'00' ... '7F'	0 <= LEN <= 127			One byte coding			
VALUE	Protocol-Parameter TPDU: PPS Request according to ISO 7816-3							
	PPPS	PPS0		PPPS1	PPPS2	PPS3	PCK	
	man	Man		opt	opt	opt	man	
	Initial Character	Format Character		Parameter Character 1	Parameter Character 2	Parameter Character 3	Check Character	
	Start of PPS Request	mainly codes the protocol		codes the clock and baud rate adjustment ³ factors		RFU, typically does not exist.	RFU typically does not exist.	XOR-Check sum inclusive PPPS to last Parameter Character (typically PPS1).
	'FF'	b8 = 0	RFU	b8...b5	FI			
		b5...b7	indicates presence of PPS1 ... PPS3					
		b4...b1	proposes ICC protocol	b4...b1	DI			

³ According to ISO7816-3: PPS1 codes the same as the TA1-Interface Character of an ICC ATR.

Discrete Protocol-Parameter Values for: PPS Request / Response TPDU according to [ISO/IEC 7816-3]							
PPPS	b8 ... b1	'FF'	PPS request or response				
PPPS0	b8	1	RFU				
		0	default				
	b7.	0	PPS3 does not exist				
		1	PPS3 exists				
	b6	0	PPS2 does not exist				
		1	PPS2 exists				
	b5	0	PPS1 does not exist				
		1	PPS1 exists				
	b4 ... b1	ICC Protocol					
		'0'	T=0	half-duplex transmission of characters			
		'1'	T=1	half-duplex transmission of blocks			
		:		other protocols optional supported			
	'F'	T=15					
	Commonly used values						
	b8 ... b1	'00'	PPS1, PPS2 and PPS3 do not exist				
			T=0 protocol				
		'01'	PPS1, PPS2 and PPS3 do not exist				
			T=1 protocol				
		'10'	PPS1 exists, PPS2 and PPS3 do not exist				
			T=0 protocol				
'11'	PPS1 exists, PPS2 and PPS3 do not exist						
	T=1 protocol						
PPPS1	PPS1 allows the interface device to propose values of F and D to the card. Encoded in the same way as in TA1 of the CC ATR.						
	b8 ... b5	reference to a clock rate conversion factor					
		FI		Fi	f (max) MHz		
		0000	'0'	372	4	Default frequency during Reset Phase	
		0001	'1'	372	5	Default frequency after Reset Phase	
		0010	'2'	558	6	The support of frequencies higher than 5 MHz is optional and vendor dependent.	
		0011	'3'	744	8		
		0100	'4'	1116	12		
		0101	'5'	1488	16		
		0110	'6'	1860	20		
		0111	'7'	RFU			
		1000	'8'				
		1001	'9'	512	5	The support of frequencies higher than 5 MHz is optional and vendor dependent.	
		1010	'A'	768	7,5		
		1011	'B'	1024	10		
		1100	'C'	1536	15		
		1101	'D'	2048	20		
		1110	'E'	RFU			
		1111	'F'				
		b4 ... b1	reference to a baud rate adjustment factor				
			DI		Di		
			0000	'0'	RFU		
			0001	'1'	1		
			0010	'2'	2		
			0011	'3'	4		
			0100	'4'	8		
			0101	'5'	16		
			0110	'6'	32		
			0111	'7'	64		
	1000		'8'	12			
	1001		'9'	20			
	1010		'A'	RFU			
	1011		'B'				
	1100		'C'				
1101	'D'						
1110	'E'						

		1111	'F'	
PPPS2	b8 ... b1	RFU	RFU	
PPPS3	b8 ... b1	RFU	RFU	
PCK	XOR-Checksum			
	b8 ... b1		Exclusive-oring all the bytes PPSS to PCK inclusive shall give '00'. Any other value is invalid.	

5.5.10.18 Application Label Data Object

Das Application Label Data Object (APPL DO) muss im SICCT – Mode unterstützt werden und muss eine ASCII-Kodierte Display-Message aufnehmen, die das Kartenterminal mit Anzeigeeinheit dem Anwender zur Benutzerführung darstellen muss. Aufgrund der Kompatibilität zum BCS-Modus, wird eine implizite ASCII-Kodierung der Display-Messages vorausgesetzt.

Vor jeder neuen Ausgabe einer Display-Message muss das Terminal das Display löschen. Ein leeres Datenobjekt bzw. eine leere Message muss das Löschen des Displays am Kartenterminal bewirken.

Das APPL DO kann in einigen SICCT-Befehlskontexten alternativ zum SICCT Message-To-be-Displayed (SMTBD) DO, welches neben einer Display-Message explizit auch den Zeichensatz angibt, verwendet werden. Innerhalb eines SICCT Befehls darf entweder die eine oder andere Objektvariante (APPL DO oder SMTBD DO) aber es dürfen nicht beide Formen zugleich zum Einsatz kommen.

Application Label Data Object (APPL DO)			
TAG	'50'	One byte tag according to ISO 7816-6: Application Label	
		Tag coding according to ASN.1 BER see 5.5.10.3	
		BER-Coding: Application context, primitive, Tag-Number = 16 ('10')	
LEN	LEN coding see 5.5.10.3		
	one byte coding - LEN in the range of: 0 <= LEN <= 127		
	'00' ... '7F'	0 <= LEN <= 127	One byte coding
		The actual supported length depends on the capabilities of the selected FU of type 'display' . At minimum 32 ('20') characters according to a two line display with 16 characters each.	
VALUE	Application Label (Display Message)		
	Sequence of ASCII characters.		

Der Display-Text im APPL DO muss in ASCII (ISO 646 bzw. nationale Referenz-Version) kodiert werden. Sofern das SICCT über eine adressierbare Anzeigeeinheit (Display) verfügt, muss der Zeichensatz die folgenden Zeichen umfassen.

Eine generelle Beschreibung von Vorgaben zum Aufbau und zur Behandlung von Display-Nachrichten findet sich in 5.6.1.

Coding of Display Messages					
Minimal supported ASCII Character Set for Display Messages					
Buchstaben	Characters	'a' ... 'z'	'61'...'7A'	Small characters	
		'A' ... 'Z'	'41' ... '5A'	Capital characters	
Ziffern	Digits	'0' ... '9'	'30' ... '39'	Numbers	
Sonstige und Sonder- zeichen	miscellane- ous and special characters	' '	'20'	Leerzeichen	Space
		'+'	'2B'	Pluszeichen	plus sign
		','	'2C'	Komma	comma
		'_'	'2D'	Minuszeichen	minus sign
		'.'	'2E'	Punkt	dot
		':'	'3A'	Doppelpunkt	colon
		'='	'3D'	Gleichheitszeichen	equal sign
		'?'	'3F'	Fragezeichen	question mark
		'Ä'	'5B'	Große Umlaute	
		'Ö'	'5C'		
		'Ü'	'5D'		
		'ä'	'7B'	Kleine Umlaute	
		'ö'	'7C'		
		'ü'	'7D'		
		'ß'	'7E'		
Kontroll- und Steuerzeichen	Control characters	BEL	'07'	Bell (Beeper)	The BEL-character is detected by the SICCT -Terminal and executed as a short acoustic sound. BEL is not replaced by other characters as SPACE. The length of the sound is vendor dependent. In case no beeper exists, the BEL character is ignored.
		CR	'0D'	Carriage Return	All sequences of Carriage Return and / or Line Feet are detected by the SICCT -Terminal and executed in the same way to position the text cursor on the display to the beginning of the next line.
		LF	'0A'	Line Feed	
		CR LF	'0D' '0A'	Carriage Return Line Feed	
		LF CR	'0A' '0D'	Line Feed Carriage Return	
Trennzeichen	Delimiter characters	VT	'0B'	Predetermined (Text) Breaking Point Character	All messages (in SICCT Mode) may contain breaking characters '0B' to permit to break the text message into defined pieces of text before displaying it. This character shall be suppressed and shall not be counted as message character. The text breaking mechanism is vendor dependent, and the terminal might filter out and ignore such characters.
		SI	'0F'	(Message Field) Separator Character	Special Display -Messages for SICCT-Commands which perform an input (data entry) can be structured into two fields (a message and a PIN-Prompt field). This character shall be suppressed and shall not be counted as message character.
Unbekannte Zeichen	Unknown characters	Unknown / unsupported characters are detected by the SICCT-Terminal and are replaced by a SPACE (0x20) character each.			
Annotation	BCS Mode: In BCS Mode the character set and control characters are defined according to the MCT specification [MKT_10].				

5.5.10.19 Character Set Data Object

Das Character Set Data Object (CS DO) darf im SICCT – Mode unterstützt werden und muss einen Zeichensatz kodieren, den das Kartenterminal für eine Display-Message verwenden muss, sofern das Terminal den entsprechenden Zeichensatz unterstützt.

Eine generelle Beschreibung von Vorgaben zum Aufbau und zur Behandlung von Display-Nachrichten findet sich in 5.6.1.

Character Set Data Object (CS DO)			
TAG	'85'	One byte tag according to SICCT-Specifications: Character Set	
		Tag coding according to ASN.1 BER see 5.5.10.3	
		BER-Coding: Context specific, primitive, Tag-Number = 5 ('05')	
LEN	LEN coding see 5.5.10.3		
	one byte coding - LEN in the range of: 0 <= LEN <= '01'		
	'01'	1	one byte coding
VALUE	Character Set Index Value		
	Coding see table below		

Character Set Index Value	Normative Reference	Description			
'00'	ISO 646	Standard – International 7 bit character set			
		Germany	man	DIN 66003	ISO 646-DE: German Variant of ISO 646
		US	opt	ASCII	ISO 646-US: 7 bit ASCII
		:	opt	:	ISO 646-xx: 7 bit national Character Set
		Other national Character Sets	opt		
'01'	ISO / IEC 8859-1	Latin-1			
		Die Codetabelle dieser Kodierung enthält die schriftspezifischen Zeichen für westeuropäische und amerikanischen Sprachen. Der Zeichenvorrat deckt die Sprachen Albanisch, Dänisch, Deutsch, Englisch, Färöisch, Finnisch, Französisch, Galizisch, Irisch, Isländisch, Italienisch, Katalanisch, Niederländisch, Norwegisch, Portugiesisch, Schwedisch und Spanisch ab. Lediglich einzelne Zeichen wie das niederländische ij, die französischen Ligaturen œ und Œ oder die deutschen Anführungszeichen "" fehlen.			
'02'	ISO / IEC 8859-2	Latin-2			
		Die Codetabelle dieser Kodierung enthält die schriftspezifischen Zeichen für die meisten mitteleuropäischen und slawischen Sprachen. Sie deckt die Sprachen Kroatisch, Polnisch, Rumänisch, Slowakisch, Slowenisch, Tschechisch und Ungarisch ab.			
'03'	ISO / IEC 8859-3	Latin-3			
		Die Codetabelle dieser Kodierung deckt die Sprachen Esperanto, Galizisch, Maltesisch und Türkisch ab.			
'04'	ISO / IEC 8859-4	Latin-4			
		Die Codetabelle dieser Kodierung enthält einige Zeichen der Sprachen Estnisch, Lettisch und Litauisch. Vergleichen Sie diese Kodierung auch mit ISO 8859-10, deren Codetabelle sehr ähnlich ist.			
'05'	ISO / IEC 8859-5	8859-5			
		Die Codetabelle dieser Kodierung enthält kyrillische Zeichen. Sie deckt weitgehend die Sprachen Bulgarisch, Mazedonisch, Russisch, Serbisch und Ukrainisch ab.			
'06'	ISO / IEC 8859-6	8859-6			
		Die Codetabelle dieser Kodierung enthält Zeichen arabischer Schrift. Die Darstellung der Zeichen in der folgenden Tabelle ist jedoch "abstrakt", da die Zeichen in der Schriftpraxis variieren, je nachdem, ob sie am Anfang, in der Mitte oder am Ende eines Wortes oder einzeln stehen. Arabisch zeichnet sich weiterhin dadurch aus, dass die Schriftrichtung von rechts nach links ist.			

'07'	ISO / IEC 8859-7	8859-7
		Die Codetabelle dieser Kodierung enthält die Zeichen der neugriechischen Schrift.
'08'	ISO / IEC 8859-8	8859-8
		Die Codetabelle dieser Kodierung enthält die Zeichen der neugriechischen Schrift.
'09'	ISO / IEC 8859-9	Latin-5
		Diese Kodierung ist speziell für Türkisch gedacht. Die Codetabelle basiert auf ISO 8859-1, enthält jedoch anstelle der isländischen Sonderzeichen türkische Zeichen.
'0A'	ISO / IEC 8859-10	Latin-6
		Die Codetabelle dieser Kodierung enthält speziell Zeichen für die Sprachen Grönländisch (Inuit) und Lap-pisch (Sami).
'0B'	ISO / IEC 8859-11	Thai
		ISO 8859-11 versucht möglichst viele Zeichen der Thai-Schrift abzudecken.
'0C'	not sup-ported	ISO 8859-12 ist kein Teil der Normenfamilie ISO/IEC 8859.
'0D'	ISO / IEC 8859-13	Latin-7 oder Baltisch
		8859-13 versucht möglichst viele Sonderzeichen der baltischen und skandinavischen Sprachen abzude-cken wie auch ISO 8859-4 (Latin-4 eher baltisch) und ISO 8859-10 (<i>Latin-6</i> eher nordisch) denen einige Zeichen fehlten.
'0E'	ISO / IEC 8859-14	Latin-8
		ISO 8859-14 versucht alle Sonderzeichen keltischer und einiger anderer westeuropäischer Sprachen ab-zudecken.
'0F'	ISO / IEC 8859-15	Latin-9
		ISO 8859-15 versucht möglichst viele Sonderzeichenvorwiegend westeuropäischer Sprachen abzudecken und deckt im Gegensatz zu ISO 8859-1 auch Französisch und Finnisch komplett ab und beinhaltet das Eurosymbol.
'10'	ISO / IEC 8859-16	Latin-10
		ISO 8859-16 versucht möglichst viele Sonderzeichen europäischer Sprachen abzudecken, darunter vor allem das südosteuropäische Albanisch, Kroatisch, Ungarisch, Italienisch, Polnisch, Rumänisch und Slo-wenisch, aber auch Finnisch, Französisch, Deutsch und irisches Gälisch (neue Rechtschreibung). Im Ver-gleich zu seinen Geschwistern legt ISO 8859-16 viel mehr Wert auf Buchstaben mit Diakriten und verzichtet dafür auf andere (Satz-)Zeichen.
'20'	[ISO/IEC 10646] UTF-8	UTF-8 according to RFC 3629 is a transformation format of [ISO/IEC 10646]
		UTF-8 (Abk. für 8-bit Unicode Transformation Format) ist die verbreitetste Kodierung für Unicode-Zeichen; dabei wird jedem Unicode-Zeichen eine speziell kodierte Bytekette von variabler Länge zugeordnet. UTF-8 unterstützt bis zu 4 Byte, auf die sich wie bei allen UTF-Formaten alle 1.114.112 Unicode-Zeichen abbil-den lassen.
		Unicode-Zeichen mit den Werten aus dem Bereich von 0 bis 127 (0 bis 7F hexadezimal) werden in der UTF-8-Kodierung als ein Byte mit dem gleichen Wert wiedergegeben. Insofern sind alle Daten, die aus-schließlich echte ASCII-Zeichen verwenden, in beiden Darstellungen identisch.
		Unicode-Zeichen größer als 127 werden in der UTF-8-Kodierung zu Byteketten der Länge zwei bis vier kodiert.
'30'	[ISO/IEC 18004]	Darstellung des übergebenen Textes als QR-Code
'40'	[ISO/IEC 16022]	Darstellung des übergebenen Textes als Data-Matrix-Code

5.5.10.20 SICCT Message-To-Be-Displayed Data Object

Das SICCT Message-To-Be-Displayed Data Object (SMTBD DO) darf im SICCT – Mode unterstützt werden und muss einen Zeichensatz mit einer entsprechend strukturierten Display Message (OCTETSTRING DO) beinhalten. Das Kartenterminal muss die Textnachricht beziehungsweise den daraus berechneten 2D-Code über eine adressierbare Anzeigeeinheit (Display) anzeigen, sofern die Anzeigeeinheit den Zeichensatz unterstützen kann, und der Ausführungskontext eines SICCT-Kommandos die Ausgabe nicht verbietet.

Vor jeder neuen Ausgabe einer Display-Message muss das Terminal die Anzeigeeinheit (Display) löschen.

Ein leeres Datenobjekt bzw. eine leere Message muss das Löschen der Anzeigeeinheit (Display) am Kartenterminal bewirken.

Das SMTBD DO kann in einigen Befehlskontexten alternativ zum Application Label (APPL) DO, welches eine ASCII-Kodierte Display-Message angibt, verwendet werden.

Die Erweiterungen bezüglich der 2D-Codes sind nur im Kontext von SICCT OUTPUT unter Nutzung des SMTBD DO erlaubt.

SICCT Message-To-Be-Displayed Data Object (SMTBD DO)					
TAG	'A0'		One byte tag according to SICCT-Specifications: SICCT Message-To-Be-Displayed		
			Tag coding according to ASN.1 BER see 5.5.10.3		
			BER-Coding: Context specific, constructed, Tag-Number = 0 ('00')		
LEN	LEN coding see 5.5.10.3				
	'00' ... '7F'		0 <= LEN <= 127	One byte coding	
	'81'	'80' ... 'FF'	128 <= LEN <= 255	Two byte coding	
	'82'	'0100' ... 'FFFB'	256 <= LEN <= MAX	Three byte coding	
VALUE	One Character Set DO and corresponding Message (OCTETSTRING DO)				
	Character Set Data Object		see 5.5.10.19		
	'04'	The actual supported length depends on the capabilities of the selected FU of type 'display'. At minimum 32 ('20') characters according to a two line display with 16 characters each.			
		'00' ... '7F'		0 <= LEN <= 127	One byte coding
		'81'	'80 ... 'FF'	128 <= LEN <= 255	Two byte coding
		'82'	'0100' ... 'FFFB'	256 <= LEN <= MAX	Three byte coding
		Byte Sequence formatted according the selected character set giving the message to be displayed.			
MAX ist abhängig von der gesamten SICCT Kommandolänge					

Eine Tabelle mit den zulässigen Zeichen und Steuersequenzen ist in Kapitel 5.5.10.18 (APPL DO) dargestellt. Eine generelle Beschreibung von Vorgaben zum Aufbau und zur Behandlung von Display-Nachrichten findet sich in 5.6.1.

Für die Darstellung von 2D-Codes werden im CS DO die entsprechenden Formate konfiguriert. Deren zu kodierender Inhalt wird im OCTETSTRING DO mit Tag '04' gesetzt. Die maximale Länge des Inhaltes bzw. Größe des 2D-Codes ist abhängig von der Anzeige des Terminals. Der 2D-Code wird im Terminal kodiert und gerendert.

5.5.10.21 Waiting Time Data Object

Das Waiting Time Data Object (WT DO) muss im SICCT – Mode unterstützt werden und kann das maximale Zeitintervall z.B. für notwendige Benutzerinteraktionen am Kartenterminal (in Sekunden) kodieren.

Aktion	Anwendung in Kommando	Referenz
Action	Usage within Command	Reference
max. Zeitdauer bis zum Stecken einer Chipkarte / PICC (RFID Token)	SICCT REQUEST ICC	see 5.13
max. Zeitdauer bis zur Entnahme einer Chipkarte / PICC (RFID Token)	SICCT EJECT ICC	see 5.14
Zeitdauer zur Anzeige einer Textmeldung und bis zur Rückkehr aus dem SICCT-Kommando	SICCT OUTPUT	see 5.18
max. Zeitdauer bis zur ersten Tastenbetätigung /Eingabestart) bei einer Keypad-Eingabe bzw. PIN-/Passwort-Eingabe	SICCT INPUT	see 5.17
max. Zeitdauer zwischen den Tastenbetätigungen (nach Eingabestart)	SICCT PERFORM VERIFICATION	see 5.19
max. Zeitdauer bis zur Betätigung der Bestätigungstaste nach einer erfolgten Keypad-Eingabe bzw. PIN-/Passwort-Eingabe	SICCT MODIFY VERIFICATION DATA	see 5.20
max. Zeitdauer (overall timer) für eine komplette PIN-/Texteingabe (z.B. für eine 12-stellige PIN)		

Waiting Time Data Object (WT DO)			
TAG	'80'	One byte tag according to MCT-Specifications: Waiting Time Data Object	
		Tag coding according to ASN.1 BER see 5.5.10.3	
		BER-Coding: Context specific, primitive, Tag-Number = 0 ('00')	
LEN	LEN coding see 5.5.10.3		
	one byte coding - LEN in the range of: 0 <= LEN <= '01'		
	'01'	1	one byte coding
VALUE	Waiting Time (in seconds): Max. waiting time in seconds, binary coding		
	'00' ... 'FF'	0 ... 255	Binary code Value: Waiting time in seconds
			<i>Note: The special value '00' overrides the timing monitoring within the terminal in a way that the command will be executed, and the terminal will return immediately with the result of the corresponding SICCT command. This matches the same behavior as in case no WTDO has been send with a SICCT command if not specified otherwise.</i>

5.5.10.22 Command-To-Perform Data Object

Das Command-To-Perform Data Object (CMD DO) darf im SICCT – Mode unterstützt werden und muss ein Template kodieren, in dem das Kartenterminal ein Chipkarten-APDU oder Teile eines Chipkarten-APDUs einbetten muss.

Der Inhalt und Aufbau des Value-Feldes kann in Abhängigkeit der Anzahl der erforderlichen PIN-Nummern variieren und muss sich am Anwendungsfall des Datenobjekts im Kontext folgender SICCT-Kommandos orientieren

- SICCT PERFORM VERIFICATION
- SICCT MODIFY VERIFICATION DATA
- Zu beachten ist, dass die zweite (2nd) Insertion Position für die Ausführung des Kommandos SICCT MODIFY VERIFICATION DATA verpflichtend ist. Dieses Byte muss beim SICCT PERFORM VERIFICATION entfallen.
- Zur Initiierung einer sog. NULL-PIN-Abfrage mittels des SICCT MODIFY VERIFICATION DATA Kommandos muss das CMD DO in allen Datenobjekten vollständig besetzt übergeben werden. Die erste 'Insert Position' muss als Indikator den Wert Null ('00') enthalten, nicht benötigte Display-Nachrichten müssen als notwendige Platzhalter in Form von 'leeren' Datenobjekten, bestehend aus Tag und Value, übergeben werden.

Command-To-Perform Data Object (CMD DO)					
TAG	'52'		One byte tag according to MCT-Specifications: Command To Perform		
			Tag coding according to ASN.1 BER see 5.5.10.3		
			BER-Coding: Application context, primitive, Tag-Number = 18 ('12')		
LEN	LEN coding see 5.5.10.3				
	one or two byte coding - LEN in the range of: 0 <= LEN <= 255				
	'00' ... '7F'		0 <= LEN <= 127	One byte coding	
	'81'	'01' ...'FF'	128 <= LEN <= 255	Two byte coding	
VALUE	Command-to-perform				
	Command-to-perform is a byte string containing the concatenation of control byte insertion position(s) CC APDU				
	Control Byte	Control Byte for user authentication			
		'x0'	BCD coded PIN / password with <x> digits or characters		
		'x1'	ASCII coded PIN / password with <x> digits or characters		
		'x2'	Format-2 PIN –Block coded PIN / password with <x> digits or characters		
			SICCT PERFORM VERIFICATION		Note: This command shall ignore b4...b3 within the Control Byte.
			SICCT MODIFY VERIFICATION DATA		Note: This command resumes from b4...b3 that the Control Byte specifies the settings for the 2 nd PIN. In this case the 1 st PIN's settings are of same PIN-type but with variable length.
		'x4'	BCD coded PIN / password with minimum <x> digits or characters for the new (2 nd) PIN.		
		'x5'	ASCII coded PIN / password with minimum <x> digits or characters for the new (2 nd) PIN.		
		'x6'	Format-2 PIN –Block coded PIN / password with minimum <x> digits or characters for the new (2 nd) PIN.		
		'FF'	In case of biometric identification		
	1 st Insertion Position Byte	'06' ... 'FF'	Mandatory Field - Insertion Position for the (1 st) PIN		
			SICCT PERFORM VERIFICATION		Mandatory Field - Insertion position for the PIN
			SICCT MODIFY VERIFICATION DATA		Mandatory Field - Insertion position for the 1 st PIN
			The insertion position for the first PIN digit / character within the CC APDU		
			The insertion position counts from one starting at 'CLA'. Typically the 1 st insertion position is 'typically equal or greater than 06' (after a five byte APDU: CLA, INS, P1, P2, Lc).		

			Null-PIN-Support - Only for usage with SICCT VERIFY VERIFICATION DATA might this value be zero ('00') in order to instruct this SICCT command to skip the 1 st PIN entry phase and process only the second PIN and conformance entry.				
[2 nd Insertion Position Byte]	'06' ... 'FF'	Conditional Field - Insertion Position for the 2 nd PIN					
		SICCT PERFORM VERIFICATION			Note: This field must not be transmitted. for this command.		
		SICCT MODIFY VERIFICATION DATA			Conditional Field - Insertion position for the 2 nd PIN		
					Note: This field has to be transmitted. for this command.		
		The insertion position for the first PIN digit / character within the CC APDU					
The insertion position counts from one starting at 'CLA'. Typically, the 2 nd insertion position is greater than the 1 st insertion position (after a five byte APDU: CLA, INS, P1, P2, Lc).							
Chipcard APDU	APDU to be sent to the Chipcard						
	CLA	INS	P1	P2	Lc	'x' bytes PIN / Password	
	Valid INS-Bytes						
	CLA	INS	Other INS Bytes rejected by the SICCT-Terminal				
	'20'	VERIFY					
		Verification data reference acc. to ISO 7816-4, table 65 [STD8]					
		VERIFY CHV					
		Verification data reference acc. to GSM 11.11					
		Verification data reference acc. to prEN 726-3					
	'24'	CHANGE REF. DATA					
		Change reference data acc. to ISO 7816-4, table 65 [STD8]					
		CHANGE CHV					
		Change Cardholder Verification according to GSM 11.11					
	'26'	Change Cardholder Verification according to prEN 726-3					
		DISABLE CHV					
		Disable Cardholder Verification according to ISO 7816-8					
	'28'	Disable Cardholder Verification according to GSM 11.11					
		Disable Cardholder Verification according to prEN 726-3					
		ENABLE CHV					
	'2A'	Enable Cardholder Verification according to ISO 7816-8					
		Enable Cardholder Verification according to GSM 11.11					
		Enable Cardholder Verification according to prEN 726-3					
		PERFORM SECURITY OPERATION					
		P1 P2 according to ISO 7816-8, 5.9. table 13					
	'2C'	'82'	'80'	PSO (Encipher) operation according to ISO 7816-8 in order to support VERSA-concept (VERSA: "distributed signature workplaces").			
		'84'					
		'86'					
		RESET RETRY COUNTER					
	RESET RETRY COUNTER data acc. to ISO 7816-4, table 65 [STD8]						
	UNBLOCK CHV						
	Verification data reference acc. to GSM 11.11						
	Verification data reference acc. to prEN 726-3						

Control Byte Coding for user authentication				
Control Byte	Coding for PIN / Password			
	b8...b5	Length of PIN to be presented.		
		0000	'0'	Variable length PIN / password
		If length = 0 (value for variable length), then pressing of validation key is required.		
		0100	'4' ...	Number of PIN digits or characters
		...	'12'	Minimal 4 digits / character
		1100		Maximal 12 digits / characters
	b4...b3	0	0	'0' default
		0	1	'1' SICCT PERFORM VERIFICATION This command shall ignore b4...b3 within the Control Byte.
		1	0	'2' ... SICCT MODIFY VERIFICATION DATA: In this case b8...b5 code the minimal length of the new (2 nd) PIN.
		1	1	'3' RFU
	b2...b1	0	0	BCD coded PIN
		0	1	T.50 coding according to International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) - Information technology - 7-bit coded character set for information interchange, see table below
		1	0	characters according to T.50 according T.50 with b8=0 (i.e. digit 0 is coded '30', digit 1 is coded '31' etc.)
		1	1	Format 2 PIN Block according to ISO 9564-1, see table below
	Coding for Biometric identification			
	b8...b1	11111111	'FF'	In case of biometric identification

T.50 coding - technology - 7-bit coded character set for information interchange				
Byte sequence – each byte codes one character / digit within 7 bits				
b8	0	7-bit coded character set for information interchange		
	Supported characters			
b8...b1	0011 0000	'30'	0	according to International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) - Information technology - 7-bit coded character set for information interchange
	0011 0001	'31'	1	
	0011 0010	'32'	2	
	0011 0011	'33'	3	
	0011 0100	'34'	4	
	0011 0101	'35'	5	
	0011 0110	'36'	6	
	0011 0111	'37'	7	
	0011 1000	'38'	8	
	0011 1001	'39'	9	

Format 2 PIN Block according to ISO 9564-1															
8 Byte sequence coding up to 12 digits within 16 nibbles															
Byte 1		Byte 2		Byte 3		Byte 4		Byte 5		Byte 6		Byte 7		Byte 8	
C	L	P	P	P	P	P / F	P / F	P / F	P / F	P / F	P / F	P / F	P / F	F	F
'2'	C = Control field, value '2'														
	Len	L = Length of PIN in BCD													
		P = PIN digit in BCD coding: min 4 up to 12 digits (nibbles)							F = Filler with value 'F': min 2 up to 10 nibbles						

Die Bedingungen, wie Informationen vom Kartenterminal eingefügt werden müssen, hängt vom entsprechenden Kartenkommando (APDU) und von dem PIN-Format ab. Eine PIN muss nach ISO 9564-1 [STD17] minimal 4 und darf maximal 12 Zeichen (digits) aufweisen.

Für die Ergänzung von PIN-Eingaben am Terminal in den Kommandos

- SICCT PERFORM VERIFICATION
- SICCT MODIFY VERIFICATION DATA

müssen folgende Formen des APDU möglich sein

- Command Header (CLA, INS, P1 P2 = 4 Bytes),
falls im Datenfeld des ICC-Kommandos nur der PIN ohne Padding eingetragen wird
- Command Header mit Längenfeld Lc und mit Paddingbytes vorformatiertem Datenfeld.

5.5.10.23 Sequence Number Data Object

Anmerkung: Kommandos und Antworten werden nach dem Command-Response-Modell ausgetauscht. Der Aufrufer muss für jedes Kommando eine Kommandosequenznummer vergeben, die im Übertragungsprotokoll angegeben werden muss, und zu der eine Antwort empfangen werden kann. Zu einem späteren Zeitpunkt kann mittels der Kommandosequenznummer auch der Bearbeitungsstatus eines SICCT-Kommandos abgefragt oder modifiziert werden.

Das Sequence Number Data Object (SEQNO DO) muss im SICCT – Mode unterstützt werden und muss der Übermittlung der Sequenznummer sowie im Kontext von SICCT CONTROL COMMAND unterstützt werden

- zur Abfrage des Bearbeitungsstatus eines zuvor an das SICCT-Terminal gesendeten Kommandos,
- zum Beenden eines zuvor an das SICCT-Terminal gesendeten Kommandos.

Wird das Datenobjekt an das Terminal gesendet, so muss dieses ein einzelnes INTEGER-Datenobjekt enthalten, dessen Value-Feld die eindeutige Kommandosequenznummer (s. 5.5.1 ff) angeben muss.

Innerhalb einer Antwort des Kartenterminals muss das SEQNO DO zusätzlich ein weiteres INTEGER-Datenobjekt enthalten, welches den Bearbeitungsstatus angibt.

Sequence Number Data Object (SEQNO DO)						
TAG	'68'	One byte tag according to SICCT-Specifications:				
		Tag coding according to ASN.1 BER see 5.5.10.3				
		BER-Coding: Application context, constructed, Tag-Number = 8 ('08')				
LEN	LEN coding see 5.5.10.3					
	one byte coding - LEN in the range of: 4 <= LEN <= 8					
	'04' <= LEN <= '08'		LEN = 4 ... 8		One byte coding	
					DO in Request: 4 Bytes	
					DO in Response: 8 Bytes	
	Command Sequence Number				A distinct two byte value	
	'04'	'L = '02'	<HB>	<LB>	0000' <= Sequence Number <= 'FCFF'	
			'00' ... 'FF'	'00' ... 'FF'	Note: 'FD00' <= Sequence Number <= 'FFFF' signal Events sent by the SICCT to the host entity.	
	[Command Sequence Status Word]				Presence – Conditional	
					For requests not. present.	
	['04']	['00' <= L <= '02']	[<HB>]	[<LB>]	For responses: A distinct two byte value set by the CT for the response APDU.	
			'90'	'00'	Sequence Number found: Command queued and not in processing state	
			'62'	'0x'	Sequence Number found: command in processing state '0x'	
			'64'	'0x'	Sequence Number found: processing state found at stage '0x' - state cannot be changed.	
			'6F'	'00'	Sequence Number not found: no information given on processing state	
						ASN.1 Universal 4 Coding

Use Cases for Sequence Number Data Object		
SICCT Command	Description	Reference
SICCT CONTROL COMMAND	Query command execution state for given Sequence No DO	see 5.9
	Terminate command execution for given Sequence No DO	

5.5.10.24 Download Parameter Data Object

Das Download Parameter Data Object (DLPARAM DO) muss im SICCT – Mode unterstützt werden und muss einzelne Parameter zum Handling des Downloads zum Austausch der Gerätefirmware enthalten:

- Timeout
- Maximale Download Data Object Größe

		Download Parameter Data Object (DLPARAM DO)				
TAG	'73'		One byte tag according to ISO 7816-6: Discretionary Data Objects			
			Tag coding according to ASN.1 BER see 5.5.10.3			
			BER-Coding: Application context, constructed, Tag-Number = 19 ('13')			
LEN	LEN coding see 5.5.10.3					
	one byte coding - LEN in the range of: 6 <= LEN <= 8					
	'06' ... '08'		6 <= LEN <= 8		One byte coding	
VALUE	Download Parameter					
	Sequence of TLV-Objects					
	'04'	'01'	'xx'	<LB>	ASN.1 BER Cod- ing OCTETSTRING (UNIVERSAL 4)	Timeout - Waiting Time
		'02'	'yyxx'	<HB> <LB>		Waiting time in seconds how long the device needs to process the last or next download data object.
	'04'	'01'	'xx'	<LB>		Maximum Length of Download Data Package element of Download Data Object as number of bytes.
		'02'	'vvyx'	<HB> <LB>		

5.5.10.25 Download Data Object

Das Download Data Object (DLDATA DO) muss im SICCT – Mode unterstützt werden und muss jeweils ein Download-Datenpaket enthalten.

Download Data Object (DLDATA DO)				
TAG	'73'		One byte tag according to ISO 7816-6: Discretionary Data Objects	
			Tag coding according to ASN.1 BER see 5.5.10.3	
			BER-Coding: Application context, constructed, Tag-Number = 19 ('13')	
LEN	LEN coding see 5.5.10.3			
	one byte coding - LEN in the range of: 0 <= LEN <= 65535			
	'00' ... '7F'		0 <= LEN <= 127	One byte coding
	'81'	'80' ... 'FF'	128 <= LEN <= 255	Two byte coding
	'82'	'0100' ... 'FFFB'	256 <= LEN <= 65531	Three byte coding ⁴
VALUE	Download Data Package			
	Vendor specific Data Objects or Format			

⁴ Siehe SICCT DOWNLOAD DATA: Max. Nc = 65535, d.h. max Länge für DLDATA DO = (65536-4) = 65531

5.5.10.26 Download Termination Data Object

Das Download Termination Data Object (DLTERM DO) muss im SICCT – Mode unterstützt werden und muss einzelne Parameter zum Abschluß des Downloads enthalten. Das DLTERM DO gibt an, wieviel Zeit vom Terminal zum Abschluss des Downloads (ggf. für die interne Verarbeitung) benötigt wird, und unter welchen Bedingungen die Download Operation abgeschlossen wird.

Download Termination Data Object (DLTERM DO)					
TAG	'73'		One byte tag according to ISO 7816-6: Discretionary Data Objects		
			Tag coding according to ASN.1 BER see 5.5.10.3		
			BER-Coding: Application context, constructed, Tag-Number = 19 ('13')		
LEN	LEN coding see 5.5.10.3				
	one byte coding - LEN in the range of: Len = 5 ... 7				
	'05' ... '07'		5 <= LEN <= 7		One byte coding
VALUE	Timeout				
	'04'	'00'	Empty DO		No time indication mandatory in case of Finalisation Condition b3=1 or b4=1
		'01'	'xx'	Timeout coded as	Extended Timeout Timeout in seconds how long the device needs to process last download data object or finish the download process.
				ASN.1 BER Coding OCTETSTRING (UNIVERSAL 4):	
	'02'	'yyxx'	<HB>] <LB>		
	Finalisation Condition				
	'04'	'01'	'xy'	The terminal shall indicate at least one condition or (bit) combination.	
				Coded as ASN.1 BER Coding OCTETSTRING (UNIVERSAL 4): <HB>] <LB>	
				b1 = '1'	A SICCT CLOSE CT SESSION (disconnection) is needed / expected by the terminal to end the download operation after the timeout period.
				b2 = '1'	The terminal needs a reboot / restart in order to end the download operation. Therefore, SICCT RESET CT is necessary after the timeout period.
				b3 = '1'	Terminal performs disconnection, reboot / restart automatically at the end of processing SICCT DOWNLOAD FINISH in order to end the download operation. Note: Do not set this b3 in combination with b1 and/or b2. Note: A new LAN connection must be established, if needed.
				b4 = '1'	The terminal needs manual assistance in order to end the download operation or to reboot / restart. Note: A new LAN connection must be established, if needed.
				'00'	Regular operation after Response
				'xy'	All other values RFU

5.5.10.27 CT SESSION Data Object

Das Cardterminal Session Data Object (CTSESS DO) muss im SICCT – Mode unterstützt werden und muss folgende Benutzerparameter zum rollenbasierten Zugriff und zur Identifikation einer Cardterminal Session enthalten

- Benutzername (Username)
- Benutzeridentifikation (Password)
- Session-Identifizier (Session ID)

CT Session Data Object (CTSESS DO)						
TAG	'69'		One byte tag according to SICCT-Specifications:			
			Tag coding according to ASN.1 BER see 5.5.10.3			
			BER-Coding: Application context, constructed, Tag-Number = 9 ('09')			
LEN	LEN coding see 5.5.10.3					
	one byte coding - LEN in the range of: 0 <= LEN <= 42					
	'06' ... '2A'		6 ... 42 Bytes		One byte coding	
VALUE	CT Session User Parameter					
	Username Data Object	Username	1st TLV DO	Tag	'13'	ASN.1 BER: Printable String
				Length	0 ... 12 Byte	
				Value	0 ... 12 alphanumeric characters (Byte)	
	Password Data Object	Password	2 nd TLV DO	Tag	'13'	ASN.1 BER: Printable String
				Length	0 ... 12 Byte	
				Value	0 ... 12 alphanumeric characters (Byte)	
	Session ID Data Object	Session ID	3 rd TLV DO	Tag	'13'	ASN.1 BER: Printable String
				Length	0 ... 12 Byte	
				Value	0 ... 12 alphanumeric characters (Byte)	

Für die Standard-Belegung (Factory Setting) der Benutzernamen und Passworte eines SICCT-Terminals muss gelten

- Role CT ADMIN: Benutzername = "admin" / Passwort = "admin"
- Role CT CONTROL: Benutzername = "user" / Passwort = "user".

Sofern die Administration des Terminals dieses zulässt, dürfen für einen optionalen anonymen Zugriff (CT CONTROL) auch leere Datenobjekte mit der Länge Null verwendet werden.

Übergibt ein Zugriff in der Administratorrolle beim Aufruf von SICCT CLOSE CT SESSION eine leere Session-ID (Tag, Länge Null, ohne Wert), so muss die CT-Session ohne (exakte) Kenntnis der Session-ID geschlossen werden.

5.5.10.28 Keypad Capabilities Data Object

Mittels des Keypad Capabilities Data Objects (KPC DO), welches im SICCT – Mode unterstützt werden darf, können die Eigenschaften einer physikalischen Eingabeeinheit (Keypad) dargestellt werden. Es muss sich um ein 'constructed' ASN.1-Datenobjekt handeln, welches weitere ASN.1-Datenobjekte aufnehmen kann. Die zulässigen Events bzw. Tastencodes, der Tasten sind in Kapitel Kapitel 'Schnittstellenbeschreibung zum Host' beschrieben.

Keypad Capabilities Data Object (KPC DO)				
TAG	'6A'	One byte tag according to SICCT-Specifications:		
		Tag coding according to ASN.1 BER see 5.5.10.3		
		BER-Coding: Application context, constructed, Tag-Number = 10 ('0A')		
LEN	LEN coding see 5.5.10.3			
	one byte coding - LEN in the range of: 0 <= LEN <= 127			
	'00' ... '7F'	0 <= LEN <= 127		One byte coding
VALUE	Keypad Options			
	Tag	LEN	'00' = None	
	'80'	'01'	Number of numeric Keys	
			b8...b1	Number of numeric Keys
	'81'	'01'	Number of alphanumeric Keys	
			b8...b1	Number of alphanumeric Keys
	'82'	'01'	Entry Key(s)	
			b8...b1	Number of Entry Key(s) available
			<> '00'	
	'83'	'01'	Abort / Cancel Key(s)	
			b8...b1	Number of Abort / Cancel Key(s) available
			<> '00'	
	'84'	'01'	Menu Key(s)	
			b8...b1	Number of Menu Key(s) available
			<> '00'	
	'85'	'01'	Correction Key(s)	
			b8...b1	Number of Correction Key(s) available
			<> '00'	
	'86'	'01'	Number of Function Key(s)	
			b8...b1	Number of Function Key(s) available
			<> '00'	

5.6 General Handling Instructions für den SICCT Betriebsmode

Zusätzlich zur Darstellung der Datenobjekte gibt dieses Kapitel einige generelle Beschreibungen zum Umgang mit Functional Units.

5.6.1 Handling von Display Messages im SICCT-Mode

Sofern ein Terminal ein (optionales) Display anbietet, muss ein Text (Display-Nachricht) explizit per Kommandoparameter in Form auf den aktuellen Befehlskontext bezogener Standard-Texte oder explizit über Daten Objekte zur Anzeige gebracht werden können. Für die Handhabung von Display-Nachrichten müssen die folgenden allgemeinen Vorgaben gelten.

- 1) Ein (optional vorhandenes) Terminal-Display muss mindestens den 7-bit-Zeichensatz ISO646DE unterstützen. In der Beschreibung zum Application Label Data Object (APPL DO) 5.5.10.18 sind die minimal zu unterstützenden Zeichen angegeben.
- 2) Textmeldungen können in Form von Application Label oder SICCT Message-To-be-Displayed Data Objects innerhalb von SICCT-Kommandos übergeben werden.
- 3) Innerhalb eines SICCT Befehls darf entweder die eine oder andere Objektvariante (Application Label oder SICCT Message-To-be-Displayed Data Objects) aber nicht beide Formen zugleich zum Einsatz kommen.
- 4) Im Fall, dass eine zu lange Display-Nachricht an Kommandos übergeben wird, muss die Nachricht unterdrückt werden, und es darf keine Ausgabe auf dem Display erfolgen. Dem Aufrufer muss ein entsprechender Status-Code ('6A80'), der auf ein fehlerhaftes Datenobjekt hinweist, zurückgegeben werden.
- 5) Der Zeichenvorrat für Messages muss sich generell nach dem aktiven Zeichensatz am Terminal und einer Menge definierter Steuerzeichen (CR, LF, und Kombinationen) sowie zulässiger Trennzeichen ('0B' und '0F') zur Formatierung von Display-Messages (s.u) richten. Eine tabellarische Übersicht findet sich in Kapitel 5.5.10.18 (APPL DO).
- 6) Im Fall, dass eine Nachricht Zeichen enthält, die kein Steuer- oder Trennzeichen darstellen und der Zeichensatz nicht enthält bzw. das Terminal Display nicht darstellen kann, ggf. Werte mit einem Code größer '7F', muss das Terminal das jeweils nicht unterstützte Zeichen durch ein Leerzeichen (Space, '20') ersetzen.
- 7) Im Gegensatz zu Steuerzeichen müssen Trennzeichen bei der Auswertung durch das Terminal unterdrückt und dürfen nicht als Textzeichen gezählt werden.
- 8) Der Aufbau von Display-Nachrichten in den Datenobjekten (APPL DO bzw. SMTBD DO) für PIN-Eingaben muss für das Value-Feld nach folgendem Schema erfolgen. Eine übergebene Text-Nachrichtenstruktur für PIN-Eingabe darf maximal 50 Zeichen (exkl. Trennzeichen) umfassen und unterteilt sich in
 - **<Message>** (max. 40 Zeichen) > **<Trennzeichen '0F'>** **<PIN-Prompt>** (max. 10 Zeichen) >
- 9) Die Darstellung von PIN-Eingaben während der Ausführung der Kommandos SICCT INPUT, PERFORM VERIFICATION, MODIFY REFERENCE DATA bedeutet, dass zuerst die max. 40 Zeichen lange **<Message>** in einer (ggf. horizontal scroll-baren) Zeile dargestellt werden, und nachfolgend unter der letzten Messagezeile die Ausgabe des **<PIN-Prompts>** erfolgen muss.
- 10) Die Darstellung des PIN-Prompts muss derart erfolgen, dass kein vertikales Scrollen erforderlich sein darf, um die PIN-Eingabe-Zeile zu erreichen. Die Anzeige der PIN-Eingabe-Zeile muss ggf. in der letzten sichtbaren Display-Zeile erfolgen.
- 11) Erfolgt in einer Nachricht keine Definition für einen **<PIN-Prompt>**, so muss der Standard-PIN-Prompt, bestehend aus dem Trennzeichen '0F', dem String "INPUT:" und eines abschliessenden Space-Zeichens, vom Terminal verwendet werden. Standard-PIN-Prompt: **<Trennzeichen '0F'>** **<"INPUT">** **<".:">** **<space>**
- 12) Für die Ausgabe von Display-Nachrichten müssen folgende minimal zu unterstützende Längen gelten
 - von 48 Zeichen für SICCT OUTPUT - Nachrichten.
 - von 50 (40 + 10 Zeichen) für PIN-Eingabe-Nachrichten (inklusive 10 Zeichen PINPrompt exklusive des Trennzeichens '0F')
- 13) Display Messages können zusätzlich Sollbruchstellen in Form des Trennzeichens '0B' für Textumbruch beinhalten. Das Terminal muss das Trennzeichen '0B' aus der Display-Nachricht erkennen und herausfiltern. Trennzeichen dürfen nicht als Textzeichen gezählt werden. Die Interpretation und Umsetzung des Textumbruchs darf herstellerspezifisch ausfallen.
- 14) Vor der Ausgabe eines neuen Display-Textes muss das adressierte Display gelöscht werden.

- 15) Das explizite Löschen eines Displays kann durch die Übergabe eines leeren Message-Datenobjekts erfolgen.
- 16) Nach der Ausführung eines SICCT-Kartenterminalkommandos müssen Textmeldungen bestehen bleiben, bis eine neue Ausgabe diese überschreibt.
- 17) Während der Ausführung kann das Terminal kontextbezogene Textnachrichten (z.B. Abbruch-Meldung nach Betätigung der Abbruch-Taste im Rahmen einer Benutzerinteraktion – siehe 5.7) über das Display ausgeben.
- 18) Lokale Benutzerinteraktionen (z.B. Tastenecho bei Keypad-Eingabe, Abbruch-Meldung) müssen direkt am Terminaldisplay dargestellt werden.
- 19) Die Darstellung impliziter Textnachrichten kann bei bestimmten SICCT-Kommandos über eine Option unterdrückt werden. Diese Kommandos müssen sich dann insofern 'display-neutral' verhalten, dass auch keine vom Terminal generierten Nachrichten angezeigt werden dürfen.
- 20) Die Darstellung der impliziten Display-Nachricht beim Anfordern und Deaktivieren einer Chipkarte durch die Kommandos SICCT REQUEST ICC und SICCT EJECT ICC muss entfallen, sofern eine Karte bereits eingeführt bzw. entnommen wurde.
- 21) Die Darstellung einer Kartenterminal-Idle-Message muss generell nach dem Reset des Terminals bzw. nach der Eröffnung bzw. nach dem Schließen einer Terminalsession oder explizit auf Anforderung per Option des SICCT OUTPUT Kommandos während einer Terminalsession erfolgen.
- 22) Der Inhalt der Idle-Message darf herstellerspezifisch ausfallen und über die Geräteadministration anpassbar gehalten werden.

5.7 Handling von Tastatureingaben (Keypad-Entries) im SICCT-Mode

- 1) SICCT-Terminals dürfen optional über eine oder mehrere Eingabeeinheiten (z.B. Keypads) verfügen, die für Benutzereingaben (z.B. PIN-Eingaben) oder Benutzerinteraktionen (Abbruch, Eingabekorrekturen) verwendet werden können.
- 2) Für SICCT-Kommandos, die eine Benutzerinteraktion erlauben, darf eine Aktion bzw. das entsprechende SICCT-Kommando per Eingabeeinheit (z.B. Abbruchtaste) abgebrochen werden. Um dem Anwender ein unmittelbares Feedback zu geben, muss an einem Terminal mit Display die Standardmeldung 12 "Abbruch" erscheinen, sofern Kommandooptionen die Ausgabe von Display-Nachrichten nicht unterbinden.
- 3) Die SICCT Kommandos SICCT INPUT, SICCT PERFORM VERIFICATION und SICCT MODIFY VERIFICATION DATA haben eine Eingabephase (input phase). Während der Eingabephase
 - a) muss das Terminal die Eingabe zeitlich mittels der Timer <t1>, <t2> und <t3> wie folgt überwachen.
 - i) Timer <t1> und <t3> werden zu Beginn der Eingabephase gestartet.
 - ii) Läuft der Timer <t1> ab, dann terminiert das Kommando mit einer Fehlermeldung.
 - iii) Timer <t1> stoppt mit dem ersten Tastendruck.
 - iv) Timer <t2> wird mit jedem Tastendruck neu gestartet.
 - v) Timer <t2> und <t3> stoppen, wenn die Eingabe beendet ist.
 - vi) Bei der Beschreibung der Kommandos wird festgelegt, wie diese auf den Ablauf von <t2> oder <t3> reagieren.
 - b) kann das Terminal optional Tasten-Events über das SICCT-Kommunikationsprotokoll, wie diese in Kapitel 'Schnittstellenbeschreibung zum Host' spezifiziert werden, versenden.
- 4) Secure PIN Entry - Während der Ausführung der SICCT-Kommandos SICCT PERFORM VERIFICATION und SICCT MODIFY VERIFICATION DATA müssen die Zeichen einer PIN-Eingabe am Display mit jeweils einem Ersatzzeichen (üblicherweise Sternchen) pro eingegebener Ziffer angezeigt werden. Zusätzlich wird dem Benutzer der sichere Eingabe-Modus ("Secure PIN Entry Mode") signalisiert. Es darf nicht möglich sein, die PIN-Eingabezeichen als Klartext im (optionalen) Terminal-Display darzustellen.

5.8 Command SICCT SELECT CT MODE

5.8.1 Funktion

Fehlerfreier Betrieb

Das Kommando SICCT SELECT CT MODE muss im SICCT – Betriebsmode unterstützt werden und muss sich an den Kommandointerpreter richten.

Sofern ein SICCT – Terminal beide zulässigen Terminal - Betriebsmodi SICCT (Default) - oder BCS - Mode (Option) unterstützt, kann mittels des SICCT SELECT CT MODE Kommandos der 'Aktive Betriebsmodus' für das Kartenterminal zur Laufzeit ausgewählt werden. Unterstützt der Terminal – Betriebsmode SICCT zusätzlich die Option zur Umschaltung der Chipkarten - Betriebsmodi zwischen ISO (Default) und EMV-Mode (Option), kann der Betriebsmode aller verfügbaren kontaktbehafteten Kartenkontaktiereinheiten (ICC-Slots) zwischen ISO (Default) und EMV-Mode (Option) gewählt werden. Die Aktivierung des jeweils zuletzt gewählten Betriebsmodus darf erst mittels eines nachfolgenden SICCT – bzw- BCS-Kommandos zum Reset des Kartenterminals erfolgen.

Der gewählte Betriebsmode muss als Parameter P2 übertragen werden. Sofern das Kartenterminal einen gewählten Betriebsmode unterstützen kann, muss es diesen übernehmen, und muss ein Statuswort SW1SW2 zurücksenden.

Der Betriebsmodus darf vom SICCT – Terminal erst zum Zeitpunkt der Abarbeitung des nächstfolgenden Resets per SICCT RESET CT bzw. BCS RESET CT Kommando aktiviert werden.

Kann ein SICCT – Terminal einen in Parameter P2 angeforderten Betriebsmode nicht (vollständig) unterstützen, darf die Auswahl des Betriebsmodes nicht übernommen werden, und es muss ein Fehlerstatus '6A00' mittels SW1SW2 zurückgegeben werden.

Konfigurationsdaten

Eine per SICCT SELECT CT MODE gewählte Einstellung muss für das gesamte Kartenterminal bis zum nächsten Kaltstart (Neustart nach System - Shutdown / Poweroff) gelten und darf die administrierte Terminalkonfiguration nicht verändern. Ein per SICCT SELECT CT MODE selektierter Betriebsmode darf nicht als persistente Geräteeinstellung übernommen werden. Nach einem Kaltstart muss das Kartenterminal in der Betriebsart starten, wie diese durch die Konfigurationsdaten des Terminals ('Standard - Betriebsmodus') vorgegeben wurden.

Ausführungsphasen SICCT SELECT CT MODE				
Phase		Stage	option	Description
preprocessing phase	Vorbereitungsphase	1		-
processing phase	Ausführungsphase	2		- check CT mode - set CT mode
postprocessing phase	Nachbereitungsphase	3		send return value

5.8.2 Anwendungsbedingungen

Sofern ein SICCT – Terminal beide zulässige Terminal – Betriebsmodi (CT Modes) unterstützt, muss das Kommando SICCT SELECT CT MODE in beiden Betriebsmodi SICCT und BCS (Option) verarbeitet werden können, um zwischen den Modi ('Aktiver Betriebsmodus') wechseln zu können.

Die optionale Umschaltung des Chipkarten - Betriebsmodus vom ISO (Default) zum EMV-Mode (Option) darf nur im Terminal – Betriebsmode SICCT unterstützt und muss im Terminal – Betriebsmodus BCS mit dem Fehlerstatus '6A00' in SW1SW2 negiert werden.

SICCT SELECT CT MODE							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
BCS	SICCT			Stage			
					1	2	3
✓	✓	✓	✓	no	-	-	-

Das Kommando darf nicht über eine Reset- oder Neustartfunktion verfügen und kann (außerhalb einer SICCT DOWNLOAD Session) zu beliebigem Zeitpunkt an das Terminal gesendet werden. Folgt dem Befehl nicht unmittelbar ein SICCT RESET CT Kommando, muss der zuvor aktive Betriebsmodus unverändert bleiben.

Ausnahmebehandlung

Für den Fall, dass ein gewählter Betriebsmode bereits aktiviert wäre, muss das Kartenterminal diese Situation per Rückgabe eines gesonderten Statuscodes "Warning - Specified Mode already set." signalisieren

Abbruchbedingung

Das Kommando SICCT SELECT CT MODE darf keine direkt erkennbare Auswirkung auf den Benutzer des Kartenterminals haben und kann durch diesen nicht beeinflusst oder abgebrochen werden.

Die Wirkung des Kommandos muss elementar und für die steuernde Entität in der Ausführung nicht abbrechbar sein.

5.8.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	[Lc]	[Data]	[Le]
SICCT SELECT CT MODE	'80'	'20'	Functional Unit	Command Qualifier	Length Command Data	Command Data	absent
	- CLA = Class - INS = Instruction - P1, P2 = Parameter 1 and 2 - Lc = Length of command data field - Le = Length of expected data				Case 1 (no cmd data, no rsp data): no Lc, no Le		
					Case 3 (cmd data, no rsp data): Lc=1-255 Bytes no Le		

Specification C-APDU		
CLA	'80'	Cardterminal Command Class
INS	'20'	SICCT SELECT CT MODE

P1	Functional Unit	
	bit8...bit1	Referenced Coding
		'FF' Escape: Signal Referenced Coding of P1 Functional Unit referenced by Functional Unit Index Data Object. FUI DO contained within Command Data Field.
		Direct Coding (mandatory)
		'00' Address Cardterminal
		other values RFU

P2	Command Qualifier	
	bit8...bit1	bit1 = 0 Select BCS Mode
		bit1 = 1 Select SICCT Mode
		bit2 = 0 Select ISO Mode (for all available ICC-Slots)
		In case bit1 == 1, SICCT Mode
		bit2 = 1 Select EMV Mode (for all available ICC-Slots)
		other values RFU

Lc	Length of Command Data Nc		
	Direct Coding (mandatory)		
	Empty	Lc absent	Nc = 0
	Referenced Coding		
	var	Length of FUI DO contained within Command Data	
	FUI DO	Lc short Lc = '04'	Nc = 4

Data	Command Data		
	In case of Direct Coding of 'P1' (mandatory)		
	Empty	No data present	
	In case of Referenced Coding of 'P1'		
	FUI DO	'84020000'	Functional Unit Index Data Object referencing the cardterminal.

Le	Length of Requested Data Ne Return up to Ne bytes of requested information		
	Empty	Le absent	No information requested Ne = 0

5.8.4 Data Objects

Mit dem SICCT SELECT CT MODE Kommando muss ein Functional Unit Index Data Object (FUI DO) übergeben werden, sofern Referenced Coding für den Parameter P1 gewählt wird.

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	In case of Referenced Coding: Functional Unit Index Data Object referenc- ing the cardterminal: '84020000'	see 5.5.10.9

5.8.5 Response Structure

SICCT SELECT CT MODE	Kodierung R-APDU		
	[Body]		Trailer
	[Requested Data / Information]		Status Byte 1 Status Byte 2
	Empty	no requested information	SW1 SW2

5.8.6 Status-Codes SW1SW2

Der Ausgang des Kommandos SICCT SELECT CT MODE muss über das SW1SW2-Statuswort an den Aufrufer signalisiert werden. Der Wert '90' in SW1 signalisiert den Erfolgsfall. Alle anderen SW1-Werte signalisieren eine Warnung oder einen Fehlerzustand. SW2 kann den weiteren Zustand qualifizieren.

SW1SW2	Specification	Priority Priorität	Meaning / Bedeutung
'6200'	Warning - Specified Mode already set.	11	- The specified CT mode had already been activated. - Der angegebene Terminal – Betriebsmode wurde bereits aktiviert.
'6700'	Wrong (command) length	3	- Wrong Lc / falscher Wert Lc: Inconsistent command body / Inkonsistenter Kommandoaufbau - No Command Data field supported. / kein Datenfeld zulässig
'6900'	Command not allowed	9	- No open CT Session found / Keine offene CT Session vorgefunden. - The command is not allowed within an open SICCT Download - Session / Das Kommando ist in einer geöffneten SICCT Download - Session nicht zulässig.
'6941'	Functional Unit busy / not available	10	- At the moment FU_CT is busy / unavailable - Funktionseinheit FU_CT ist z.Zt. belegt / nicht verfügbar
'6A00'	Wrong parameters P1, P2	5	- P1 addresses invalid Functional Unit. - P1 has to address FU_CT / P1 muss FU_CT adressieren. - In case of Referenced Coding – Invalid FUI DO referenced by P1 (command data). FUI DO has to address FU_CT / Das FUI DO muss FU_CT adressieren. - P2 specifies an invalid / not supported value - P2 gibt einen nicht spezifizierten / unterstützten Wert an.
'6A80'	Invalid Data Object	6	- In case of Referenced Coding – The data field contains an invalid DO / - Das Datenfeld enthält ein unzulässiges DO.
'6A88'	Missing Data Object	7	- In case of Referenced Coding – The data field does not contain a FUI DO / - Das Datenfeld enthält kein FUI DO.
'6A89'	Too many Data Objects	8	Too many Data Objects (FUI DO) of same type. Es sind zu viele Datenobjekte gleichen Typs (FUI DO) enthalten.
'6C00'	Wrong Length Le	4	- Wrong Le / falscher Wert Le: - Le exists, but is not allowed to. - Le existiert, darf aber nicht angegeben werden.
'6D00'	Wrong Instruction	2	Wrong / unbekanntes Instruction-Byte
'6E00'	Class not supported	1	Falsches / unbekanntes Class-Byte
'6F00'	Communication with CT not possible	11	- In case - the terminal detects general protocol errors or - the terminal detects errors within CT Session or TLS connection
'9000'	Mode Selection was successful	12	- The specified CT mode has been selected. - Der angegebene Terminal – Betriebsmode wurde gewählt. - In order to activate the mode the cardterminal has to be reset. Um den Modus zu aktivieren, muss ein Reset auf das Terminal erfolgen.

5.9 Command SICCT CONTROL COMMAND

5.9.1 Funktion

Das Kommando SICCT CONTROL COMMAND⁵ muss im SICCT – Betriebsmode unterstützt werden und muss sich an den Kommandointerpreter richten.

Das Kommando SICCT CONTROL COMMAND darf zwei funktionale Varianten aufweisen:

- Es kann Auskunft über ein zuvor gesendetes SICCT - Kommando (Funktion "GET STATUS") erteilen
- Es kann ein an das Terminal bereits gesendetes SICCT - Kommando zur Laufzeit abbrechen oder beenden, wenn das SICCT – Terminal diese Option (Command Queue) unterstützt. Ein SICCT - Kommando, das sich noch in einem Wartestatus vor der Ausführung befindet (Command Queue), muss in diesem Fall aus dem Speicher gelöscht (Funktion "TERMINATE") werden.

Fehlerfreier Betrieb

Das SICCT CONTROL COMMAND kann der Steuerung der Kommandobearbeitung im SICCT – Betriebsmodus dienen. Das Kommando SICCT CONTROL COMMAND muss sich an den Kommandointerpreter richten und kann diesem eine Kommandosequenznummer des zu adressierenden Kommandos übergeben. Im Fall, dass der Kommandoverwalter die Kommandosequenznummer auffinden kann, muss er dem Aufrufer den Status der Operation mitteilen. Für die Variante "TERMINATE" kann er die Ausführung des Kommandos zusätzlich unterbinden bzw. beenden.

Ausführungsphasen SICCT CONTROL COMMAND				
Phase		Stage	option	Description
Preprocessing phase	Vorbereitungsphase	1		
Processing phase	Ausführungsphase	2		▪ check sequence number ▪ check for termination condition ▪ terminate corresponding command
Postprocessing phase	Nachbereitungsphase	3		▪ send return value

5.9.2 Anwendungsbedingungen

Das Kommando SICCT CONTROL COMMAND darf im Betriebsmodus BCS nicht unterstützt werden.

Abbruchbedingung

Das Kommando SICCT CONTROL COMMAND kann selbst nicht abgefragt, beeinflusst oder abgebrochen werden.

SICCT CONTROL COMMAND							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
BCS	SICCT			Stage			
				1	2	3	
no	✓	✓	✓	no	-	-	-

⁵ Anmerkung: in den früheren Versionen SICCT V1.0, V1.03 wurde ein Vorläufer dieses Kommando noch unter der Bezeichnung "SICCT TERMINATE COMMAND" aufgeführt.

5.9.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	Lc	Data	[Le]
SICCT CONTROL COMMAND	'80'	'27'	Functional Unit	Command Qualifier	Length Command Data	Command Data	absent
	- CLA = Class - INS = Instruction - P1, P2 = Parameter 1 and 2 - Lc = Length of command data field - Le = Length of expected data				Case 3 (cmd data, no rsp data:) Lc=1-255 Bytes no Le		

Specification C-APDU		
CLA	'80'	Cardterminal Command Class
INS	'27'	SICCT CONTROL COMMAND

P1	Functional Unit	
	bit8...bit1	Referenced Coding
		'FF' Escape: Signal Referenced Coding of P1 Functional Unit referenced by Functional Unit Index Data Object (FUI DO. One FUI DO contained within Command Data Field.
		Direct Coding (mandatory)
		'00' Address Cardterminal
		other values RFU

P2	Command Qualifier		
	bit8...bit1	'00'	termination of command before processing (queue)
		'01'	termination of command at stage 1 or before processing (if the command allows to abort at stage 1)
		'02'	termination of command at or before stage 2 (if the command allows to abort at stage 2)
		'03'	termination of command at or before stage 3 (if the command allows to abort at stage 3)
		'0F'	Force abruption of command at once (abort - if the command allows)
		'0E'	Discard all commands in queue (if queue available)
		'80'	checking command at any stage ("GET STATUS")
		other values RFU	

Lc	Length of Command Data Nc		
	Direct Coding (mandatory, unless P2='0E')		
	var	Length of Sequence Number Data Object	
		Lc short	Nc = 4
	Referenced Coding		
	var	Length of FUI DO and Sequence Number Data Object	
		Lc short	Nc = 8

Data	Command Data		
	In case of Direct Coding of 'P1' (mandatory) and 'P2' <> '0E'		
	SEQNO DO	see 5.5.10.23	Sequence Number Data Object
			'0000' <= Sequence Number <= 'FCFF'
	In case of Referenced Coding of 'P1'		
	FUI DO	'84020000'	Functional Unit Index Data Object referencing the cardterminal.
	In case of Referenced Coding of 'P1' and 'P2' <> '0E'		
	SEQNO DO	see 5.5.10.23	Sequence Number Data Object
'0000' <= Sequence Number <= 'FCFF'			

Le	Length of Requested Data Ne Return up to Ne bytes of requested information		
	Empty	Le absent	No information requested Ne = 0

5.9.4 Data Objects

Mit dem SICCT CONTROL COMMAND Kommando muss ein Functional Unit Index Data Object (FUI DO) übergeben werden, sofern Referenced Coding für den Parameter P1 gewählt wird. Die Angabe der eindeutigen Kommandosequenznummer (außer im Fall P2='0E') muss mittels Übergabe eines SEQNO Datenobjekts erfolgen.

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	In case of Referenced Coding: Functional Unit Index Data Object referencing the cardterminal: '84020000'	see 5.5.10.9
SEQNO DO	CMD	Sequence Number Data Object Sequence number of command to terminate.	see 5.5.10.23

5.9.5 Response Structure

SICCT CONTROL COMMAND	Kodierung R-APDU			
	[Body]		Trailer	
	[Requested Data / Information]		Status Byte 1	Status Byte 2
	Empty	no requested information	SW1	SW2

5.9.6 Status-Codes SW1SW2

Der Ausgang des Kommandos muss über das SW1SW2-Statuswort an den Aufrufer signalisiert werden. Der Wert '90' in SW1 signalisiert den Erfolgsfall. Alle anderen SW1-Werte signalisieren eine Warnung oder einen Fehlerzustand. SW2 kann weiteren Zustand qualifizieren.

SW1SW2	Specification	Priority / Priorität	Meaning / Bedeutung
'6200'	Warning - Specified command not found	11	- The specified command could not be found. - Das angegebene Kommando wurde nicht gefunden.
'64xx'	Warning: Command not terminated.	12	- The ct could not terminate the specified command. / - Das KT konnte das angegebene Kommando nicht beenden.
'6401'		12	- Not terminated; command currently at stage 1 / - Nicht beendet, Kommandoausführung aktuell in Stufe 1
'6402'		12	- Not terminated; command currently at stage 2 / - Nicht beendet, Kommandoausführung aktuell in Stufe 2
'6403'		12	- Not terminated; command currently at stage 3 / - Nicht beendet, Kommandoausführung aktuell in Stufe 3
'6700'	Wrong (command) length	3	- Wrong Lc / falscher Wert Lc: - Inconsistent command body / - Inkonsistenter Kommandoaufbau - In case P2 = '0E' and direct coding: No Command Data field supported. / kein Datenfeld zulässig
'6900'	Command not allowed	9	- No open CT Session found / Keine offene CT Session vorgefunden. - The command is not allowed within an open SICCT - Download - Session / Das Kommando ist in einer geöffneten SICCT Download - Session nicht zulässig.
'6941'	Functional Unit busy / not available	10	- At the moment FU_CT is busy / unavailable - Funktionseinheit FU_CT ist z.Zt. belegt / nicht verfügbar
'6A00'	Wrong parameters P1, P2	5	- P1 addresses invalid Functional Unit. - P1 has to address FU_CT / P1 muss FU_CT adressieren. - In case of Referenced Coding – - Invalid FUI DO referenced by P1 (command data). - FUI DO has to address FU_CT / Das FUI DO muss FU_CT adressieren. - P2 specifies an invalid / not supported value - P2 gibt einen nicht spezifizierten / unterstützten Wert an.
'6A80'	Invalid Data Object	6	- In case of Referenced Coding – - The data field contains an invalid DO / - Das Datenfeld enthält ein unzulässiges DO.
'6A88'	Missing Data Object	7	- In case of Referenced Coding – - The data field does not contain a FUI DO / - Das Datenfeld enthält kein FUI DO.
'6A89'	Too many Data Objects	8	- Too many Data Objects (FUI DO) of same type. - Es sind zu viele Datenobjekte gleichen Typs (FUI DO) enthalten.
'6C00'	Wrong Length Le	4	- Wrong Le / falscher Wert Le: - Le exists, but is not allowed to. - Le existiert, darf aber nicht angegeben werden.
'6D00'	Wrong Instruction	2	Wrong / unbekanntes Instruction-Byte
'6E00'	Class not supported	1	Falsches / unbekanntes Class-Byte

SW1SW2	Specification	Priority / Priorität	Meaning / Bedeutung
'6F00'	Communication with CT not possible	11	- In case - the terminal detects general protocol errors or - the terminal detects errors within CT Session or TLS connection
'90xx'	Operation was successful	12	The specified command with the given sequence number (not P2='0E') is in actual stage 'xx' – or has been terminated at stage 'xx' / Das Kommando mit der angegebenen Sequenznummer (nicht bei P2='0E') befindet sich in der aktuellen Stufe 'xx' – oder wurde vom Terminal in der Stufe 'xx' beendet.
'9000'			- Command status: prior to execution (queue) - or command discarded before execution – or all commands discarded before execution / - Kommandostatus: bevor Ausführung (queue) – oder Kommando vor der Ausführung beendet - oder Kommandos vor der Ausführung beendet
'9001'			- Command status: preprocessing - or command terminated at preprocessing phase / - Kommandostatus: im Zustand 'Vorbereitungsphase' - oder Kommandos wurde im Zustand 'Vorbereitungsphase' beendet
'9002'			- Command status: processing - or command terminated at processing phase / - Kommandostatus: im Zustand 'Ausführungsphase' - oder Kommandos wurde im Zustand 'Ausführungsphase' beendet
'9003'			- Command status: postprocessing - or command terminated at postprocessing phase / - Kommandostatus: im Zustand 'Nachbereitungsphase' - oder Kommandos wurde im Zustand 'Nachbereitungsphase' beendet
'900F'			- Command abortion forced / - Kommandoabbruch wurde forciert.
'90FF'			- Command status unknown - or command terminated at unknown stage / - Kommandostatus: unbekannt - oder Kommandos wurde im unbekannten Zustand beendet

5.10 Command SICCT INIT CT SESSION

5.10.1 Funktion

Der Zugriff auf das SICCT-Terminal erfolgt stets rollenbasiert im Rahmen einer Cardterminal Session (CT Session). Bevor Kommandos vom SICCT-Kartenterminal verarbeitet werden können, muss eine SICCT - CT Session geöffnet werden. Eine SICCT - CT Session muss solange bestehen, bis ein korrespondierendes Kommando SICCT CLOSE CT SESSION die geöffnete Session wieder schließt oder die Netzwerkverbindung zum SICCT-Kommandointerpreter beendet wurde.

Zur Eröffnung der CT Session dient das Kommando SICCT INIT CT SESSION, welches folgende Funktionen wahrnehmen muss

- Benutzer- Identifikation und Authentifizierung,
- Zugriffsreglung entsprechend der Benutzerrolle,
- Session-Initialisierung (Vergabe einer CT Session-ID),
- Öffnen (Benutzer-Login) der CT Session.

Je nachdem ob die Login-Informationen eine Benutzerkennung und Authentifizierungsdaten (Password-Phrase) enthält, muss das SICCT-Terminal den Zugriff anhand der Rollenangabe differenzieren

- Administrator-Rolle (CT ADMIN Session)
- (optional anonyme) Control-Rolle (CT CONTROL Session).

Das Terminal muss die Login-Informationen gegen zuvor administrierte Zugangsdaten prüfen. Es obliegt der Terminalkonfiguration, ob ein anonym Zugriff, d.h. ohne Angabe von Benutzer und Passwortphrase erfolgen darf.

Stereotyp zum Verhalten des Kommandos SICCT RESET CT müssen zum Zeitpunkt der CT-Session-Eröffnung alle Functional Units (FU) zurückgesetzt bzw. Kartenslots deaktiviert werden. Optional vorhandene Displays müssen nach der Ausführung jeweils die Kartenterminal Idle-Message anzeigen.

Fehlerfreier Betrieb

Der Aufrufer muss mit dem SICCT INIT CT SESSION Kommando ein Datenobjekt CT Session Data Object übermitteln, welches folgende Login-Information in Form dreier TLV-Objekte enthalten muss

- Benutzername (Username, CT CONTROL: kann leeres DO sein)
- Benutzerpasswort (Password, CT CONTROL: kann leeres DO sein)
- CT Session ID (Tag, Len = 0, leeres Datenobjekt).

Der SICCT-Kommandointerpreter muss den Zugang prüfen und anhand zuvor hinterlegter Zugangsdaten eine Benutzerrolle (SICCT-Administrator oder SICCT-Benutzer) und eine Session-ID zuordnen. Das Terminal muss dem Aufrufer ebenfalls ein CT Session Data Object (CTSESS DO) mit folgendem Inhalt zurücksenden:

- Benutzername (Username, CT-CONTROL: kann leeres DO sein)
- Benutzerpasswort (Tag, Len = 0, leeres Datenobjekt)
- CT Session ID.

Ausführungsphasen SICCT INIT CT SESSION				
Phase		Stage	option	Description
preprocessing phase	Vorbereitungsphase	1		
processing phase	Ausführungsphase	2		▪ check username ▪ check identification data (password) ▪ check and grant authorization ▪ open ct session
postprocessing phase	Nachbereitungsphase	3		▪ send return value

5.10.2 Anwendungsbedingungen

Das Kommando SICCT INIT CT SESSION muss sich über eine bestehende Verbindung an den SICCT-Kommandointerpreter richten. Die zur Prüfung der Login Information benötigten Konfigurationsdaten müssen im Kartenterminal vorliegen.

Nach der Kommandoausführung müssen alle Functional Units im Reset-Zustand bzw. deaktiviert sein. Optional vorhandene Anzeigeeinrichtungen (Displays) müssen nach der Ausführung jeweils die Karten-terminal Idle-Message anzeigen.

Abbruchbedingung

Das Kommando SICCT INIT CT SESSION kann selbst nicht beeinflusst oder abgebrochen werden.

SICCT INIT CT SESSION					
Conditions					
CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
			Stage		
			1	2	3
✓	✓	No	-	-	-

5.10.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	Lc	Data	Le
SICCT INIT CT SESSION	'80'	'28'	Functional Unit	'00'	Length Command Data	Command Data	Length Requested Data
	▪ CLA = Class ▪ INS = Instruction ▪ P1, P2 = Parameter 1 and 2 ▪ Lc = Length of command data field ▪ Le = Length of expected data				Case 4 (cmd data, rsp data): Lc=1-255 Bytes Le=1-256 Bytes		

Specification C-APDU		
CLA	'80'	Cardterminal Command Class
INS	'28'	SICCT INIT CT SESSION

P1	Functional Unit		
	bit8...bit1	Referenced Coding	
		'FF'	Escape: Signal Referenced Coding of P1 Functional Unit referenced by Functional Unit Index Data Object. FUI DO contained within Command Data Field.
		Direct Coding (mandatory)	
		'00'	Address Cardterminal
		other values RFU	

P2	Command Qualifier		
	bit8...bit1	'00'	other values RFU

Lc	Length of Command Data Nc		
	Direct Coding (mandatory)		
	var	Length of CT Session Data Object	
		Lc short	Nc <= 255
	Referenced Coding		
	var	Length of FUI DO and CT Session Data Object	
		Lc short	Nc <= 255

Data	Command Data						
	In case of Direct Coding of 'P1' (mandatory)						
	CTSESS DO	CT Session Data Object					
		'69'	L	Value			
			'13'	Len	<1 ... 12 Byte>	Username	
			'13'	Len	<1 ... 12 Byte>	Password	
			'13'	'00'	Absent	Session ID	
	In case of Referenced Coding of 'P1'						
	FUI DO	'84020000'		Functional Unit Index Data Object referencing the cardterminal.			
	CTSESS DO	CT Session Data Object					
Coding see above: Direct Coding.							

Le	Length of Requested Data Ne		
	Return up to Ne bytes of requested information		
	variable length	Le short '01 <= Le <= 'FF'	1 <= Ne <= 255
		Le short Le = '00'	Ne = 256:

5.10.4 Data Objects

Mit dem SICCT INIT CT SESSION Kommando muss ein Functional Unit Index Data Object (FUI DO) übergeben werden, sofern Referenced Coding für den Parameter P1 gewählt wird.

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	In case of Referenced Coding: Functional Unit Index Data Object referencing the CardTerminal: '84020000'	see 5.5.10.9
CTSESS DO	CMD	CT Session Data Object	see 5.5.10.27
	RSP		

5.10.5 Response Structure

SICCT INIT CT SESSION	Kodierung R-APDU							
	[Body]					Trailer		
	[Requested Data / Information]					Status Byte 1	Status Byte 2	
	Empty	in case of error					SW1	SW2
	CTSESS DO	CT Session Data Object Returns Username and Session-ID.						
		'69'	L	Value				
			'13'	Len	<1 ... 12 Byte>	Username		
			'13'	'00'	Absent	Password		
		'13'	Len	<1 ... 12 Byte>	Session ID			

5.10.6 Status-Codes SW1SW2

Der Ausgang des Kommandos muss über das SW1SW2-Statuswort an den Aufrufer signalisiert werden. Der Wert '90' in SW1 signalisiert den Erfolgsfall. Alle anderen SW1-Werte signalisieren eine Warnung oder einen Fehlerzustand. SW2 kann den weiteren Zustand qualifizieren.

SW1SW2	Specification	Priority / Priorität	Meaning / Bedeutung
'6400'	Error - Opening CT Session was not successful	12	Error / Fehler - Invalid / ungültiger Username - Invalid / ungültiges Password - Invalid Session-ID. DO Session ID has to be empty. Ungültiges Session-ID. DO Die Session ID muss leer sein
'6403'			
'6700'	Wrong (command) length	3	- Wrong Lc: Inconsistent command body or - no Command Data field supported. /
'6900'	Command not allowed	9	Open CT Session found / Offene CT Session vorgefunden
'6941'	Functional Unit busy / not available	10	At the moment FU_CT is busy / unavailable. Funktionseinheit FU_CT ist z.Zt. belegt / nicht verfügbar
'6A00'	Wrong parameters P1, P2	5	- P1 addresses invalid Functional Unit. P1 has to address FU_CT / P1 muss FU_CT adressieren - In case of Referenced Coding – Invalid FUI DO referenced by P1 (command data). FUI DO has to address FU_CT / Das FUI DO muss FU_CT adressieren. - P2 specifies an invalid / not supported value P2 gibt einen nicht spezifizierten / unterstützten Wert an.
'6A80'	Invalid Data Object	6	- Incorrect data objects / parameters in the command data field - In case of Referenced Coding – The data field contains an invalid DO / Das Datenfeld enthält ein unzulässiges DO. - Invalid FUI DO / Unzulässiges FUI DO Invalid CTSESS DO / Unzulässiges CTSESS DO
'6A88'	Missing Data Object	7	- In case of Referenced Coding – The data field does not contain a FUI DO / Das Datenfeld enthält kein FUI DO. - The data field does not contain a CTSESS DO / Das Datenfeld enthält kein CTSESS DO.
'6A89'	Too many Data Objects	8	- Too many Data Objects (FUI DO) of same type. - Es sind zu viele Datenobjekte gleichen Typs (FUI DO oder CTSESS DO) enthalten.
'6C00'	Wrong Length Le	4	Wrong Le / falscher Wert Le:
'6D00'	Wrong Instruction	2	Wrong / unbekanntes Instruction-Byte
'6E00'	Class not supported	1	Falsches / unbekanntes Class-Byte
'6F00'	Communication with CT not possible	11	- In case - the terminal detects general protocol errors or - the terminal detects errors within CT Session or TLS connection
'9000'	Opening CT Session was successful	12	The specified CT session has been opened. / Die angegebene CT Session wurde geöffnet.

5.11 Command SICCT CLOSE CT SESSION

5.11.1 Funktion

Das Kommando SICCT CLOSE CT SESSION muss eine CT Session schließen. Stereotyp zum Verhalten des Kommandos SICCT RESET CT müssen zum Zeitpunkt des Schließens alle Functional Units (FU) deaktiviert werden.

Fehlerfreier Betrieb

Der Aufrufer muss mit dem SICCT CLOSE CT SESSION Kommando ein Datenobjekt CT Session Data Object (CTSESS DO) übermitteln. Der SICCT-Kommandointerpreter muss die Gültigkeit der Session-ID prüfen und nach Prüfung der Zugriffsberechtigung eine offene Cardterminal Session schließen.

Sonderfall – Beenden einer CT-Session durch den Administrator

Im Fall eines Administratorzugriffs, d.h. Username und Passwort-Phrase konnten erfolgreich einer Administratorrolle zugewiesen werden, kann eine offene CT_Session ebenfalls geschlossen werden, wenn der Aufruf ein leeres Session ID Datenobjekt (Länge NULL) angibt.

Bei Verlust der Netzwerkverbindung des SICCT-Kommandointerpreters wird eine geöffnete CT Session auch unabhängig vom Kommando SICCT CLOSE CT SESSION geschlossen.

Nach der erfolgreichen Kommandoausführung müssen alle Functional Units im Reset-Zustand bzw. deaktiviert sein, und das Terminal darf keine offene CT Session unterhalten.

Optional vorhandene Anzeigeeinrichtungen (Displays) müssen nach der Ausführung die Kartenterminal-Idle-Message anzeigen.

Ausführungsphasen SICCT CLOSE CT SESSION				
	Phase	Stage	option	Description
preprocessing phase	Vorbereitungsphase	1		
processing phase	Ausführungsphase	2		▪ check ct session state ▪ check ct session id ▪ close ct session
postprocessing phase	Nachbereitungsphase	3		▪ send return value

5.11.2 Anwendungsbedingungen

Das Kommando SICCT CLOSE CT SESSION muss sich über eine bestehende Verbindung an den SICCT-Kommandointerpreter richten.

Das Beenden einer CT-Session ohne Administrator-Berechtigung kann nur unter Angabe der Session-ID geschehen. Das Beenden einer CT-Session ohne korrespondierende Session-ID darf nur mit Administrator-Berechtigung erfolgen.

Normalfall - Beenden einer CT-Session (ohne Administratorberechtigung)

Im Fall eines User-Zugriffs muss die Session-ID im CTSESS DO korrekt zum Wert der offenen CT Session angegeben sein. Im CTSESS DO können optional auch Username und Passwort-Phrase angegeben sein. Sofern diese angegeben wurden, muss das Terminal neben der Session-ID die Zugangsberechtigung prüfen, und bei fehlender Übereinstimmung bzw. Berechtigung die Ausführung des Kommandos abbrechen.

Abbruchbedingung

Das Kommando SICCT CLOSE CT SESSION kann selbst nicht beeinflusst oder abgebrochen werden.

SICCT CLOSE CT SESSION					
Conditions					
CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
			Stage		
			1	2	3
✓	✓	no	-	-	-

5.11.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	Lc	Data	[Le]
SICCT CLOSE CT SESSION	'80'	'29'	Functional Unit	'00'	Length Command Data	Command Data	absent
	- CLA = Class - INS = Instruction - P1, P2 = Parameter 1 and 2 - Lc = Length of command data field - Le = Length of expected data				Case 3 (cmd data, no rsp data:) Lc=1-255 Bytes no Le		

Specification C-APDU		
CLA	'80'	Cardterminal Command Class
INS	'29'	SICCT CLOSE CT SESSION

P1	Functional Unit		
	bit8...bit1	Referenced Coding	
		'FF'	Escape: Signal Referenced Coding of P1 Functional Unit referenced by Functional Unit Index Data Object. FUI DO contained within Command Data Field.
		Direct Coding (mandatory)	
		'00'	Address Cardterminal
		other values RFU	

P2	Command Qualifier		
	bit8...bit1	'00'	other values RFU

Lc	Length of Command Data Nc		
	Direct Coding (mandatory)		
	var	Length of CT Session Data Object	
		Lc short	Nc <= 255
	Referenced Coding		
	var	Length of FUI DO and CT Session Data Object	
		Lc short	Nc <= 255

Data	Command Data						
	In case of Direct Coding of 'P1' (mandatory)						
	CTSESS DO	Cardterminal Session Data Object					
		'69'	L	Value			
			'13'	Len	<0 ... 12 Byte>	Username	
			'13'	Len	<0 ... 12 Byte>	Password	
			'13'	Len	<0 ... 12 Byte>	Session ID	
			'00'	Absent	Admin: Close all sessions.		
	In case of Referenced Coding of 'P1'						
	FUI DO	'84020000'		Functional Unit Index Data Object referencing the cardterminal.			
CTSESS DO	Cardterminal Session Data Object						
	Coding see above: Direct Coding.						

Le	Length of Requested Data Ne Return up to Ne bytes of requested information		
	Empty	Le absent	No information requested Ne = 0

5.11.4 Data Objects

Mit dem SICCT CLOSE CT SESSION Kommando muss ein Functional Unit Index Data Object (FUI DO) übergeben werden, sofern Referenced Coding für den Parameter P1 gewählt wird.

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	In case of Referenced Coding: Functional Unit Index Data Object referenc- ing the cardterminal: '84020000'	see 5.5.10.9
CTSESS DO	CMD	Datenobjekt CT Session Data Object	see 5.5.10.27

5.11.5 Response Structure

SICCT CLOSE CT SESSION	Kodierung R-APDU			
	[Body]			Trailer
	[Requested Data / Information]			Status Byte 1 Status Byte 2
	Empty	no requested information		SW1 SW2

5.11.6 Status-Codes SW1SW2

Der Ausgang des Kommandos muss über das SW1SW2-Statuswort an den Aufrufer signalisiert werden. Der Wert '90' in SW1 signalisiert den Erfolgsfall. Alle anderen SW1-Werte signalisieren eine Warnung oder einen Fehlerzustand. SW2 kann den weiteren Zustand qualifizieren.

SW1SW2	Specification	Priority / Priorität	Meaning / Bedeutung
			Error / Fehler
'6400'	Error - Closing CT Session was not successful	12	- Invalid / ungültiger Username - Invalid / ungültiges Password
'6403'			Invalid Session-ID / Ungültige Session-ID
'6700'	Wrong (command) length	3	Wrong Lc: Inconsistent command body or no Command Data field supported. /
'6900'	Command not allowed	9	No Open CT Session found / Keine offene CT Session vorgefunden
'6941'	Functional Unit busy / not available	10	At the moment FU_CT is busy / unavailable. Funktionseinheit FU_CT ist z.Zt. belegt / nicht verfügbar
'6A00'	Wrong parameters P1, P2	5	- P1 addresses invalid Functional Unit. P1 has to address FU_CT / P1 muss FU_CT adressieren. - In case of Referenced Coding – Invalid FUI DO referenced by P1 (command data). FUI DO has to address FU_CT / Das FUI DO muss FU_CT adressieren. - P2 specifies an invalid / not supported value P2 gibt einen nicht spezifizierten / unterstützten Wert an.
'6A80'	Invalid Data Object	6	- Incorrect data objects / parameters in the command data field - In case of Referenced Coding – The data field contains an invalid DO / Das Datenfeld enthält ein unzulässiges DO. - Invalid FUI DO / Unzulässiges FUI DO - Invalid CTSESS DO / Unzulässiges CTSESS DO
'6A88'	Missing Data Object	7	- In case of Referenced Coding – The data field does not contain a FUI DO / Das Datenfeld enthält kein FUI DO. - The data field does not contain a CTSESS DO / Das Datenfeld enthält kein CTSESS DO.
'6A89'	Too many Data Objects	8	Too many Data Objects (FUI DO) of same type. Es sind zu viele Datenobjekte gleichen Typs (FUI DO oder CTSESS DO) enthalten.
'6C00'	Wrong Length Le	4	Wrong Le / falscher Wert Le:
'6D00'	Wrong Instruction	2	Wrong / unbekanntes Instruction-Byte
'6E00'	Class not supported	1	Falsches / unbekanntes Class-Byte
'6F00'	Communication with CT not possible	11	- In case - the terminal detects general protocol errors or - the terminal detects errors within CT Session or TLS connection
'9000'	Closing CT Session was successful	12	The specified CT session has been opened. / Die angegebene CT Session wurde geöffnet.

5.12 Command SICCT RESET CT / ICC

5.12.1 Funktion

Das Kommando SICCT RESET CT / ICC muss dem selektiven Rücksetzen (Reset) einer vorhandenen Chipkarte (ICC oder RFID Token bzw. PICC) bzw. einer Chipkartenfunktionseinheit oder alternativ aller adressierbaren und verfügbaren Funktionseinheiten dienen. Das Rücksetzen kann elektrisch und / oder logisch erfolgen, um das Kartenterminal und / oder Chipkarten (-Funktionseinheiten) in einen definierten initialen Zustand zu setzen.

Je nach adressierter Functional Unit, Kartenterminal oder Chipkartenkontaktiereinheit (Slot), muss die Ausführung des SICCT RESET CT / ICC Kommandos einen entsprechend unterschiedlichen Grundzustand herstellen. Der hauptsächliche Unterschied der Grundzustände ist

- die Deaktivierung aller Slots und Chipkarten im Wirkungsbereich bei Reset des Terminals,
- die (Re-)Aktivierung eines einzelnen Slots bzw. der enthaltenen Chipcard bei Adressierung eines einzelnen Kartenslots.

Das Kommando adressiert das Kartenterminal

Ein Reset des Kartenterminals muss allgemein ein Rücksetzen von Kartenterminalressourcen sowie eine Neuinitialisierung der adressierbaren Functional Units (FUs) bzw. deren interner Zustands- und Steuerlogik, der Protokolllogik, der Interaktions- und Anzeigeelemente wie z.B. Keypad, LCDisplay, Statusanzeigen (LED), Sensoren bedeuten.

Nach Ausführung des Kartenterminal-Resets müssen alle Chipkartenfunktionseinheiten bzw. im Wirkungsbereich befindlichen Chipkarten elektrisch / logisch deaktiviert sein. SICCT RESET CT darf keinen Einfluss auf eine bestehende CT Session oder den Status der Kommunikationsschicht zur externen Steuereinheit haben.

Anzeigeeinrichtung (LCDisplay)

Sofern eine Anzeigeeinrichtung (Display) vorhanden ist, muss nach Ausführung des SICCT RESET CT Kommandos die Idle Message des Kartenterminals im Display erscheinen.

Warm / Cold Reset

Das Kartenterminal darf die Anforderung nicht differenzieren, und muss immer einen Cold Reset ausführen, der alle Functional Units deaktiviert bzw. in deren Grundzustand überführen muss.

Übergabe eines Parameter-Request- / Rückgabe eine PPSR Response Datenobjekts (PPSR DO)

Adressiert das SICCT RESET CT Kommando mit PPSR DO das Terminal, müssen vorgegebene PPS-Parameter unberücksichtigt bleiben, das Kommando muss abgewiesen und mit der Rückgabe eines Fehlerstatus beendet werden. Es darf kein (leeres) PPSR DO zurückgegeben werden.

Rückgabe von Reset-Information

In Analogie zum Chipkarten-ATR (Answer To Reset) kann ein SICCT-Kartenterminal-Reset eine Reset-Antwort in Form eines Cardterminal Status Information Datenobjekts (CTS DO) generieren, welches optional (über den Parameter P2) angefordert werden kann.

Das Kommando adressiert eine (logische) Chipkartenkontaktiereinheit

Der Reset von Chipkartenkontaktiereinheiten, welche eine Chipkarte in ihrem Wirkungsbereich beinhalten, muss die Parametrierung der Chipkartenkontaktiereinheit sowie nachfolgend den elektrisch / logischen Reset der im Zugriff befindlichen Chipkarte(n) bedeuten. Nach erfolgreicher Ausführung muss eine Chipcard im Wirkungsbereich aktiviert (Reset Zustand) und betriebsbereit für den Empfang von Kommandos (APDUs) sein.

Ein, an einen Slot gerichtetes, SICCT RESET ICC kann alternativ zu SICCT REQUEST ICC zur Aktivierung einer im Wirkungsbereich befindlichen Chipcard verwendet werden.

Warm / Cold Reset

Das CT-Kommando kann (über den Parameter P2) wahlweise einen Warm - oder Cold - Reset an einer ICC – Funktionseinheit bewirken. Dieser erfolgt für kontaktbehaftete Karten nach ISO 7816-3 (ICC) oder nach ISO 14443 A / B für kontaktlosen Karten (RFID Token, PICC).

Während ein Kaltstart immer, d.h. auch bei deaktivierter Chipcard, erfolgen kann, darf ein Warmstart nur ausgeführt werden, sofern die Chipcard bereits aktiviert und betriebsbereit war.

Rückgabe von Reset-Information

Bei kontaktbehafteten Karten muss ein Chipkarten-Reset mit einer Answer-To-Reset-Information (ATR DO) und bei kontaktlosen Karten mit einem Answer-To-Select (ATS DO) beantwortet werden.

Automatisches und explizites PPS-Verfahren

Das SICCT-Kartenterminal muss generell die ATR / ATS - Informationen einer zurückgesetzten Chipkarte auswerten, um festzustellen, ob eine Protokoll Parameter Selection (PPS) erfolgen kann.

Eine PPS-Negotiation kann vom Terminal generell nicht durchgeführt werden, wenn

- der Reset der Chipcard nicht erfolgreich war,
- eine Chipcard den negotiable mode nicht durch den ATR bzw. ATS anzeigt (Speicherkarte nach ISO 7816-10 [STD9], ISO 14443 B, Prozessorkarte nach ISO 7816-3 [STD6] mit specific mode, unbekannter Kartentyp).

Im Gutfall muss das Terminal automatisch ein implizites Protocol Parameter Selection (PPS)-Verfahren nach ISO 7816-3 [STD6] durchführen, sofern kein PPSR DO übergeben oder PPS per Option in Parameter 'P2' untersagt wurde.

Hinsichtlich der Wahl der Kommunikationsparameter zur Chipkarte kann eine explizite Auswahl von Protokoll und Übertragungsdatenrate vorgegeben werden, sofern die PPS-Parameter mit dem Kommando übermittelt werden. Hierzu kann optional ein Datenobjekt (PPSR DO) mit dem PPS-TPDU zur Umschaltung des Chipkartenprotokolls und des Teilers zur Einstellung einer höheren Übertragungsrate an das Kartenterminal übergeben werden.

Übergabe eines Parameter-Request- / Rückgabe eines PPS Response Datenobjekts

Sofern ein SICCT RESET ICC Kommando mit PPSR DO ausgeführt wird, muss der PPS-Mechanismus des Kartenterminals diese Parameter unmittelbar für die Aushandlung der Protokollparameter (Protocol Negotiation) verwenden, nachdem die Karte per Cold- oder Warm-Reset zurückgesetzt und der ATR empfangen wurde.

Der Aufbau, die Struktur und Inhalt des PPS-Request (Value-Teil des PPSR DOs) sind spezifisch und müssen entsprechend der adressierten Kartentechnologie erfolgen:

- ISO 7816-3 [STD6] für Prozessorchipkarten
- ISO 14443 A für kontaktlose Karten / Token (PICC)

Sofern eine erfolgreiche Protokoll - Negotiation stattgefunden hat, und eine PPS-Response von der Chipcard fehlerfrei empfangen wurde, muss ein PPSR DO zurückgegeben werden, sofern der Aufrufer diese Information (über Parameter P2) angefordert hat.

Rückgabe des Status

Das CT-Kommando muss den Kommandostatus (SW1SW2) sowie auf Anforderung des Aufrufers die angeforderte Information (Datenobjekte) der zurückgesetzten Chipkartenkontaktiereinheit) oder des Kartenterminals liefern. Sofern ein Fehlerstatus angezeigt wird, dürfen keine Datenobjekte zurückgeliefert werden.

Ausführungsphasen SICCT RESET CT / ICC				
Phase		Stage	option	Description
preprocessing phase	Vorbereitungsphase	1		▪ check command APDU and optional data objects

processing phase	Ausführungsphase	2	- check CT / ICC / PICC state - Reset CT / ICC / PICC - optional: check ATR / ATS of ICC / PICC for negotiable mode - optional: Perform PPS with ICC / PICC - optional: check PPS response
postprocessing phase	Nachbereitungsphase	3	- optional: generate ICC / PICC PPS Response - optional: generate CT / ICC / PICC Reset Information - send return value <SW1SW2> and optional data object(s)

5.12.2 Anwendungsbedingungen

Ein Reset des Kartenterminals kann

- nach Auftreten einer Kommunikationsstörung zwischen Anwendungssystem und Kartenterminal gegeben werden, sofern die Applikation keine andere Möglichkeit zur Beseitigung der Störung hat.
- nach der Initialisierung des Kommunikationskanals zum Kartenterminal gegeben werden, um einen Grundzustand des gesamten Geräts herzustellen.
- vor Beendigung der Kartenterminalsession verwendet werden, um den Grundzustand des gesamten Geräts herzustellen.
- nicht zum Schließen einer bestehenden CT Session verwendet werden.

Der Reset einer Chipkarte kann unter folgenden Gesichtspunkten ausgeführt werden:

- Ein Reset zur Chipkarte kann vom Anwendungssystem veranlasst werden, wenn dies auf Anwendungsebene erforderlich ist (z.B. bei einer Kommunikationsstörung, zur Erkennung der Chipkartenpräsenz oder zur Typfeststellung).
- Optional können mit dem Resetvorgang der Chipkarte verbundene Betriebsparameter, sog. PPS-Parameter, für die Kommunikation zur Chipkarte festgelegt werden.

Abbruchbedingung

Das Kommando SICCT RESET CT/ ICC hat nur dann eine direkt erkennbare Auswirkung auf den Benutzer, sofern das Kartenterminal insgesamt zurückgesetzt wird, und Statuswechsel an der Mensch-Maschine-Schnittstelle entstehen. Die Ausführung kann durch einen Anwender des Kartenterminals per Entnahme einer adressierten Karte (Slot) aber nicht für das Terminal oder per Keypad beeinflusst oder abgebrochen werden.

Die Wirkung des Kommandos muss elementar ausfallen und kann durch die steuernde Entität in der Ausführung nicht abgebrochen werden.

SICCT RESET CT							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
BCS	SICCT				Stage		
					1	2	3
no	✓	✓	✓	no	-	-	-

5.12.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	[Lc]	[Data]	[Le]
SICCT RESET CT / ICC	'80'	'11'	Functional Unit	Command Qualifier	Length Command Data	Command Data	Length Requested Data
	- CLA = Class - INS = Instruction - P1, P2 = Parameter 1 and 2 - Lc = Length of command data field - Le = Length of expected - SW1, SW2 = Status Bytes				Case 1 (no cmd data, no rsp data): no Lc, no Le		
					Case 2 (no cmd data, rsp data): no Lc, Le=1-256 Bytes		
					Case 3 (cmd data, no rsp data): Lc=1-255 Bytes no Le		
					Case 4 (cmd data, rsp data): Lc=1-255 Bytes Le=1-256 Bytes		

Specification C-APDU		
CLA	'80'	Cardterminal Command Class
INS	'11'	SICCT RESET CT / ICC

P1	Functional Unit		
	bit8...bit1	Referenced Coding	
		'FF'	Escape: Signal Referenced Coding of P1 Functional Unit referenced by Functional Unit Index Data Object. FUI DO contained within Command Data Field.
	bit8...bit5	Direct Coding (mandatory)	
		'0'	Address contact bound chipcard interface or Cardterminal
		'1'	Address contactless chipcard interface
		'x'	x <> '0', '1': other values RFU
	bit4...bit1	'0'	bit 8 ... bit 5 = '0' Cardterminal (CT)
		'1'	bit 8 ... bit 5 = '0' 1 st ICC-Interface ... 14 th ICC-Interface
		:	
		'E'	bit 8 ... bit 5 = '1' 1 st CL-IF ... 14 th CL-IF

P2	Command Qualifier:		
	bit8	Operation to perform	
		In case P1 addresses the Cardterminal (CT)	
		0	Cold Reset of Cardterminal
		1	RFU
		In case P1 addresses an ICC, CL-IF	
		0	Cold Reset of ICC, PICC
		1	Warm Reset of ICC, PICC
	bit7...bit6	Protocol Parameter Selection Control	
		In case P1 addresses an ICC, CL-IF	
		00	Let the CT perform PPS automatically
		01	Perform PPS given by PPS-Data Object
		10	Do not perform PPS
		11	RFU
	bit5	Request ICC Interface Data Object(s)	
		In case P1 addresses an ICC , CL-IF	
		0	No requested data

	bit4...bit1	1	Request ICC Interface Status Data Object
		Request Reset Information	
		'0'	No requested data
		In case P1 addresses the Cardterminal:	
		'1'	Return complete Card Terminal Status Data Object
		'2'	Return Reduced Card Terminal Status Data Object
		In case P1 addresses an ICC , CL-IF	
		'1'	Request complete ATR Data Object(s)
		'2'	Request Historical Byte Data Object(s) (of ATR information)
		'4'	Request PPS Response Data Object

Lc	Length of Command Data Nc		
	Empty	Lc absent	no Command Data provided; Nc = 0
	variable length	Length of Command Data Nc (no. of bytes contained in Data field)	
		Lc short '01' <= Lc <= 'FF':	1 <= Nc <= 255

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	Empty	non-existent: in case Nc = '00': no Command Data provided
	PPSR DO	PPS-Request Data Object
	In case of Referenced Coding of 'P1'	
	FUI DO	Functional Unit Index Data Object
	PPSR DO	PPS-Request Data Object

Le	Length of Requested Data Ne Return up to Ne bytes of requested information		
	Empty	Le absent	No information requested Ne = 0
	variable length	Le short '01' <= Le <= 'FF'	1 <= Ne <= 255
		Le short Le = '00'	Ne = 256

5.12.4 Data Objects

Das SICCT RESET CT / ICC Kommando kann mit den folgenden Daten Objekten arbeiten.

Data Object	COMMAND RESPONSE	optional	Description	Remarks
FUI DO	CMD	✓	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10.9
ATR DO	RESP	✓	Answer To Reset Information of a Chipcard	see 5.5.10.4
HB DO	RESP	✓	Historical Byte Data Object of an ATR of a Chipcard	see 5.5.10.5
CTS DO	RESP	✓	Cardterminal Status Information	see 5.5.10.13
INTFS DO	RESP	✓	ICC Interface Status Data Object	see 5.5.10.14
PPSR DO	CMD	✓	PPS - Request Data Object	see 5.5.10.17
	RESP	✓	PPS - Response Data Object	

5.12.5 Response Structure

SICCT RESET CT / ICC	Kodierung R-APDU			
	[Body]		Trailer	
	[Requested Data / Information]		Status Byte 1	Status Byte 2
	Empty	in case no requested information	SW1	SW2
		in case invalid parameter 'P1' / 'P2'		
		in case Lc was invalid		
		in case Le was invalid or too small		
		in case of error		
	[Requested Information]	in case of valid command and error free operation		
	INTFS DO	ICC Interface Status Data Object		
	CTS DO	Card Terminal Status Data Object		
		Reduced Card Terminal Status Data Object		
	ATR DO	ATR Data Object(s)		
	HB DO	Historical Byte Data Object(s)		
	PPSR DO	PPS - Response Data Object		

5.12.6 Status-Codes SW1SW2

Nach erfolgreicher Ausführung des SICCT RESET CT / ICC -Kommandos muss das erste Byte des Statusworts SW1 den Wert '90' enthalten'.

Das Statusbyte SW2 kann weitere Umstände in Abhängigkeit zu Kommandoparametern qualifizieren. Dieser Wert SW2 muss eine Typerkennung der Chipkarte nach dem Reset zulassen:

- '00' synchrone Chipkarte (Speicherkarte)
- '01' asynchrone Chipkarte (Prozessorkarte)

SW1SW2	Addressed Functional Unit			Priority Priorität	Specification	Meaning / Bedeutung
	P1					
	Of Type					
	CT	ICC <n>	CL- IF <n>			
'6400'	x	-	-	12	Reset not successful	- Error occurred during cardterminal reset. - State of terminal and functional units possibly unchanged.
	-	x	x	12	Reset not successful	- Error occurred during chipcard reset. - In case - the ICC / PICC does not react on reset or - the ICC / PICC reacts on reset but does not send correct or incomplete reset information (ATR), or - the ICC / PICC does not provide any terminal supported protocol. - State of addressed chipcard set to deactivated state.
'64A1'	-	x	x	12	No Card present	Addressed ICC Interface / slot did not contain or contact a chipcard
'6700'	x	x	x	3	Wrong (command) length	Wrong Lc: Inconsistent command body or no Command Data field supported.
'6900'	x	x	x	9	Command not allowed.	- The terminal cannot perform the command because has there is no valid (open) - CT Session - TLS connection
'6941'	x	x	x	10	Functional Unit busy / not available	- At the moment FU_CT is busy / unavailable. - Funktionseinheit FU_CT ist z.Zt. belegt / nicht verfügbar
'6A00'	x	x	x	5	Wrong parameters P1, P2	- P1 addresses invalid Functional Unit. - In case of Referenced Coding – - Invalid / ungültiges FUI DO referenced by P1 (command data). - P2 specifies not supported value
'6A80'	x	x	x	6	Invalid Data Object	- Incorrect data objects / parameters in the command data field - In case of Referenced Coding – - The data field contains an invalid DO / - Das Datenfeld enthält ein unzulässiges DO. - Invalid FUI DO / Unzulässiges FUI DO
'6A88'	x	x	x	7	Missing Data Object	- In case of Referenced Coding – - The data field does not contain a FUI DO / - Das Datenfeld enthält kein FUI DO.
'6A89'	x	x	x	8	Too many Data Objects	- Too many Data Objects (FUI DO) of same type. - Es sind zu viele Datenobjekte gleichen Typs (FUI DO, ...) enthalten
'6C00'	x	x	x	4	Wrong Length Le	Wrong Le / falscher Wert Le
'6D00'	x	x	x	2	Wrong Instruction	Wrong / unbekanntes Instruction-Byte

'6E00'	x	x	x	1	Class not supported	Falsches / unbekanntes Class-Byte
'6F00'	x	-	-	11	Communication with CT not possible	In case - the terminal detects general protocol errors or - the terminal detects errors within CT Session or TLS connection - State of addressed chipcard set to deactivated state.
	-	x	x	11	Communication with ICC not possible	In case - the ICC / PICC reacts on reset but does not send correct or incomplete protocol parameter selection (PPS) information - the ICC / PICC reacts on reset but generates protocol errors.
'9000'	x	-	-	12	Reset successful	Cardterminal and all functional units successfully set to reset state. All chipcards deactivated.
		x		12	Reset successful, synchronous ICC	Synchronous chipcard detected, activated and successfully set to demanded reset state.
		x	x	12	Reset successful, asynchronous ICC	Asynchronous chipcard detected, activated and successfully set to demanded reset state

5.13 Command SICCT REQUEST ICC

5.13.1 Funktion

Das Kommando SICCT REQUEST ICC kann zur Anforderung und Aktivierung einer Chipkarte an einer adressierten Chipkartenkontakteinheit verwendet werden. Das Kommando muss stets einen Cold -Reset der Chipkarte bewirken und kann mit der Antwort Informationen über den Typ der Chipkarte sowie den generellen Status zurückliefern.

Für den Fall, dass sich keine aktivierte Chipkarte im Wirkungsbereich der adressierten Chipkartenfunktionseinheit befindet, kann optional der "Einführungs- und Aktivierungsprozess" gesteuert und überwacht werden. Für Kartenterminals mit Display kann eine Textmeldung als Eingabeaufforderung angezeigt werden. Ebenso kann optional die Zeit der Einführung bzw. Aktivierung überwacht werden. Nach der Einführung muss automatisch ein COLD-Reset sowie eine Aktivierung der angeforderten Chipkarte erfolgen. Das mit dem Einführungsprozess verbundene Rücksetzen der Chipkarte bzw. der Chipkartenfunktionseinheit muss elektrisch und / oder logisch erfolgen, um die Chipkarten - (Funktionseinheit) in einen definierten Initialzustand zu versetzen. Optional kann die Reset-Information der Chipkarte zurückgegeben werden.

Für den Fall, dass sich eine bereits zuvor aktivierte Chipkarte im Wirkungsbereich der Chipkartenfunktionseinheit befindet, darf das Kommando keinen COLD - Reset ausführen und keine Typinformation liefern. Der elektrisch / logische Zustand der Chipkarte muss unverändert bleiben. Ein gesonderter Fehlercode muss den Aufrufer über diese Situation informieren. Die ggf. im Terminal vorliegende Reset-Information kann angefordert werden.

Folgende Optionen können generell gewählt werden:

- Anzeige eines Bediendialogs: Bei Kartenterminals mit Display kann eine Standard- oder frei wählbare Eingabeaufforderung als Bediendialog angezeigt werden.
- Setzen eines optischen oder akustischen Signals: Bei Kartenterminals mit akustischem Tongeber oder optischer Anzeige (z.B. LED) kann ein Signal als Eingabeaufforderung ausgelöst werden.
- die Zeit für die Einführung und / oder Aktivierung der Chipkarte kann getrennt vorgegeben und überwacht werden;

Das Kommando muss nach Beendigung, d.h. nach erfolgtem Reset der Chipkarte, folgende Informationen zurückliefern können

- 1) Typ der aktivierten Chipkarte
(Synchrone / asynchrone Karte bzw. Speicher- oder Prozessorkarte),
- 2) Kennung der Chipkarte: Answer-To-Reset-String (Cold ATR) bzw. den Answer-To-Select-String (ATS) der aktivierten Chipkarte,
- 3) aktives Protokoll der aktivierten Chipkarte
- 4) logischer Betriebsmodus der aktivierten Chipkarte
(specific / negotiable).
- 5) elektrischer Status der aktivierten Chipkarte
(defekt, falsche Lage, nicht kommunikationsfähig, ...).

Der Reset von Chipkartenkontakteinheiten, welche eine Chipkarte in ihrem Wirkungsbereich beinhalten, muss die Parametrierung der Chipkartenkontakteinheit sowie nachfolgend den elektrisch / logischen Reset der im Zugriff befindlichen Chipkarte(n) bedeuten. Das CT-Kommando muss stets einen Cold - Reset an einer ICC – Funktionseinheit bewirken.

Ein erfolgreicher Chipkarten-Reset einer kontaktbehafteten oder kontaktlosen Karte muss mit einer Answer-To-Reset-Information (ATR DO) bzw. mit einem Answer-To-Select (ATS DO) beantwortet werden. Das SICCT-Kartenterminal muss generell die ATR / ATS - Informationen einer zurückgesetzten Chipkarte auswerten, und muss generell ein implizites Protocol Parameter Selection (PPS)-Verfahren nach [ISO/IEC 7816-3] durchführen.

Das CT-Kommando muss den Kommandostatus sowie auf Anforderung des Aufrufers die korrespondierende Reset-Information der zurückgesetzten Chipkartenkontakteinheit) oder des SICCT-Kartenterminals liefern.

Ausführungsphasen SICCT REQUEST ICC				
Phase		Stage	option	Description
preprocessing phase	Vorbereitungsphase	1		- display message - perform user dialog - wait for card insertion or user abort via keypad
processing phase	Ausführungsphase	2		- check CT / ICC / PICC state - Reset CT / ICC / PICC - Perform PPS with ICC / PICC
postprocessing phase	Nachbereitungsphase	3		- return ICC / PICC PPS Response - return CT / ICC / PICC Reset Information - send return value

5.13.2 Anwendungsbedingungen

Das Kommando kann generell die Kommunikation und den Beginn einer Session zu einer angeforderten Chipkarte eröffnen.

Eine Sitzung zu einer Chipkarte muss stets mit einem COLD-Reset der Chipkarte und der Erkennung des Cold-ATR beginnen. Das SICCT-Kartenterminal muss mit dem Cold-ATR die von der Chipkarte angezeigten Kommunikationsparameter und Optionen empfangen und muss ein automatisches PPS-Verfahren zur Einstellung von Kommunikationsparametern (Protokollwahl, Datenübertragungsrate) durchführen. Um weitere Kommunikationsparameter per PPS wirksam zu machen oder die Chipkarte über einen Warm-Reset (nach ISO 7816-3 [STD6]) in eine andere Betriebsart versetzen zu können, muss nach einem SICCT REQUEST ICC mit dem SICCT RESET CT / ICC Kommando gearbeitet werden. Anderenfalls müssen die, mittels des SICCT-Kartenterminals aus dem Cold-Reset-verfahren gewählten, Betriebsparameter bestehen bleiben.

Das Kommando dient primär zur Anforderung einer Chipkarte. Die Ausgabe einer Display-Nachricht muss entfallen, wenn der adressierte Slot bereits eine gesteckte Karte beinhaltet.

Abbruchbedingung

Das Kommando SICCT REQUEST ICC kann nur dann eine direkt erkennbare Auswirkung auf den Benutzer haben, sofern Statuswechsel an der Mensch-Maschine-Schnittstelle entstehen. Die Anforderung, dem System eine Karte zuzuführen, kann durch eine entsprechende Interaktion des Anwenders des SICCT-Kartenterminals beeinflusst werden. Diese Interaktionen sind: Zuführen der Karte, Betätigen der Abbruchtaste am Keypad des SICCT-Kartenterminals.

Sobald eine angeforderte Karte zugeführt wurde, muss diese zurückgesetzt werden. Dieser Vorgang kann nicht durch den Anwender beeinflusst werden.

Eine steuernde Entität kann das Kommando während der Anforderungsphase abbrechen. Nach erfolgtem Abbruch muss die Kontaktiereinheit elektrisch abgeschaltet sein.

SICCT REQUEST ICC							
CT Mode		Conditions					
BCS	SICCT	CT ADMIN Session	CT CONTROL Session		Abortable by SICCT CONTROL COMMAND		
					Stage		
					1	2	3
no	✓	✓	✓	✓	✓	no	no

5.13.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	[Lc]	[Data]	[Le]
SICCT REQUEST ICC	'80'	'12'	Functional Unit	Command Qualifier	Length Command Data	Command Data	Length Requested Data

	- CLA = Class - INS = Instruction - P1, P2 = Parameter 1 and 2 - Lc = Length of command data field - Le = Length of expected - SW1, SW2 = Status Bytes	Case 1 (no cmd data, no rsp data): no Lc, no Le
		Case 2 (no cmd data, rsp data): no Lc, Le=1-256 bytes
		Case 3 (cmd data, no rsp data): Lc=1-255 Bytes no Le
		Case 4 (cmd data, rsp data): Lc=1-255Bytes Le=1-256 Bytes

Specification C-APDU		
CLA	'80'	Cardterminal Command Class
INS	'12'	SICCT REQUEST ICC

P1	Functional Unit		
	bit8...bit1	Referenced Coding	
		'FF'	Escape: Signal Referenced Coding of P1 Functional Unit(s) referenced by Functional Unit Index Data Object (FUI DO) or Functional Unit Context Data Object (FU CON DO). Either one FUI DO or one FU CON DO contained within Command Data Field.
	bit8...bit5	Direct Coding (mandatory)	
		'0'	Address contact bound chipcard interface
		'1'	Address contactless chipcard interface (CL-IF)
		'x'	x <> '0', '1': other values RFU
	bit4 ... bit1	'1'	bit 8 ... bit 5 = '0' 1 st ICC-Interface ... 14 th ICC-Interface
		:	
		'E'	Bit 8 ... bit 5 = '1' 1 st CL-IF ... 14 th CL-IF

P2	Command Qualifier:		
	bit8...bit5	Request handling Instructions	
		SICCT with no display: don't care	
		SICCT with display	
		'0'	Display standard message or message found in data field according to Application Label DO
		'F'	No message to be displayed ⁶
	Option Setting		
	bit4	'0'	Acoustic Signal: None
		'1'	Acoustic Signal: Give Acoustic Signal
	bit3	'0'	Optical Signal: None
		'1'	Optical Signal: Show Optical Signal
	bit2...bit1	Request Reset Information	
		'0'	No requested data
		'1'	Request complete ATR Data Object(s) (Cold ATR)
		'2'	Request Historical Byte Data Object(s)

	Length of Command Data Nc
--	---------------------------

⁶ Suppress all internal ('Abort', ...) and external messages.

Lc	Empty	non-existent: no Command Data provided; interpret Lc = '0'	
	variable length	Length of Command Data (no. of bytes contained in Data field)	
		Lc short '01' <= Lc <= 'FF'	1 <= Nc <= 255

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	Empty	non-existent: in case Nc = '0': no Command Data provided
	APPL DO	Application Label Data Object
	SMTBD DO	SICCT Message-To-Be-Displayed Data Object
	WT T DO	Waiting Time Data Object
	In case of Referenced Coding of 'P1'	
	FUI DO	Functional Unit Index Data Object
	FU CON DO	Functional Unit Context Data Object
	APPL DO	Application Label Data Object
	SMTBD DO	SICCT Message-To-Be-Displayed Data Object
	WT DO	Waiting Time Data Object

Le	Length of Requested Data Ne		
	Return up to Ne bytes of requested information		
	Empty	No information requested	
	variable length	Le short '01' <= Le <= 'FF'	1 <= Ne <= 255
		Le short Le = '00'	Ne = 256

5.13.4 Data Objects

Das SICCT REQUEST ICC Kommando kann mit den folgenden Daten Objekten arbeiten.

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10.9
FU CON DO	CMD	Functional Unit Context Data Object	see 5.5.10.10
APPL DO	CMD	Text / display message.	see 5.5.10.18
SMTBD DO	CMD	Constructed TLV-DO containing one Character Set and one Application Label DO.	see 5.5.10.20
ATR DO	RESP	Answer To Reset Information (Cold ATR)	see 5.5.10.4
HB DO	RESP	Part of ATR Information: Historical Byte Data Object	see 5.5.10.5
WT DO	CMD	Max. Waiting Time in seconds	see 5.5.10.21

Erlaubte Kombinationen aus P1 und Data Objects			
P1	FU CON DO	FUI DO	Bemerkung
Direct	No	No	P1 adressiert direkt den Slot. Implizit ist die Anzeigeeinheit (Display) und das Standard-Keypad (für Benutzer-Abbruch-Aktion) selektiert.
Referenced	No	Yes	P1 - Referenced Coding: P1 adressiert den Slot per FUI DO. Implizit ist das Standard-Keypad (für Benutzer-Abbruch-Aktion) selektiert.
Referenced	Yes	No	P1 - Referenced Coding: P1 adressiert den Slot, die Anzeigeeinheit (Display) und das Keypad (für Benutzer-Abbruch-Aktion) innerhalb des FU CON DOs.

Anmerkung:
Weitere Permutationen müssen jeweils einen Fehlerfall darstellen, in denen DO(e) fehlen oder zuviel übergeben werden.
Als Fehlerfälle müssen auch FU CON DO interpretiert werden, die keinen zulässigen oder eindeutigen Kontext für SICCT REQUEST ICC (min. 2 FUs: min. einen Slot, optional eine Anzeigeeinheit (Display) oder ein Keypad ergeben.)

5.13.5 Response Structure

SICCT REQUEST ICC	Kodierung R-APDU			
	[Body]			Trailer
	[Requested Data / Information]			Status Byte 1 Status Byte 2
	Empty	in case no requested information	SW1	SW2
		in case invalid parameter 'P1' / 'P2'		
		in case Lc was invalid		
		in case Le was invalid or too small		
		in case of error		
	Requested Information	in case of valid command and error free operation		

5.13.6 Status-Codes SW1SW2

Nach erfolgreicher Ausführung des SICCT REQUEST ICC Kommandos kann das erste Byte des Statusworts SW1 den Wert '90' enthalten.

Das Statusbyte SW2 kann weitere Umstände in Abhängigkeit zu Kommandoparametern qualifizieren.

Dieser Wert Sw2 muss eine Typerkennung der Chipkarte nach dem Reset zulassen:

- '00' synchrone Chipkarte (Speicherkarte)
- '01' asynchrone Chipkarte (Prozessorkarte)

SW1SW2	Addressed		Priority Priorität	Specification	Meaning / Bedeutung
	Functional				
	Unit				
	P1				
	Of Type				
	ICC	CL- IF			
	<n>	<n>			
'6200'	x	x	11	Warning: No card presented in time	No card presented within specified time. / Innerhalb der vorgegebenen Zeit wurde Keine Karte eingeführt
'6201'	x	x	11	Warning: Reset successful	Card (ICC) already present and activated. / Karte (ICC) war bereits eingeführt und aktiviert.
'6400'	x	x	12	Reset not succes- sful	Error occurred during chipcard reset in case - the ICC / PICC does not react on reset or - the ICC / PICC reacts on reset but does not send correct or incomplete reset information (ATR), or - the ICC / PICC does not provide any terminal supported protocol. - State of addressed chipcard set to deactivated state.
'6401'	x	x	12	Process aborted by pressing of CANCEL key	Process aborted by pressing of CANCEL key. / Der Prozeß via ABBRUCH-Taste abgebrochen
'64A1'	x	x	12	No Card present	Addressed ICC Interface / slot did not contain or contact a chipcard / RFID token.
'64A2'	x	x	11	Card not activated	"Warm reset or PPS procedure with given PPSR DO not processed because card not activated"
'6700'	x	x	3	Wrong (command) length	Wrong Lc: Inconsistent command body or no Command Data field supported.
'6900'	x	x	9	Command not al- lowed.	The terminal cannot perform the command because has there is no valid (open) - CT Session - TLS connection
'6941'	x	x	10	Functional Unit busy / not availa- ble	At the moment FU_CT is busy / unavailable. Funktionseinheit FU_CT ist z.Zt. belegt / nicht verfügbar
'6942'	x	x	8	Selected Charac- ter Set not sup- ported.	The addressed display does not support the sel- ected character set. / Die adressierte Anzeige un- terstützt den gewählten Zeichensatz nicht.
'6A00'	x	x	5	Wrong parameters P1, P2	- P1 addresses invalid Functional Unit. - In case of Referenced Coding – - Invalid / ungültiges FUI DO referenced by P1 (command data). - P2 specifies not supported value
'6A80'	x	x	6	Invalid Data Ob- ject	- Incorrect data objects / parameters in the com- mand data field - In case of Referenced Coding – - The data field contains an invalid DO / - Das Datenfeld enthält ein unzulässiges DO. - Invalid FUI DO / Unzulässiges FUI DO
'6A88'	x	x	7	Missing Data Ob- ject	- In case of Referenced Coding – - The data field does not contain a FUI DO / - Das Datenfeld enthält kein FUI DO.

'6A89'	x	x	8	Too many Data Objects	- Too many Data Objects (FUI DO, ...) of same type. - Es sind zu viele Datenobjekte gleichen Typs (FUI DO, ...) enthalten
'6C00'	x	x	4	Wrong Length Le	Wrong Le / falscher Wert Le:
'6D00'	x	x	2	Wrong Instruction	Wrong / unbekanntes Instruction-Byte
'6E00'	x	x	1	Class not supported	Falsches / unbekanntes Class-Byte
'6F00'	x	x	11	Communication with ICC not possible	- the ICC / PICC reacts on reset but does not send correct or incomplete protocol parameter selection (PPS) information - the ICC / PICC reacts on reset but generates protocol errors.
'9000'	x	x	11	Reset successful, synchronous ICC	Synchronous chipcard detected, activated and successfully set to demanded reset state.
'9001'	x	x	12	Reset successful, asynchronous ICC	Asynchronous chipcard detected, activated and successfully set to demanded reset state.

5.14 Command SICCT EJECT ICC

5.14.1 Funktion

Das Kommando SICCT EJECT ICC kann zur elektrisch / logischen Deaktivierung einer adressierten Chipkartenkartenfunktionseinheit bzw. einer Chipkarte verwendet werden und die Ausführung optionaler Zusatzfunktionen veranlassen.

Ausführungsphasen SICCT EJECT ICC				
Phase		Stage	option	Description
preprocessing phase	Vorbereitungsphase	1		
processing phase	Ausführungsphase	2		- check CT / ICC / PICC state - deactivate CT / ICC / PICC
postprocessing phase	Nachbereitungsphase	3		- display message - perform user dialog - wait for card removal - send return value

5.14.2 Anwendungsbedingungen

Das Kommando muss generell die Kommunikation beenden und das Ende einer Session zu einer Chipkarte markieren, indem die Chipkarte elektrisch / logisch deaktiviert werden muss.

Folgende Optionen können generell gewählt werden:

- Setzen eines akustischen Signals
- Setzen eines optischen Signals
(Bei SICCT-Kartenterminals mit Display kann eine Chipkarten-Entnahmeanforderung angezeigt werden),
- bei vorhandener mechanischer Funktion: Auswurf der Chipkarte
- Setzen eines Zeitgebers (Timers) zur opt. Überwachung der Entnahme.

Das Kommando muss primär zur Deaktivierung und Entnahme einer Chipkarte dienen. Die Ausgabe einer Display-Nachricht muss entfallen, wenn der adressierte Slot bereits keine gesteckte Karte aufweist.

Das Kommando muss mit der Antwort Informationen über den Zustand der Kontaktiereinheit bzw. der Chipkarte zurückliefern können.

Abbruchbedingungen

Die Anforderung vom System, eine Karte zu deaktivieren, kann nicht durch eine entsprechende Interaktion des Anwenders des SICCT-Kartenterminals beeinflusst werden.

Diese Interaktionen zur Entnahme kann von einem Benutzer per Betätigung der Abbruchtaste am Keypad des SICCT-Kartenterminals abgebrochen werden.

SICCT EJECT ICC							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
BCS	SICCT				Stage		
					1	2	3
no	✓	✓	✓	✓	no	no	✓

5.14.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	[Lc]	[Data]	[Le]
SICCT EJECT ICC	'80'	'15'	Functional Unit	Command Qualifier	Length Command Data	Command Data	Length Requested Data
	▪ CLA = Class ▪ INS = Instruction ▪ P1, P2 = Parameter 1 and 2 ▪ Lc = Length of command data field ▪ Le = Length of expected ▪ SW1, SW2 = Status Bytes				Case 1 (no cmd data, no rsp data): no Lc, no Le		
					Case 3 (cmd data, no rsp data): Lc=1-255 Bytes no Le		

Specification C-APDU				
CLA	'80'	Cardterminal Command Class		
INS	'15'	SICCT EJECT ICC		
P1	Functional Unit			
	bit8...bit1	Referenced Coding		
		'FF'	Escape: Signal Referenced Coding of P1 Functional Unit(s) referenced by Functional Unit Index Data Object (FUI DO) or Functional Unit Context Data Object (FU CON DO). Either one FUI DO or one FU CON DO contained within Command Data Field.	
	bit8...bit5	Direct Coding (mandatory)		
		'0'	Address contact bound chipcard	
		'1'	Address contactless chipcard interface (CL-IF)	
		'x'	x <> '0', '1': other values RFU	
	bit4...bit1	'1'	bit 8 ... bit 5 = '0' 1 st ICC-Interface ... 14 th ICC-Interface	
		:		
		'E'	bit 8 ... bit 5 = '1' 1 st CL-IF ... 14 th CL-IF	

P2	Command Qualifier:		
	bit8...bit5	Eject Handling Instructions	
		SICCT with no display: don't care	
		SICCT with display	
		'0'	display standard message or message found in data field according to Application Label DO
		'F'	no message to be displayed ⁷
	Option Setting		
	bit1	'0'	(RFU)
	bit2	'0'	Delivery: Mechanical Throw out
		'1'	Delivery: No Mechanical Throw out - Keep chipcard
	bit3	'0'	Optical Signal: None
		'1'	Optical Signal: Show Optical Signal
	bit4	'0'	Acoustic Signal: None
		'1'	Acoustic Signal: Give Acoustic Signal

⁷ Suppress all internal ('Abort', ...) and external messages.

Lc	Length of Command Data Nc		
	Empty	Lc absent	no Command Data provided; Nc = 0
	variable length	Length of Command Data (no. of bytes contained in Data field)	
		Lc short '01' <= Lc <= 'FF'	1 <= Nc <= 255

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	Empty	non-existent: in case Nc = '0': no Command Data provided
	APPL DO	Application Label Data Object
	SMTBD DO	SICCT Message-To-Be-Displayed Data Object
	WT DO	Waiting Time Data Object
	In case of Referenced Coding of 'P1'	
	FUI DO	Functional Unit Index Data Object
	FU CON DO	Functional Unit Context Data Object
	APPL DO	Application Label Data Object
	SMTBD DO	SICCT Message-To-Be-Displayed Data Object
	WT DO	Waiting Time Data Object

Le	Length of Requested Data Ne	
	Return up to Ne bytes of requested information	
	Empty	No information requested

5.14.4 Data Objects

Das SICCT EJECT ICC Kommando arbeitet mit den folgenden Daten Objekten.

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10.9
FU CON DO	CMD	Functional Unit Context Data Object	see 5.5.10.10
APPL DO	CMD	Text / display message.	see 5.5.10.18
SMTBD DO	CMD	Constructed TLV-DO containing one Character Set and one Application Label DO.	see 5.5.10.20
WT DO	CMD	Max. Waiting Time in seconds	see 5.5.10.21

P1	FU CON DO	FUI DO	Bemerkung
Direct	No	No	P1 adressiert direkt den Slot. Implizit ist die Anzeigeeinheit. (Display) und das Standard-Keypad (für Benutzer-Abbruch-Aktion) selektiert.
Referenced	No	Yes	P1 - Referenced Coding: P1 adressiert den Slot per FUI DO. Implizit ist das Standard-Keypad (für Benutzer-Abbruch-Aktion) selektiert.
Referenced	Yes	No	P1 - Referenced Coding: P1 adressiert den Slot, die Anzeigeeinheit (Display) und das Keypad (für Benutzer-Abbruch-Aktion) innerhalb des FU CON DOs.
Anmerkung: - Weitere Permutationen müssen jeweils einen Fehlerfall darstellen, in denen DO(e) fehlen oder zuviel übergeben werden. Als Fehlerfälle müssen auch FU CON DO bewertet werden, die keinen zulässigen oder eindeutigen Kontext für SICCT EJECT ICC (min. 2 FUs: min. einen Slot, optional eine Anzeigeeinheit (Display) oder ein Keypad ergeben).			

5.14.5 Response Structure

SICCT EJECT ICC	Kodierung R-APDU				
	[Body]			Trailer	
	[Requested Data / Information]			Status Byte 1	Status Byte 2
	Empty	no requested information		SW1	SW2
		in case invalid parameter 'P1' / 'P2'			
		in case Lc was invalid			
		in case Le was invalid or too small			
in case of error					

5.14.6 Status-Codes SW1SW2

Nach erfolgreicher Ausführung des SICCT EJECT ICC Kommandos enthält das erste Byte des Statusworts SW1 den Wert '90'. Bei Anweisung „No Mechanical Throw out - Keep chipcard“ (P2 bit2=1) ist immer, auch unabhängig von der Art der Kartenkontaktiereinheit (auch bei Kontaktiereinheiten mit Auswurfmechanik), von einer erfolgreichen Ausführung des Kommandos auszugehen. SW1 enthält daher bei „No Mechanical Throw out - Keep chipcard“ immer den Wert '90'. Auch bei Anweisung „Mechanical Throw out“ (P2 bit2=0) enthält SW1 den Wert '90', wenn die Karte gezogen/ausgeworfen wurde.

Nur bei Anweisung „Mechanical Throw out“ (P2 bit2=0) enthält SW1 den Wert '62' (Warning), wenn die Karte gesteckt bleibt.

Das Statusbyte SW2 qualifiziert bei erfolgreicher Ausführung weitere Umstände in Abhängigkeit zu Kommandoparametern.

Dieser Wert SW2 lässt eine Statuserkennung der Chipkartenfunktionseinheit nach dem Deaktivieren zu:

- '01' Command Successful, card removed

SW1SW2	Addressed Functional Unit		Priority Priorität	Specification	Meaning / Bedeutung
	P1				
	Of Type				
	ICC <n>	CL-IF <n>			
'6200'	x	x	10	Warning: Card not removed within specified time	▪ Chipcard (ICC) deactivated but not removed (within specified time). / Die Karte (ICC) wurde nicht innerhalb der vorgegebenen Zeit entnommen.
'6400'	x	x	12	Execution Error	▪ An undetermined error occurred during execution of the command. / Unbestimmter Fehler während der Kommandoausführung.
'6401'	x	x	12	Process aborted by pressing of CANCEL key	▪ Process aborted by pressing of CANCEL key. / Der Prozeß via ABBRUCH-Taste abgebrochen.
'6700'	x	x	3	Wrong (command) length	▪ Wrong Lc: Inconsistent command body or no Command Data field supported.

SW1SW2	Addressed Functional Unit		Priority Priorität	Specification	Meaning / Bedeutung
	P1				
	Of Type				
	ICC <n>	CL-IF <n>			
'6900'	x	x	9	Command not allowed.	▪ The terminal cannot perform the command because there is no valid (open) ▪ CT Session ▪ TLS connection
'6930'	x	x	8	Command with timer not supported.	▪ Terminal does not support the timer option. / Das Terminal unterstützt die Option Zeitüberwachung nicht.
'6940'	x	x	8	Command with Display not supported.	▪ Terminal does not support the display option. / Das Kommando unterstützt die Option Display (Anzeige) nicht.
'6941'	x	x	10	Functional Unit (FU) busy / not available.	▪ The addressed FU (FU_CT, display, slot, ...) is busy at the moment or not available. / Die adressierte Funktionseinheit (FU_CT, Anzeige, Konaktiereinheit, ...) ist im Moment belegt und ist nicht verfügbar.
'6942'	x	x	8	Selected Character Set not supported.	▪ The addressed display does not support the selected character set. / Die adressierte Anzeige unterstützt den gewählten Zeichensatz nicht.
'6A00'	x	x	5	Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ In case of Referenced Coding – Invalid / ungültiges FUI DO referenced by P1 (command data). ▪ P2 specifies not supported value
'6A80'	x	x	6	Invalid Data Object	▪ Incorrect data objects / parameters in the command data field ▪ In case of Referenced Coding – The data field contains an invalid DO / Das Datenfeld enthält ein unzulässiges DO. ▪ Invalid FUI DO / Unzulässiges FUI DO
'6A88'	x	x	7	Missing Data Object	▪ In case of Referenced Coding – The data field does not contain a FUI DO / Das Datenfeld enthält kein FUI DO.
'6A89'	x	x	8	Too many Data Objects	▪ Too many Data Objects (FUI DO, ...) of same type. ▪ Es sind zu viele Datenobjekte gleichen Typs (FUI DO, ...) enthalten
'6C00'	x	x	4	Wrong Length Le	▪ Wrong Le / falscher Wert Le: ▪ Le not allowed / Le nicht erlaubt.
'6D00'	x	x	2	Wrong Instruction	▪ Wrong / unbekanntes Instruction-Byte
'6E00'	x	x	1	Class not supported	▪ Falsches / unbekanntes Class-Byte
'6F00'	x	x	11	Communication with CT not possible	▪ In case ▪ the terminal detects general protocol errors or the terminal detects errors within CT Session or TLS connection

SW1SW2	Addressed Functional Unit		Priority Priorität	Specification	Meaning / Bedeutung
	P1				
	Of Type				
	ICC <n>	CL-IF <n>			
'9000'	x	x	12	Command successful	▪ Chipcard deactivated but no information if card has been removed (within specified time) / Chipkarte (ICC) deaktiviert aber keine Information, ob die Karte (in der spezifizierten Zeitspanne) entnommen wurde.
'9001'	x	x	12	Command successful	▪ Chipcard (ICC) deactivated and removed (within specified time). / Chipkarte deaktiviert und (in der spezifizierten Zeitspanne) entnommen.

5.15 Command SICCT GET STATUS

5.15.1 Funktion

Das SICCT GET STATUS - Kommando muss die Adressierung eines GET- Interfaces des SICCT-Kartenterminals bzw. dessen Funktionseinheiten ermöglichen, um Statusinformation abfragen zu können. Je nach Funktionseinheit und Typus des Datenobjekts können entsprechende Informationen angefordert werden.

Über folgende Datenobjekte können z.B. folgende Inhalte abgefragt werden:

Data Object	Description
CardTerminal Manufacturer Data Object	Query common Vendor or device information ▪ Vendor ID ▪ SW-Version, ▪ (optional) Serial Number, ▪ (optional) Approval Number,
CardTerminal Status Data Object	Query ▪ Functional Unit Data Object ▪ ICC Status Data Object ▪ optional - if present: CL-IF Status Data Object ▪ Maximum Length of SICCT Commands – and Response APDUs
Functional Unit Data Object	Query ▪ Presence, ▪ Quantity ▪ State ▪ Availability of Functional Units.
Functional Unit Name Data Object	Query ▪ Friendly Name of an Functional Unit
ICC Status Data Object	Query ▪ Presence of an ICC ▪ Electrical State of an ICC
CL-IF Status Data Object	Query ▪ Presence of a PICC / RFID Token ▪ State of a PICC / RFID-Token
Interface Capabilities Data Object	Query ▪ Physical attributes of an ICC Interface
Display Capabilities Data Object	Query ▪ Physical attributes of a FU of type Display
Keypad Capabilities Data Objects (KPC DO)	Query ▪ Physical attributes of a FU of type Keypad

Das Kommando muss mit der Antwort den Ausgang der Operation sowie optional angeforderte Datenobjekte zurückliefern.

Ausführungsphasen SICCT GET STATUS				
Phase		Stage	option	Description
preprocessing phase	Vorbereitungsphase	1		
processing phase	Ausführungsphase	2		▪ check corresponding state or data object
postprocessing phase	Nachbereitungsphase	3		▪ return corresponding state or data object ▪ send return value

5.15.2 Anwendungsbedingungen

Das Auslesen von Datenobjekten über die Kommandoschnittstelle kann nur in Abhängigkeit der Bedeutung des jeweiligen Datenobjekts und der aktiven Zugriffsrechte geschehen. Zurzeit sind in der Definition des SICCT-Standards die Zugriffsmöglichkeiten und Zugriffsberechtigungen für diejenigen Datenobjekte definiert, für welche die SICCT-Working Group Anwendungsfälle festgelegt hat.

Abbruchbedingung

Die Ausführung des Kommandos SICCT GET STATUS kann durch einen Anwender des SICCT-Kartenterminals weder beeinflusst noch abgebrochen werden.

Die Wirkung des Kommandos muss elementar sein und kann durch die steuernde Entität in der Ausführung nicht abgebrochen werden.

SICCT GET STATUS							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
BCS	SICCT			Stage			
					1	2	3
no	✓	✓	✓	no	no	no	no

5.15.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	[Lc]	[Data]	Le
SICCT GET STATUS	'80'	'13'	Functional Unit	Command Qualifier	Length Command Data	Command Data	Length Re- quested Data
	- CLA = Class - INS = Instruction - P1, P2 = Parameter 1 and 2 - Lc = Length of command data field - Le = Length of expected - SW1, SW2 = Status Bytes				Case 2 (no cmd data, rsp data): no Lc, Le=1-256 Bytes or extended Le		
					Case 4 (cmd data, rsp data): Lc=1-255 Bytes, Le=1-256 Bytes or extended Le		

Specification C-APDU			
CLA	'80'	Cardterminal Command Class	
INS	'13'	SICCT GET STATUS	
P1	Functional Unit		
	bit8...bit1	Referenced Coding (optional)	
		'FF'	Escape: Signal Referenced Coding of P1 Functional Unit referenced by Functional Unit Index Data Object. FUI DO contained within Command Data Field.
	bit8...bit5	Direct Coding (mandatory)	
		'0'	Address contact bound chipcard interface or cardterminal
		'1'	Address contactless chipcard interface (CL-IF)
		'4'	Address FU of type Display Unit
		'5'	Address FU of type Keypad Unit
		'6'	Address the Printer Port
		'7'	Address FU of type Biometric Sensor Unit

	bit4...bit1	'0'	bit8 ... bit5 = '0'	Cardterminal (CT)
			bit8 ... bit5 = '4'	Standard Display
			bit8 ... bit5 = '5'	Standard Keypad
			bit8 ... bit5 = '6'	Standard Printer Port
			bit8 ... bit5 = '7'	Standard Biometric Sensor Unit: Fingerprint
		'1' ... 'E'	bit8 ... bit5 = '0'	1 st ICC-Interface ... 14 th ICC-Interface
			bit8 ... bit5 = '1'	1 st CL-IF ... 14 th CL-IF
			bit8 ... bit5 = '4'	2 nd ... 14 th Display
			bit8 ... bit5 = '5'	2 nd ... 14 th Keypad
			bit8 ... bit5 = '6'	RFU
			bit8 ... bit5 = '7'	Specific Biometric Sensor Unit Type

P2	Command Qualifier: indicates requested Data Object		
	bit8 ... bit1	Referenced Coding	
		'FF'	Escape: Signal Referenced Coding of 'P2': Requested Data Object contained within Command Data Field: An empty Data Object (Tag and L=0) is given within the Command Data Field e.g. 'A1' '00' for FU NAME DO.
	bit8 ... bit1	Direct Coding	
		Data Objects served by all Functional Units	
		'A1'	Functional Unit Name Data Object
		In case P1 addresses the Cardterminal (CT):	
		'46'	CardTerminal Manufacturer Data Object
		'63'	CardTerminal Status Data Object
		'80'	ICC Status Data Object (all ICC Interfaces)
		'81'	Functional Unit Data Object
		In case P1 addresses an ICC	
		'80'	ICC Status Data Object (ICC Interface as addressed by 'P1')
		'66'	Interface Capabilities Data Object
		In case P1 addresses an CL-IF	
		'83'	CL-IF Status Data Object
		'66'	Interface Capabilities Data Object
		In case P1 addresses a Display FU	
		'85'	Character Set data Object
		'67'	Display Capabilities Data Object
		In case P1 addresses a Keypad FU	
		'6A'	Keypad Capabilities Data Object

Lc	Length of Command Data Nc		
	Empty	Lc absent	no Command Data provided; Nc = 0
	variable length	Length of Command Data Nc (no. of bytes contained in Data field)	
		Lc short '01' <= Lc <= 'FF'	1 <= Nc <= 255
		Lc extended 3 Byte Coding '000001' <= Lc <= '00FFFF'	1 <= Nc <= 65535

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	Empty	non-existent: in case Nc = '00': no Command Data provided
	One Requested Data Object	In case 'P2' set to 'FF':
		available data objects: see 5.15.4
	In case of Referenced Coding of 'P1'	
	FUI DO	Functional Unit Index Data Object
	One Requested Data Object	In case 'P2' set to 'FF':
		available data objects: see 5.15.4

Le	Length of Requested Data Ne		
	Return up to <Ne> bytes of requested information		
	variable length	Le short '01' <= Le <= 'FF'	1 <= Ne <= 255
		Le short Le = '00'	Ne = 256
		Le extended Lc extended 2 Byte Coding '0001' <= Le <= 'FFFF'	1 <= Ne <= 65535
		Le extended Lc extended 2 Byte Coding Le <= '0000'	Ne = 65536
		Le extended Lc absent or Lc short 3 Byte Coding '000001' <= Le <= '00FFFF'	1 <= Ne <= 65535
		Le extended Lc absent or Lc short 3 Byte Coding Le <= '000000'	Ne = 65536

5.15.4 Data Objects

Das SICCT GET STATUS Kommando kann mit den folgenden Daten Objekten arbeiten. Zur Zeit sind in der Definition des SICCT-Standards noch nicht alle Datenobjekte, Zugriffsmöglichkeiten und Zugriffsberechtigungen definiert.

Anmerkung: Künftige Spezifikationstände des SICCT-Standards könnten um weitere Datenobjekte, Zugriffsmöglichkeiten und Zugriffsberechtigungen erweitert werden.

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10.9
FU NAME DO	CMD RESP	Functional Unit Name Data Object	see 5.5.10.11
CTM DO	CMD RESP	Card Terminal Manufacturer Data Object	see 5.5.10.6
CTS DO	CMD RESP	Card Terminal Status Data Object	see 5.5.10.13
FU DO	CMD RESP	Functional Unit Data Object	see 5.5.10.8
ICCS DO	CMD RESP	ICC Status Data	see 5.5.10.7
CLIF DO	CMD RESP	CL-IF Status	see 5.5.10.12
DSPLC DO	CMD RESP	Display Capabilities Data Object	see 5.5.10.16
KPC DO	CMD RESP	Keypad Capabilities Data Object	see 5.5.10.28
CS DO	CMD RESP	Character Set Data Object	see 5.5.10.19
INTFC DO	CMD RESP	ICC Interface Capabilities Data Object	see 5.5.10.15

5.15.5 Response Structure

SICCT GET STATUS	Kodierung R-APDU			
	[Body]			Trailer
	[Requested Data / Information]			Status Byte 1 Status Byte 2
	Empty	in case no requested information	SW1	SW2
		in case invalid parameter 'P1' / 'P2'		
		in case Lc was invalid		
		in case Le was invalid or too small		
		in case of error		
	Requested Information	in case of valid command and error free operation		

5.15.6 Status-Codes SW1SW2

Nach erfolgreicher Ausführung des SICCT GET STATUS - Kommandos muss das erste Byte des Statusworts SW1 den Wert '90' enthalten.

Das Statusbyte SW2 kann ggf. weitere Umstände in Abhängigkeit zu Kommandoparametern qualifizieren.

SW1SW2	Addressed Functional Unit			Priority Priorität	Specification	Meaning / Bedeutung
	P1					
	Of Type					
	CT	ICC <n>	FU <n>			
CL-IF <n>						
'6400'	x	x	x	12	Execution Error	An undetermined error occurred during execution of the command. / Unbestimmter Fehler während der Kommandoausführung.
'6700'	x	x	x	3	Wrong (command) length	Wrong Lc: Inconsistent command body or no Command Data field supported.
'6900'	x	x	x	9	Command not allowed.	- The terminal cannot perform the command because there is no valid (open) CT Session (Admin Access Rights required.) - TLS connection
'6941'	x	x	x	10	Functional Unit busy / not available	At the moment FU_CT is busy / unavailable. Funktionseinheit FU_CT ist z.Zt. belegt / nicht verfügbar
'6A00'	x	x	x	5	Wrong parameters P1, P2	- P1 addresses invalid Functional Unit. - In case of Referenced Coding – Invalid / ungültiges FUI DO referenced by P1 (command data). - P2 specifies not supported value
'6A80'	x	x	x	6	Invalid Data Object	- Incorrect data objects / parameters in the command data field - In case of Referenced Coding – The data field contains an invalid DO / Das Datenfeld enthält ein unzulässiges DO. - Invalid FUI DO / Unzulässiges FUI DO
'6A88'	x	x	x	7	Missing Data Object	- Referenced data or reference data not found - In case of Referenced Coding – The data field does not contain a FUI DO / Das Datenfeld enthält kein FUI DO.
'6A89'	x	x	x	8	Too many Data Objects	- Too many Data Objects (FUI DO) of same type. - Es sind zu viele Datenobjekte gleichen Typs (FUI DO, ...) enthalten
'6C00'	x	x	x	4	Wrong Length Le	Wrong Le / falscher Wert Le:
'6D00'	x	x	x	2	Wrong Instruction	Wrong / unbekanntes Instruction-Byte

SW1SW2	Addressed Functional Unit			Priority Priorität	Specification	Meaning / Bedeutung
	P1					
	Of Type					
	CT	ICC <n>	FU <n>			
CL-IF <n>						
'6E00'	x	x	x	1	Class not supported	Falsches / unbekanntes Class-Byte
'6F00'	x	x	x	11	Communication with CT not possible	- In case - the terminal detects general protocol errors or - the terminal detects errors within CT Session or TLS connection
'9000'	x	x	x	12	Command successful	Data Object query successful / Dateobjektabfrage erfolgreich.

5.16 Command SICCT SET STATUS

5.16.1 Funktion

Das SICCT SET STATUS - Kommando muss die Adressierung eines SET-Interfaces des SICCT-Kartenterminals bzw. der adressierbaren Funktionseinheiten erlauben, um generell Statusinformation oder Geräteparameter setzen zu können.

Über den Mechanismus des SET - Interfaces können prinzipiell Datenobjekte übergeben werden, die generelle Zustands-, Chipkarten oder Geräteparameter beeinflussen können. Das Setzen kann in Folge der Parametrisierung eine generelle Betriebsmodus- oder Zustandsänderung bewirken. Die Auswirkung wird durch die Art der übergebenen Daten und der Functional Unit bestimmt.

Anmerkung: Dieses Kommando könnte in einer weiteren Ausbaustufe z.B. herstellerspezifische Initialisierungen, oder die Initialisierung und Aktivierung von Betriebsparametern zulassen.

Folgende SICCT-Terminalkonfigurationsparameter müssen z.Zt. von diesem SICCT Spezifikationsstand vorgesehen⁸ und innerhalb einer ADMIN-CT-Session gesetzt werden können:

Data Object	CT Mode		Addressed FU	Description		
	BCS	SICCT	P1			
Functional Unit Name Data Object	-	✓	CT	see 5.5.10.11	Konfiguration bzw. Vergabe eines logischen Namens für eine Functional Unit des SICCT-Terminals	see Chapter 6 " Terminal Managementverfahren"
					Configuration of a Friendly name for Functional Unit.	

Das Kommando muss mit der Antwort den Ausgang der Operation sowie optional angeforderte Datenobjekte zurückliefern.

Ausführungsphasen SICCT SET STATUS				
Phase		Stage	option	Description
preprocessing phase	Vorbereitungsphase	1		
processing phase	Ausführungsphase	2		- check for CT session state - check for access conditions - check corresponding state - set corresponding state
postprocessing phase	Nachbereitungsphase	3		send return value

⁸ Die Übertragung weiterer Daten(objekte) an weitere Adressaten (weitere FUs) kann (herstellerspezifisch) möglich sein.

5.16.2 Anwendungsbedingungen

Das Setzen von Datenobjekten über die Kommandoschnittstelle kann nur in Abhängigkeit des jeweiligen Datenobjekts und der aktiven Zugriffsrechte geschehen.

Anmerkung: Zurzeit sind in der Definition des SICCT-Standards nur Zugriffe unter ADMIN-Berechtigung in einer CT-Session für ein Datenobjekt definiert.

Abbruchbedingung

Die Ausführung des Kommandos SICCT SET STATUS kann durch Anwender des SICCT-Kartenterminals weder beeinflusst noch abgebrochen werden.

Die Wirkung des Kommandos muss elementar sein und kann durch die steuernde Entität in der Ausführung nicht abgebrochen werden.

SICCT SET STATUS							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
BCS	SICCT				Stage		
					1	2	3
no	✓	✓	no	no	no	no	no

5.16.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	Lc	Data	[Le]
SICCT SET STATUS	'80'	'23'	Functional Unit	Command Qualifier	Length Command Data	Command Data	absent
	▪ CLA = Class ▪ INS = Instruction ▪ P1, P2 = Parameter 1 and 2 ▪ Lc = Length of command data field ▪ Le = Length of expected ▪ SW1, SW2 = Status Bytes				Case 3 (cmd data, no rsp data:) Lc=1-255 Bytes or extended Lc no Le		

Specification C-APDU		
CLA	'80'	Cardterminal Command Class
INS	'23'	SICCT SET STATUS

P1	Functional Unit		
	Addressed FU which receives the given data or data object to be processed.		
	bit8...bit1	Referenced Coding	
		'FF'	Escape: Signal Referenced Coding of P1 Functional Unit referenced by Functional Unit Index Data Object. FUI DO contained within Command Data Field.
	bit8...bit5	Direct Coding (mandatory)	
		'0'	Address contact bound chipcard interface or cardterminal
		'1'	Address contactless chipcard interface (CL-IF)
		'4'	Address FU of type Display Unit
		'5'	Address FU of type Keypad Unit
		'6'	Address the Printer Port
		'7'	Address FU of type Biometric Sensor Unit
	bit4...bit1	'0'	bit8 ... bit5 = '0' Cardterminal (CT)
			bit8 ... bit5 = '4' Standard Display
			bit8 ... bit5 = '5' Standard Keypad
			bit8 ... bit5 = '6' Standard Printer Port
			bit8 ... bit5 = '7' Standard Biometric Sensor Unit: Fingerprint
		'1" : 'E'	bit8 ... bit5 = '0' 1 st ICC-Interface ... 14 th ICC-Interface
			bit8 ... bit5 = '1' 1 st CL-IF ... 14 th CL-IF
			bit8 ... bit5 = '4' 2 nd ... 14 th additional Display
			bit8 ... bit5 = '5' 2 nd ... 14 th additional Keypad
			bit8 ... bit5 = '6' RFU
			bit8 ... bit5 = '7' Specific Biometric Sensor Unit Type

P2	Command Qualifier: indicates Data Object to be set		
	bit8...bit1	Referenced Coding	
		'00'	The data to be processed is given in data field.
		'FF'	Escape: Signal Referenced Coding of 'P2': The data to be processed is given by a Data Object contained within Command Data Field.
		'xx'	Other values RFU

Lc	Length of Command Data Nc		
	variable length	Length of Command Data Nc (no. of bytes contained in Data field)	
		Lc short '01' <= Lc <= 'FF'	1 <= Nc <= 255
		Lc extended 3 Byte Coding '000001' <= Lc <= '00FFFF'	1 <= Nc <= 65535

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	One Data Object with values to be set	In Case 'P2' set to 'FF'
		available data objects see 5.16.4
	In case of Referenced Coding of 'P1'	
	FUI DO	Functional Unit Index Data Object
	Data	In case 'P2' set to '00'

	One Data Object with data to be processed or set	In case 'P2' set to 'FF'
		Available data objects see 5.16.4

Le	Length of Requested Data Ne Return up to <Ne> bytes of requested information		
	Empty	Le absent	No information requested: Ne = 0

5.16.4 Data Objects

Das SICCT SET STATUS Kommando kann mit den folgenden Daten Objekten arbeiten.

Anmerkung: Künftige Spezifikationsstände des SICCT-Standards könnten um weitere Datenobjekte, Zugriffsmöglichkeiten und Zugriffsberechtigungen erweitert werden.

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10.9
FU NAME DO	CMD	Functional Unit Name Data Object	see 5.5.10.11

Note: In case the data part of the command contains data objects which also address a Functional Unit, the value given by parameter P1 (or its referenced FUI DO) shall have priority.

5.16.5 Response Structure

SICCT SET STATUS	Kodierung R-APDU			
	[Body]			Trailer
	[Requested Data / Information]			Status Byte 1 Status Byte 2
	Empty	in case no requested information	SW1	SW2
		in case invalid parameter 'P1' / 'P2'		
		in case Lc was invalid		
		in case Le was invalid or too small		
		in case of error		

5.16.6 Status-Codes SW1SW2

Nach erfolgreicher Ausführung des SET STATUS - Kommandos muss das erste Byte des Statusworts SW1 den Wert '90' enthalten.

- Das Statusbyte SW2 kann ggf. weitere Umstände in Abhängigkeit zu Kommandoparametern qualifizieren.

SW1SW2	Addressed Functional Unit			Priority Priorität	Specification	Meaning / Bedeutung
	P1					
	Of Type					
	CT	ICC <n>	FU <n>			
CL- IF <n>						
'6400'	x	x	x	12	Execution Error	- An undetermined error occurred during execution of the command. / - Unbestimmter Fehler während der Kommando- ausführung.
	-	x	x	12	Reset not suc- cessful	- Error occurred during chipcard reset. - In case - the ICC / PICC does not react on reset or - the ICC / PICC reacts on reset but does not send correct or incomplete reset information (ATR), or - the ICC / PICC does not provide any terminal sup- ported protocol. - State of addressed chipcard set to deactivated state.
'64A1'	-	x	x	11	No Card present	Addressed ICC Interface / slot did not contain or contact a chipcard / RFID token. / Die adressierte (ICC-) Kontaktiereinheit enthielt oder konnte keine Chipkarte / Kontaktlos-
'6700'	x	x	x	3	Wrong (command) length	Wrong Lc: Inconsistent command body or no Command Data field supported.
'6900'	x	x	x	9	Command not allowed.	- The terminal cannot perform the command be- cause there is no valid (open) - CT Session (Admin Access Rights required.) - TLS connection
'6941'	x	x	x	10	Functional Unit busy / not available	- At the moment FU_CT is busy / unavailable. - Funktionseinheit FU_CT ist z.Zt. belegt / nicht ver- fügbar
'6A00'	x	x	x	5	Wrong param- eters P1, P2	- P1 addresses invalid Functional Unit. - In case of Referenced Coding – - Invalid / ungültiges FUI DO referenced by P1 (command data). - P2 specifies not supported value
'6A80'	x	x	x	6	Invalid Data Object	- Incorrect data objects / parameters in the com- mand data field - In case of Referenced Coding – - The data field contains an invalid DO / - Das Datenfeld enthält ein unzulässiges DO. - Invalid FUI DO / Unzulässiges FUI DO
'6A88'	x	x	x	7	Missing Data Object	- Referenced data or reference data not found - In case of Referenced Coding – - The data field does not contain a FUI DO / - Das Datenfeld enthält kein FUI DO.
'6A89'	x	x	x	8	Too many Data Objects	- Too many Data Objects (FUI DO) of same type. - Es sind zu viele Datenobjekte gleichen Typs (FUI DO, ...) enthalten
'6C00'	x	x	x	4	Wrong Length Le	Wrong Le / falscher Wert Le:
'6D00'	x	x	x	2	Wrong Instruc- tion	Wrong / unbekanntes Instruction-Byte
'6E00'	x	x	x	1	Class not sup- ported	Falsches / unbekanntes Class-Byte

'6F00'	X	-	-	11	Communication with CT not possible	▪ In case ▪ the terminal detects general protocol errors or ▪ the terminal detects errors within CT Session or TLS connection ▪ State of addressed chipcard set to deactivated state.
	-	x	x	11	Communication with ICC not possible	▪ In case ▪ the ICC / PICC reacts on reset but does not send correct or incomplete protocol parameter selection (PPS) information ▪ the ICC / PICC reacts on reset but generates protocol errors.
'9000'	-	x	x	12	Command successful	▪ Data Object adjusted: Value set.

5.17 Command SICCT INPUT

5.17.1 Funktion

Das SICCT INPUT-Kommando kann dazu dienen, eine Benutzereingabe über eine vom SICCT-Kartenterminal kontrollierte Eingabefunktionseinheit (Keyboard oder biometrische Einheit) im Datenfeld der Kommandoantwort (Response) zurückzuliefern.

Der Parameter 'Le' bestimmt die Anzahl der Eingabedaten (Bytes). Das Kommando muss mit der Antwort den Ausgang der Operation sowie angeforderte Daten zurückliefern.

Für die Kodierung der Anzahl der Eingabedaten ('Le') müssen die folgenden Regeln gelten

- Le ist nicht vorhanden (absent): Diese Option (Ne = 0) muss die Möglichkeit bieten, ggf. nach der Ausgabe einer Textnachricht, auf die Betätigung der Bestätigungstaste oder Abbruchtaste warten zu können.
- Le = '00': Muss eine variable Eingabe, mit nicht vorgegebener Länge bedeuten.
- Le = <n>: Muss eine Eingabe von <n> Eingabezeichen (Bytes) bedeuten.

Zur Eingabeanforderung kann ein optionaler Benutzerdialog und drei Timereinstellungen zur Zeitüberwachung gesetzt werden. Die Timereinstellungen sind mit Defaultwerten belegt, welche sich mittels Waiting Time Data Objects überschreiben lassen, siehe Tabelle 20 und 5.7.

Tabelle 19: Ausführungsphasen SICCT INPUT

Phase		Stage	option	Description
preprocessing phase	Vorbereitungsphase	1		display message
processing phase with input phase	Ausführungsphase mit Eingabephase, siehe auch 5.7	2		- perform user dialog - wait for user input or abort via keypad - monitor time and process user input until completion or until abortion via keypad - process user input or abort via keypad (input phase)
postprocessing phase	Nachbereitungsphase	3		- end user dialog - display message - send return value

Eingabe via Funktionseinheit vom Typ Keypad

Im Falle der Eingabe über ein Keypad können Zusatzoptionen den Benutzerdialog am SICCT-Kartenterminal steuern, sofern das SICCT-Kartenterminal über eine Anzeigeeinrichtung (Display) verfügt.

- 1) Damit der Benutzer seine Eingabe am Display verfolgen kann, kann optional eine Indikation über das Keypad eingegebener Zeichen dargestellt werden. Entweder kann diese im Klartext oder als Sequenz von Asterics (*)-Zeichen erfolgen. Andere Zeichen dürfen nicht verwendet werden.
- 2) Wird mit einer Anzeigenachricht kein Datenobjekt (Application Label DO oder SICCT Message To Be Displayed DO) übergeben, so muss an einem SICCT mit Display die Darstellung der Standardnachricht Nummer 11 (siehe 7.2) "Bitte Dateneingabe" erfolgen.
- 3) Vor der Ausgabe einer Display-Nachricht wird das Display gelöscht.
- 4) Die Ausgabe von allen, internen wie externen, Display-Nachrichten muss an einem SICCT-Terminal mit Display generell unterdrückt werden (siehe Parameter P2) können.
- 5) Wenn die Nachrichtausgabe via Parameter P2 (bit7 ... bit5) unterdrückt wird, darf keine Nachricht angezeigt und der Display-Inhalt muss gelöscht werden.
Anmerkung: Eine generelle Beschreibung von Vorgaben zum Aufbau und zur Behandlung von Display-Nachrichten findet sich in 5.6.1.
- 6) Ein akustisches oder optisches Signal (siehe Parameter P2) kann optional zu Beginn der angeforderten Dateneingabe ausgegeben werden, sofern das SICCT-Terminal diese Option unterstützt.
- 7) Eine zeitliche Überwachung kann durch das SICCT-Terminal mit Defaultwerten von 15 Sekunden bis zur ersten Eingabe und 5 Sekunden zwischen den einzelnen Eingaben bzw. Tastenbetätigungen erfolgen, wenn kein(e) Datenobjekt(e) (WT DO) mit expliziter Vorgabe angegeben werden.

Tabelle 20: Aktionen des SICCT INPUT Kommandos

Aktion	Standard	Änderungsmöglichkeit	Bedingung
Action	Default-/Min/Max-Value	Override	Condition
Bis zur Eingabe des ersten Zeichens oder Abbruchtaste	15 s min 15 s max 30 s	1 st WAITING TIME Data Object specifying timeout <t1>	Start: Nach Darstellung der ersten Eingabeaufforderung
Until input of 1 st character or cancel key			Starts after presentation of the 1 st Display message
Bis zur Eingabe des jeweils nächsten Zeichens oder Abbruchtaste oder Bestätigungstaste	5 s min 5 s max 30 s	2 nd WAITING TIME Data Object specifying timeout <t2>	Start: Nach jedem Tastendruck
Until input of subsequent characters or cancel key or enter key			Starts after every key stroke
Gesamtwartezeit für die PIN-Eingabe	180 s min 60 s max 255 s	3 rd WAITING TIME Data Object specifying overall timeout <t3>	Start: Nach der Darstellung der ersten Eingabeaufforderung
Overall-Time for the complete PIN-entry			Starts after presentation of the 1 st Display message

- 8) WT DO
 - a) Das erste vorhandene WT DO gibt den Timeout <t1> an. Fehlt das erste WT DO, dann gilt der in Tabelle 20 angegebene Wert.
 - b) Das zweite vorhandene WT DO gibt den Timeout <t2> an. Fehlt das zweite WT DO, dann gilt der in Tabelle 20 angegebene Wert.
 - c) Das dritte vorhandene WT DO gibt den overall Timeout <t3> an. Fehlt das dritte WT DO, dann gilt der in Tabelle 20 angegebene Wert.
- 9) Über die Anzahl der angeforderten Daten (Wert von Le) wird vorgegeben, ob es sich um eine variable Anzahl Daten handelt, bzw. nach welcher Anzahl Eingabezeichen, die Eingabe als abgeschlossen gelten soll.
- 10) Wird das Kommando mit variabler Eingabelänge verwendet, muss folgendes beachtet werden:
 - Die Eingabe muss mit der Bestätigungstaste beendet werden.
 - Sollte nach der Eingabe des letzten Eingabezeichens die Timeout Zeit <t2> oder <t3> überschritten werden, so können folgende SICCT-Terminalaktionen standardkonform gelten:
 - (1.) Die Standardnachricht 12 "Abbruch" wird angezeigt, die Eingabe muss abgebrochen und der entsprechende Rückgabewert generiert werden.
 - (2.) Die Standardnachricht 10 "Bitte Eingabe bestätigen" wird angezeigt und es kann erneut die Timeoutzeit <t2> auf die Bestätigungstaste gewartet werden. Wird innerhalb der Timeout Zeit <t2> oder <t3> die Bestätigungstaste nicht gedrückt, muss die Eingabe wie unter (1.) abgebrochen werden. Die Unterstützung der Standardnachricht Nummer 10 "Bitte Eingabe bestätigen" kann optional erfolgen.
- 11) Bei Erreichen der angeforderten Anzahl Eingabezeichen kann an einem SICCT-Terminal mit Display die Standardnachricht Nummer 10 "Bitte Eingabe bestätigen" erscheinen. Die Unterstützung der Standardnachricht Nummer 10 "Bitte Eingabe bestätigen" kann optional erfolgen.
- 12) Wird das Kommando mit fester Eingabelänge verwendet, dann kann optional die finale Bestätigung der Eingabe durch den Benutzer per Bestätigungstaste explizit angefordert werden (siehe Parameter P2). Wird die finale Bestätigung
 - a) nicht angefordert, dann gilt die Eingabe als bestätigt, wenn nach Eingabe aller geforderten Zeichen der Timeout <t2> abläuft.
 - b) angefordert, dann gilt die Eingabe als abgebrochen, wenn nach Eingabe aller geforderten Zeichen der Timeout <t2> abläuft.
- 13) An einem SICCT-Terminal mit Keypad muss eine angeforderte Dateneingabe generell durch die Betätigung der Bestätigungstaste enden. Das SICCT-Terminal muss die Eingabe beenden und die bis dahin ggf. eingelesene Information an den Aufrufer senden.
- 14) An einem SICCT-Terminal mit Keypad muss eine angeforderte Dateneingabe generell durch die Betätigung der Abbruchtaste enden. Das SICCT-Terminal muss die Eingabe beenden und darf

keine ggf. eingelesenen Informationen an den Aufrufer senden. Bei einem SICCT-Terminal mit Display muss die Standardnachricht Nummer 12 "Abbruch" erscheinen.

- 15) An einem SICCT-Terminal mit Keypad kann eine angeforderte Dateneingabe generell durch die Betätigung der Korrekturtaste korrigiert werden, sofern die Taste innerhalb der zulässigen Timeouts betätigt wurde. Die Betätigung muss das letzte Zeichen löschen und die Eingabeposition entsprechend (nach links) zurücksetzen.
- 16) Bei Überschreiten der Timeout-Grenzen muss das SICCT-Terminal die Dateneingabe, beenden. Ein SICCT-Terminal mit Display muss die Standardnachricht Nummer 12 "Abbruch" darstellen. Das SICCT-Terminal muss die Eingabe beenden und darf keine ggf. eingelesenen Informationen an den Aufrufer senden.

Eingabe via Funktionseinheit vom Typ Biometrischer Sensor:

Im Falle der Eingabe bzw. Aufnahme eines biometrischen Merkmals über einen biometrischen Sensor kann ebenfalls ein Benutzerdialog (Eingabeaufforderung) über das Display stattfinden. Nach der Erfassung des Merkmals (z.B. Fingerprint) kann eine akustische oder optische Bestätigung erfolgen, um den Beginn der Datenaufnahme zu signalisieren.

- Für die Dateneingabe per biometrischer Sensoreinheit müssen prinzipiell dieselben Abläufe wie bei der Keypad-Eingabe gelten.
- Es muss die erste Angabe als Timeout bis zur Datenpräsentation des biometrischen Merkmals gelten. Weitere Timeoutangaben müssen ignoriert werden.
- Eine variable Eingabe muss dann mit der Bestätigungstaste abgeschlossen werden, sofern dieses als Option in Parameter P2 angefordert wurde. Die Unterstützung der Standardnachricht Nummer 10 "Bitte Eingabe bestätigen" kann optional erfolgen.

5.17.2 Anwendungsbedingungen

Das Kommando SICCT INPUT darf generell nur zulässig sein, sofern das SICCT-Kartenterminal die Präsenz von Eingabefunktionseinheiten anzeigt (siehe SICCT GET STATUS). Dieses muss eine Eingabeeinrichtung (entweder ein Keypad oder eine biometrische Sensoreinheit) sein.

Abbruchbedingung

Das Kommando SICCT INPUT kann nur dann eine direkt erkennbare Auswirkung auf den Benutzer haben, sofern ein Statuswechsel an der Mensch-Maschine-Schnittstelle entsteht. Die Ausführung, d.h. eine Anforderung einer Nutzereingabe, kann durch eine Anwenderinteraktion am SICCT-Kartenterminal beeinflusst oder abgebrochen werden. Zulässige Interaktionen sind: Betätigung der Abbruchtaste am Keypad des SICCT-Kartenterminals.

Das Kommando SICCT INPUT kann durch die steuernde Entität während der Ausführung nur innerhalb der Vorbereitungsphase (stage 1) abgebrochen werden.

SICCT INPUT							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
BCS	SICCT			Stage			
				1	2	3	
No	✓	✓	✓	✓	no	no	

5.17.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	[Lc]	[Data]	[Le]
SICCT INPUT	'80'	'16'	Functional Unit	Command Qualifier	Length Command Data	Command Data	Length Re- quested Data
	▪ CLA = Class ▪ INS = Instruction ▪ P1, P2 = Parameter 1 and 2 ▪ Lc = Length of command data field ▪ Le = Length of expected ▪ SW1, SW2 = Status Bytes				Case 1 (no cmd data, no rsp data): no Lc, no Le		
					Case 2 (no cmd data, rsp data): no Lc, Le=1-256 Bytes or extended Le		
					Case 3 (cmd data, no rsp data): Lc=1-255 Bytes or extended Lc no Le		
					Case 4 (cmd data, rsp data): ⁹ Lc=1-255 Bytes or extended Lc Le=1-256 Bytes or extended Le		

Specification C-APDU		
CLA	'80'	Cardterminal Command Class
INS	'16'	SICCT INPUT

P1	Functional Unit		
	bit8...bit1	Referenced Coding	
		'FF'	Escape: Signal Referenced Coding of P1 Functional Unit(s) referenced by Functional Unit Index Data Object (FUI DO) or Functional Unit Context Data Object (FU CON DO). Either one FUI DO or one FU CON DO contained within Command Data Field.
	bit8...bit5	Direct Coding (mandatory)	
		'5'	Address FU of type Keypad Unit
		'7'	Address FU of type Biometric Sensor Unit
	bit4...bit1	'0'	bit8 ... bit5 = '5' Standard Keypad: '50'
			bit8 ... bit5 = '7' Standard Biometric Sensor Unit: '70' Fingerprint Sensor
		'1'	bit8 ... bit5 = '5' 1 st ... 14 th additional Keypad
		'E'	bit8 ... bit5 = '7' 1 st ... 14 th specific Biometric Sensor Unit

⁹ According to [ISO/IEC 7816-4] either both Le and Lc are short or extended.

P2	Command Qualifier:		
	bit8	0	Input does not need Confirmation Key
		1	Input needs Confirmation Key
	bit7...bit5	'7'	Suppress all display messages during command execution ¹⁰
		'0'	Display standard message no. 11 or message found in data field according to Application Label DO
	Option Setting		
	bit4	'0'	Acoustic Signal: None
		'1'	Acoustic Signal: Give Acoustic Signal
	bit3	'0'	Optical Signal: None
		'1'	Optical Signal: Show Optical Signal
	bit2...bit1	In case P1 addresses FU of type Keypad	
		'0'	No input indication
		'1'	Input indication: characters
		'2'	Input indication: asterics
		'3'	RFU
		In case P1 addresses FU of type Biometric Sensor	
		'0'	no input indication
		'x'	1 <= x <= 3: RFU

Lc	Length of Command Data Nc		
	Empty	Lc absent	no Command Data provided; Nc = 0
	Variable length	Length of Command Data Nc (no. of bytes contained in Data field)	
		Lc short '01' <= Lc <= 'FF'	1 <= Nc <= 255
		Lc extended 3 Byte Coding '000001' <= Lc <= '00FFFF'	1 <= Nc <= 65535

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	Empty	non-existent: in case Nc = '00': no Command Data provided
	APPL DO	Application Label Data Object
	SMTBD DO	SICCT Message-To-Be-Displayed Data Object
	WT DO	Waiting Time Data Object
	In case of Referenced Coding of 'P1'	
	FUI DO	Functional Unit Index Data Object
	FU CON DO	Functional Unit Context Data Object
	APPL DO	Application Label Data Object
	SMTBD DO	SICCT Message-To-Be-Displayed Data Object
	WT DO	Waiting Time Data Object(s)

¹⁰ Suppress all internal ('Abort', ...) and external messages.

Le	Length of Requested Data Ne Return up to <Ne> bytes of requested information		
	Empty	Le absent	No information requested: Ne = 0
			Usage: Wait for confirmation or abort key without return of input data.
	Variable length	Le short '01' <= Le <= 'FF'	1 <= Ne <= 255
		Le short Le = '00'	Ne = variable input data length up to 256
		Le extended Lc extended 2 Byte Coding '0001' <= Le <= 'FFFF'	1 <= Ne <= 65535
		Le extended Lc extended 2 Byte Coding Le = '0000'	Ne = variable input data length up to 65536
		Le extended Lc absent 3 Byte Coding '000001' <= Le <= '00FFFF'	1 <= Ne <= 65535
		Le extended Lc absent 3 Byte Coding Le = '000000'	Ne = variable input data length up to 65536

5.17.4 Data Objects

Das SICCT INPUT Kommando kann mit den folgenden Daten Objekten arbeiten.

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10.9
FU CON DO	CMD	Functional Unit Context Data Object (Referenced Coded FU)	see 5.5.10.10
APPL DO	CMD	Text / display message.	see 5.5.10.18
SMTBD DO	CMD	Constructed TLV-DO containing one Character Set and one Application Label DO.	see 5.5.10.20
WT DO	CMD	Up to three (3) Waiting Time Data Objects in fixed order specifying timeout periods in seconds <t1>, >t2> and <t3>.	see 5.5.10.21
		1 st WT DO Max. waiting time <t1> in seconds for 1 st input character, default is 15 a	see 5.17.1
		2 nd WT DO Max. waiting time <t2> in seconds for following input character or entry key, default is 5 s	
		3 rd WT DO Overall waiting time <t3>, default is 180 s	

Erlaubte Kombinationen P1 und Data Objects			
P1	FU CON DO	FUI DO	Bemerkung
Direct	No	No	P1 adressiert direkt die Eingabeeinheit (Keypad oder Bio-Sensor). Implizit ist das Standard-Display selektiert (optional, sofern vorhanden).
Referenced	No	Yes	P1 - Referenced Coding: P1 adressiert direkt die Eingabeeinheit (Keypad oder Bio-Sensor) per FUI DO. Implizit ist das Standard-Display selektiert (optional, sofern vorhanden).
Referenced	Yes	No	P1 - Referenced Coding: P1 adressiert direkt die Eingabeeinheit (Keypad oder Bio-Sensor) innerhalb des FU CON DOs. Das selektierte Display wird innerhalb des FU CON DOs adressiert.

Anmerkung:
 - Weitere Permutationen müssen jeweils einen Fehlerfall darstellen, in denen DO(e) fehlen oder zuviel übergeben werden.
 Als Fehlerfälle sind auch FU CON DO zu bewerten, die keinen zulässigen oder eindeutigen Kontext für SICCT INPUT (min. 2 FUs: ,in. eine Eingabeeinheit (Keypad oder Bio-Sensor), min. eine Ausgabeeinheit (Display) ergeben.

5.17.5 Response Structure

SICCT INPUT	Kodierung R-APDU			
	[Body]			Trailer
	[Requested Data / Information]			Status Byte 1 Status Byte 2
	Empty	in case no requested information	SW1	SW2
		in case invalid parameter 'P1' / 'P2'		
		in case Lc was invalid		
		in case Le was invalid or too small		
		in case of error		
	Requested Information	in case of valid command and error free operation	SW1	SW2
		ASCII characters in case of keyboard		
		biometrical measure data in case of biometrical unit		

5.17.6 Status-Codes SW1SW2

Nach erfolgreicher Ausführung des SICCT INPUT Kommandos muss das erste Byte des Statusworts SW1 den Wert '90' enthalten.

Das Statusbyte SW2 kann weitere Umstände in Abhängigkeit zu Kommandoparametern qualifizieren.

SW1SW2	Addressed Functional Unit		Priority Priorität	Specification	Meaning / Bedeutung
	P1				
	Of Type				
	'50'	'7x'			
	Keypad	Biometric Sensor			
'6400'	x	x	12	No or incomplete input in time	▪ Abort condition raised by timeout.
'6401'	x	x	12	Process aborted by pressing of CANCEL key	▪ Abort: User canceled input operation by pressing of CANCEL key. / ▪ Abbruch - Der Benutzer beendete die Eingabe via ABBRUCH-Taste.
'6700'	x	x	3	Wrong (com- mand) length	▪ Wrong Lc: Inconsistent command body or no Command Data field supported.
'6900'	x	x	9	Command not allowed.	The terminal cannot perform the command because has there is no valid (open) ▪ CT Session ▪ TLS connection
'6930'	x	x	8	Command with timer not sup- ported.	▪ Terminal does not support the timer option. / ▪ Das Terminal unterstützt die Option Zeitüberwachung nicht.
'6940'	x	x	8	Command with Display not sup- ported.	▪ Terminal does not support the display option. / ▪ Das Kommando unterstützt die Option Display (Anzeige) nicht.
'6941'	x	x	10	Functional Unit (FU) busy / not available.	▪ The addressed FU (FU_CT, display, slot, ...) is busy at the moment or not available. / ▪ Die adressierte Funktionseinheit (FU_CT, Anzeige, Kontaktfreieinheit, ...) ist im Moment belegt und ist nicht verfügbar.
'6942'	x	x	8	Selected Char- acter Set not supported.	▪ The addressed display does not support the selected character set. / Die adressierte Anzeige unterstützt den gewählten Zeichensatz nicht.
'6A00'	x	x	5	Wrong param- eters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ In case of Referenced Coding – ▪ Invalid / ungültiges FUI DO referenced by P1 (command data). ▪ P2 specifies not supported value
'6A80'	x	x	6	Invalid Data Ob- ject	▪ Incorrect data objects / parameters in the command data field ▪ In case of Referenced Coding – ▪ The data field contains an invalid DO / ▪ Das Datenfeld enthält ein unzulässiges DO. ▪ Invalid FUI DO / Unzulässiges FUI DO
'6A88'	x	x	7	Missing Data Object	▪ In case of Referenced Coding – ▪ The data field does not contain a FUI DO / ▪ Das Datenfeld enthält kein FUI DO.
'6A89'	x	x	8	Too many Data Objects	▪ Too many Data Objects (FUI DO, ...) of same type. ▪ Es sind zu viele Datenobjekte gleichen Typs (FUI DO, ...) enthalten
'6C00'	x	x	4	Wrong Length Le	▪ Wrong Le / falscher Wert Le
'6D00'	x	x	2	Wrong Instruc- tion	▪ Wrong / unbekanntes Instruction-Byte

'6E00'	x	x	1	Class not supported	Falsches / unbekanntes Class-Byte
'6F00'	x	x	11	Communication with addressed FU not possible	In case the terminal detects errors within CT Session or TLS connection
'9000'	x	x	12	Command successful	the terminal detects general protocol errors or

5.18 Command SICCT OUTPUT

5.18.1 Funktion

Mit dem SICCT OUTPUT-Kommando können generell Daten an eine 'Functional Unit' ausgegeben werden. Aufbau und Inhalt der Ausgabedaten sind abhängig von der Ausprägung der zur Ausgabe adressierten Functional Unit.

Eine als 2D-Code darzustellende Nachricht wird nicht als Bildobjekt übergeben, sondern erfolgt als Textnachricht, aus welcher das SICCT OUTPUT Kommando die entsprechende Grafikdarstellung errechnet. Dabei ist die maximale Länge des Inhaltes bzw. Größe des 2D-Codes von den Anzeigemöglichkeiten des Terminals abhängig.

Die Erweiterungen bezüglich der 2D-Codes sind nur im Kontext von SICCT OUTPUT unter Nutzung des SMTBD DO erlaubt.

Das SICCT OUTPUT-Kommando kann in der aktuellen Basisspezifikation die folgenden optional vorhandenen 'Functional Units' unterstützen:

- Display:
Ausgabe einer Textnachricht zur Benutzerführung am SICCT-Kartenterminal (die Anzeige bleibt erhalten, bis ein neuer Text gesetzt wird)
- biometrische Einheit:
Daten zur Konfigurierung einer biometrischen Einheit.
- Druckerport:
Ausgabe von Daten über optional vorhandene Ausgabe- oder Kommunikationsports des SICCT-Kartenterminals: z.B. Druckdaten für einen ggf.(optional) am SICCT angeschlossenen Druckerport.

Das Kommando muss die entsprechende Ausgabeaktion durchführen und den Ausgang der Operation mit der Antwort zurückliefern.

Folgende Regeln müssen bei der Ansteuerung eines Displays zur Ausgabe von Textnachrichten gelten:

- 1) Zur Ausgabe von Textnachrichten muss entweder ein Application Label oder ein SICCT Message To Be Displayed Datenobjekt übergeben werden.
- 2) Der Aufbau und die Behandlung von Display-Nachrichten findet sich in 5.6.1.
- 3) Das adressierte Display muss vor der Ausgabe der Textnachricht gelöscht werden.
- 4) Die Übergabe einer leeren Textnachricht in Form eines Application Label oder eines SICCT Message To Be Displayed Datenobjekts der Länge Null, muss ein Löschen des Displays bewirken.
- 5) Die Darstellung einer Terminaleigenen Standardtextnachricht (Idle-Message) kann optional über den Parameter P2 erfolgen.
- 6) Die Darstellungsdauer einer Textnachricht kann mittels eines Waiting Time Datenobjekts (WT DO) gesteuert werden, sofern das SICCT-Terminal diese Option unterstützt. Anderenfalls muss das SICCT-Terminal das OUTPUT-Kommando mit Waiting Time-Angabe mit einem entsprechenden Status negieren.
- 7) Die Funktion ist blockierend, d.h. die Anzeige muss erhalten bleiben, bis ein neuer Text gesetzt wird, aber mindestens also so lange, bis ein optional angegebener Timerwert für die Dauer der Darstellung abgelaufen ist.
- 8) Erfolgt keine Angabe eines Timerwertes über ein Waiting Time DO oder wenn der Wert Null explizit in einem WT DO angegeben wird, muss zum Zeitpunkt der Ausführung die unmittelbare Textausgabe und sofortige Rückkehr des Kommandos erfolgen.
- 9) Eine Angabe eines Timerwertes per WT DO mit einem Wert ungleich Null muss die unmittelbare Textausgabe und Rückkehr des Kommandos nach Ablauf der Wartezeit bewirken.

Ausgaben an eine biometrische Einheit oder einen Druckerport dürfen nicht mit einer Timerangabe in Form eines Waiting Time Datenobjekts kombiniert werden.

5.18.2 Anwendungsbedingungen

Das SICCT OUTPUT-Kommando darf nur zulässig sein, wenn die jeweils adressierte und zulässige Functional Unit (Display, Druckerport bzw. biometrischer Sensor) vorhanden ist. Anderenfalls muss das SICCT-Terminal die Ausführung mit einem Statuscode ablehnen.

Abbruchbedingung

SICCT-Kartenterminal mit Keypad: Die Ausführung des Kommandos SICCT OUTPUT kann durch den Anwender des SICCT-Kartenterminals per Abbruchtaste abgebrochen werden.

SICCT-Kartenterminal ohne Keypad: Im Fall, dass kein Keypad verfügbar ist, kann der Anwender die Ausführung des Kommandos nicht beeinflussen.

Das Kommando SICCT OUTPUT kann durch die steuernde Entität in der Ausführung abgebrochen werden.

SICCT OUTPUT							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
BCS	SICCT			Stage			
				1	2	3	
no	✓	✓	✓	✓	no	no	

5.18.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	[Lc]	[Data]	[Le]
SICCT OUTPUT	'80'	'17'	Functional Unit	Command Qualifier	Length Command Data	Command Data	absent
	▪ CLA = Class ▪ INS = Instruction ▪ P1, P2 = Parameter 1 and 2 ▪ Lc = Length of command data field ▪ Le = Length of expected ▪ SW1, SW2 = Status Bytes				Case 1 (no cmd data, no rsp data): no Lc, no Le		
					Case 3 (cmd data, no rsp data:) Lc=1-255 Bytes or extended Lc no Le ¹¹		

Specification C-APDU		
CLA	'80'	Cardterminal Command Class
INS	'17'	SICCT OUTPUT

P1	Functional Unit		
	bit8...bit1	Referenced Coding	
		'FF'	Escape: Signal Referenced Coding of P1 Functional Unit(s) referenced by Functional Unit Index Data Object (FUI DO) or Functional Unit Context Data Object (FU CON DO). Either one FUI DO or one FU CON DO contained within Command Data Field.
	bit8...bit5	Direct Coding (mandatory)	
		'4'	Address FU of type Display
		'6'	Address the Printer Port
		'7'	Address FU of type Biometric Sensor Unit
	bit4...bit1	'0'	bit 8 ... bit 5 = '4' Standard Display: '40'
			bit 8 ... bit 5 = '6' Standard Printer Port: '60'
			bit 8 ... bit 5 = '7' Standard Biometric Sensor Unit: '70' Fingerprint
			bit 8 ... bit 5 = '4' 1 st ... 14 th additional Display

¹¹ Der Fall / Use Case – "Lc darf ,absent" sein, z.B. wenn via P2 nur die Idle-Message dargestellt werden soll.

		'1' : 'E'	bit 8 ... bit 5 = '7'	1 st ... 14 th specific Biometric Sensor Unit
--	--	-----------------	-----------------------	---

P2	Command Qualifier:		
	bit8...bit1	'00'	Default value = do not care
		In case P1 addresses a FU other than Display type.	
		'xx'	other values RFU
		In case P1 addresses a Display FU	
		'01'	Display the cardterminals' Idle Message
		'70'	Suppress internal display messages during command execution ¹²
		'xx'	other values RFU

Lc	Length of Command Data Nc		
	Empty	Lc absent	no Command Data provided; Nc = 0
	variable length	Length of Command Data Nc (no. of bytes contained in Data field)	
		Lc short '01' <= Lc <= 'FF'	1 <= Nc <= 255
		Lc extended 3 Byte Coding '000001' <= Lc <= '00FFFF'	1 <= Nc <= 65535

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	APPL DO	Application Label Data Object
	SMTBD DO	SICCT Message-To-Be-Displayed Data Object
	WT DO	Waiting Time Data Object
	In case of Referenced Coding of 'P1'	
	FUI DO	Functional Unit Index Data Object
	FU CON DO	Functional Unit Context Data Object
	APPL DO	Application Label Data Object
	SMTBD DO	SICCT Message-To-Be-Displayed Data Object
	WT DO	Waiting Time Data Object

Le	Length of Requested Data Ne Return up to Ne bytes of requested information		
	Empty	Le absent	No information requested Ne = 0

¹² Suppress internal ('Abort') message.

5.18.4 Data Objects

Das SICCT OUTPUT Kommando kann mit den folgenden Daten Objekten arbeiten.

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10.9
FU CON DO	CMD	Functional Unit Context Data Object	see 5.5.10.10
APPL DO	CMD	Text / display message.	see 5.5.10.18
SMTBD DO	CMD	Constructed TLV-DO containing one Character Set and one Application Label DO.	see 5.5.10.20
WT DO	CMD	Max. Display Time in seconds, binary coding	see 5.5.10.21

Erlaubte Kombinationen P1 und Data Objects			
P1	FU CON DO	FUI DO	Bemerkung
Direct	No	No	P1 adressiert direkt die Ausgabeeinheit. (Drucker oder Display). Implizit ist das Standard-Keypad (für Benutzer-Abbruch-Aktion) selektiert.
Referenced	No	Yes	P1 - Referenced Coding: P1 adressiert die Ausgabeeinheit (Drucker oder Display) per FUI DO. Implizit ist das Standard-Keypad (für Benutzer-Abbruch-Aktion) selektiert.
Referenced	Yes	No	P1 - Referenced Coding: P1 adressiert die Ausgabeeinheit (Drucker oder Display) innerhalb des FU CON DOs. Das selektierte Keypad (für Benutzer-Abbruch-Aktion) wird innerhalb des FU CON DOs adressiert.
Anmerkung:			
- Weitere Permutationen müssen jeweils einen Fehlerfall darstellen, in denen DO(e) fehlen oder zuviel übergeben werden.			
Als Fehlerfälle sind auch FU CON DO zu bewerten, die keinen zulässigen oder eindeutigen Kontext für SICCT OUTPUT (min. 2 FUs: min. eine Ausgabeeinheit (Drucker oder Display), min. eine Eingabeeinheit (Keypad)) ergeben.			

5.18.5 Response Structure

SICCT OUTPUT	Kodierung R-APDU			
	[Body]		Trailer	
	absent		Status Byte 1	Status Byte 2
	Empty	in case no requested information	SW1	SW2
		in case invalid parameter 'P1' / 'P2'		
		in case Lc was invalid		
		in case Le was invalid or too small		
in case of error				
in case of valid command and error free operation				

5.18.6 Status-Codes SW1SW2

Nach erfolgreicher Ausführung des SICCT OUTPUT Kommandos muss das erste Byte des Statusworts SW1 den Wert '90' enthalten.

Das Statusbyte SW2 kann weitere Umstände in Abhängigkeit zu Kommandoparametern qualifizieren.

SW1SW2	Addressed Functional Unit P1 of Type			Priority Priorität	Specification	Meaning / Bedeutung
	'40'	'60'	'7x'			
	Display	Printer	Biometric Sensor			
'6400'	x	x	x	12	Execution Error	An undetermined error occurred during execution of the command. / Unbestimmter Fehler während der Kommandoausführung.
					Output not successful	- Error occurred during output operation. - Rendering 2D-code image is too big.
'6401'	x	x	x		Process aborted by pressing of cancel key	Abort condition: User canceled input operation. / Abbruchbedingung: Anwender brach Ausführung der Ausgabe ab.
'6410'	x	x	x	11	Functional Unit (FU) unable to process the output data	the addressed FU does not support the quantity or type of provided output data
'6700'	x	x	x	3	Wrong (command) length	Wrong Lc: Inconsistent command body or command data (message, ...) is too long.
'6900'	x	x	x	9	Command not allowed.	The terminal cannot perform the command because there is no valid (open) - CT Session (Admin Access Rights required.) - TLS connection
'6930'	x	x	x		Command with timer not supported	Terminal does not support the timer option. Das Terminal unterstützt die Timer-Option nicht.
'6940'	x	x	x	8	Command with Display not supported	Command with not supported for addressed FU (display, printer, biometric sensor)
'6941'	x	x	x	10	Functional Unit busy / not available	At the moment FU_CT or addressed FU is busy / unavailable. Funktionseinheit FU_CT oder die adressierte FU ist z.Zt. belegt / nicht verfügbar
'6942'	x	x	x	8	Selected Character Set not supported	The addressed FU (display, printer, ...) does not support the selected character set. Die adressierte FU unterstützt den angegebenen Zeichensatz nicht.
'6A00'	x	x	x	5	Wrong parameters P1, P2	- P1 addresses invalid Functional Unit. - In case of Referenced Coding – - Invalid / ungültiges FUI DO referenced by P1 (command data). - P2 specifies not supported value
'6A80'	x	x	x	6	Invalid Data Object	- Incorrect data objects / parameters in the command data field - In case of Referenced Coding – - The data field contains an invalid DO / Das Datenfeld enthält ein unzulässiges DO. Invalid FUI DO / Unzulässiges FUI DO - In case e.g. output date (Display message, ...) too long.
'6A88'	x	x	x	7	Missing Data Object	- Referenced data or reference data not found - In case of Referenced Coding – - The data field does not contain a FUI DO / Das Datenfeld enthält kein FUI DO.
'6A89'	x	x	x	8	Too many Data Objects	Too many Data Objects (FUI DO) of same type. Es sind zu viele Datenobjekte gleichen Typs (FUI DO, ...) enthalten
'6C00'	x	x	x	4	Wrong Length Le	- Wrong Le / falscher Wert Le: - Le has to absent / Le darf nicht angegeben sein.

'6D00'	x	x	x	2	Wrong Instruction	Wrong / unbekanntes Instruction-Byte
'6E00'	x	x	x	1	Class not supported	Falsches / unbekanntes Class-Byte
'6F00'	X	-	-	11	Communication with CT not possible	In case the terminal detects - general protocol errors or - errors within CT Session or - TLS connection - the addressed FU has reported a general error.
	x	x	x	11	Communication with addressed FU not possible	
'9000'	x	x	x	12	Command successful	Output data processed by addressed FU / Ausgabedaten wurden von der adressierten FU verarbeitet.

Beispiel – SICCT OUTPUT an FU(Display) zur Ausgabe eines QR-Codes

Ausgabe der Nachricht „the brown fox jumps over the lazy dogs bag.“ am Terminal Display für maximal 50 Sekunden als QR-Code:

```

CAPDU   =   SICCT OUTPUT || SMTBD DO || WT DO
CAPDU   =   80 17 40 00 CLA, INS, P1, P2
           38          Lc
           A0 32
           85 01 30 character set DO encoding QR-Code
           04 2d 207468652062726f776e20666f78206a
                756d7073206f76657220746865206c61
                797a2064666773206261672e20
           80 01 32 WT DO encoding 50 s
RAPDU   =   9000' SW1SW2

```

5.19 Command SICCT PERFORM VERIFICATION

5.19.1 Funktion

Das Kommando SICCT PERFORM VERIFICATION muss der sicheren PIN-basierten und biometrischen Benutzerauthentisierung des Kartenhalters über Funktionseinheiten des SICCT-Kartenterminals dienen.

Das Kommando kann hierzu die optional zeitgesteuerten Teilfunktionalitäten ermöglichen:

- die (optionale) Anzeige von Texten zur Bedienerführung am SICCT-Kartenterminal mit Display,
- die Entgegennahme einer Abbruchinteraktion an einem SICCT-Kartenterminal mit PIN-Tastatur (Keypad),
- die Entgegennahme der PIN-Eingabe an einem SICCT-Kartenterminal mit PIN-Tastatur (Keypad),
- die (optionale) Erfassung biometrischer Messdaten an einem SICCT-Kartenterminal mit biometrischer Sensoreinheit,
- die entsprechende Interaktion mit einer adressierten Chipkarte oder einem RFID Token

Die Interaktion mit der Chipkarte oder dem RFID Token muss bestehen aus

- 1) dem Senden des im Datenfeld des SICCT PERFORM VERIFICATION-Kommandos übergebenen Kommandos (Data Object Command-to-Perform, see 5.5.10.22), nachdem die vom SICCT-Terminal erfasste PIN, der Resetting Code oder ein vom SICCT-Terminal aufbereiteter biometrischer Messdatensatz in das im DO 'Command-to-perform' angegebene Chipkartenkommando (APDU) an entsprechender Stelle (Insertion Position) eingesetzt wurde, und
- 2) in dem Empfang und der Rückgabe des von der Chipkarte oder RFID-Token (PICC) generierten Ergebniswerts.

Das Kommando muss den Status (Erfolg, Abbruch) bzw. das Ergebnis der Verifikation (Erfolg, Fehlerkodierung der Chipkarte) zurückgeben. Generell gilt: Das Kommando muss den Ausgang der Operation mit der Antwort zurückliefern. Wurde vom SICCT-Terminal ein Kommando an die Chipkarte gesendet (Command To Perform), muss das SICCT-Terminal die von der Chipkarte zurückgegebene Statusinformation (SW1SW2) an den Aufrufer weiterreichen.

Tabelle 21: Ausführungsphase SICCT PERFORM VERIFICATION

Phase		Stage	Option	Description
Preprocessing phase	Vorbereitungsphase	1		▪ display message
Processing phase with input phase	Ausführungsphase mit Eingabephase, siehe auch 5.7	2		▪ perform user dialog ▪ wait for user input or abort via keypad ▪ monitor time and process user input until completion or until abortion via keypad ▪ in case of successful input completion: build ICC APDU (command to perform) ▪ in case of successful input completion: perform ICC / PICC command
Postprocessing phase	Nachbereitungsphase	3		▪ display message ▪ send return value

5.19.2 Anwendungsbedingungen

Das Kommando darf generell nur zulässig sein, wenn das SICCT-Kartenterminal über entsprechende Eingabefunktionseinheiten, d.h. die Functional Units Keypad oder eine biometrische Sensorik, verfügt. Die Ausführung von Benutzerdialogen darf nur möglich werden, sofern das SICCT-Terminal über mindestens ein vom SICCT-Terminal kontrolliertes Display verfügt. Die Anzeige von Benutzerdialogen kann durch die steuernde Entität generell gesteuert werden. Im SICCT-Terminal vorgehaltene Standard-Meldungen müssen zur Verfügung stehen oder können für die Ausführungszeit des Kommandos individuell überschrieben werden, indem der Aufrufer entsprechende Datenobjekte im Datenteil des Kommandos übergibt. Verfügt das SICCT-Terminal über multiple Ausgabeeinrichtungen (Displays) kann zurzeit nur eine angesprochen werden.

Anmerkung: Eine generelle Beschreibung von Vorgaben zum Aufbau und zur Behandlung von Display-Nachrichten, insbesondere mit PIN-Eingabe-Prompt, findet sich in 5.6.1.

Numerische bzw. alphanumerische Eingaben müssen den nachfolgend aufgeführten PIN-/ Passwort-Formaten entsprechen, und dürfen nur möglich werden, sofern das SICCT-Terminal über mindestens ein vom SICCT-Terminal kontrolliertes Keypad mit dem entsprechend benötigtem Tastensatz besitzt. Verfügt das SICCT-Terminal über mehrere Eingabeeinrichtungen (Keypads) kann zurzeit nur ein Keypad zur Eingabe angefordert werden.

Das SICCT-Kartenterminal muss standardmäßig das erste Keypad und Display, sowie Standardmeldungen verwenden. Ein hiervon abweichender Ausführungskontext kann mit dem Kommando in Form des 'P2'-Parameters und entsprechender Datenobjekte im Datenteil übergeben werden.

Tabelle 22: Standardausgabertext und Änderungsmöglichkeiten

Standardausgabertext		Darstellung in	Änderungsmöglichkeit	
Default Message		Displayed at	Override	
4	"Bitte Geheimzahl eingeben"	Eingabephase	1 st APPL DO	1 st SMTBD DO
	"Please enter PIN"	Stage 1		
5	"Aktion erfolgreich"	Ausführungsphase	2 nd APPL DO	2 nd SMTBD DO
	"Action successful"	Stage 2		
6	"Geheimzahl falsch/gesperrt"	Nachbereitungsphase	3 rd APPL DO	3 rd SMTBD DO
	"PIN wrong or blocked"	Stage 3		
10	"Bitte Eingabe bestätigen"	Ende der Eingabephase bei variabler PIN-Länge	4 th APPL DO	4 th SMTBD DO
	"Please confirm input"	End of stage 1 and in case of indefinite PIN-Length		
12	"Abbruch"	Abbruch durch ▪ Timeout oder Abbruchtaste vor oder während der Eingabephase ▪ Entzug der Karte ▪ PIN-Eingabe mit inkorrekturer Länge	5 th APPL DO	5 th SMTBD DO
	"Abort"	Abort by ▪ pressing of Cancel Key before or during stage 1 ▪ card withdraw ▪ PIN entry too long/short		

Für eine Benutzerauthentisierung (CHV, Card Holder Verification) nach ISO 9564-1 [STD17], [ISO/IEC 7816-4] oder CEN 726-3 [STD19] (VERIFY-Kommando) müssen folgende PIN-Formate zur Verfügung stehen:

Format 2 PIN Block	see 5.5.10.22
Binary Coded Digit (BCD)	
ASCII, d.h. Zeichen nach T.50-Standard	

Zur Vorgabe der Parameter zur Anwahl des PIN-Formats, der APDU-Bytesequenz, der PIN-Länge sowie der Einfügeposition (Insertion Position), muss dem SICCT-Terminal ein übergebenes Command-To-Perform Data Object (CMD DO) vorgegeben werden können. Zusätzlich muss sich über das Control-Byte des CMD DOs bestimmen lassen, ob die Eingabe automatisch mit dem Erreichen der entsprechenden Anzahl Eingabezeichen oder vom Karteninhaber erst durch die Betätigung der Bestätigungstaste am Keypad enden muss.

Die Eingabephase muss mit der Ausgabe einer ersten Meldung über das gewählte Display beginnen, welche den Kartenhalter zur Eingabe auffordert. Der Standardtext Nr. 4 muss dargestellt werden, sofern kein Application-Data Object übergeben wurde. Den zeitlichen Ablauf der Eingabe kann das SICCT-Terminal über folgende vorgegebene Timer-Einstellungen überwachen, die ebenfalls per WaitingTime Data Objects überschrieben werden können.

Tabelle 23: Aktionen SICCT PERFORM VERIFICATION

Aktion	Standard	Änderungsmöglichkeit	Bedingung
Action	Default-/ min / max- Value	Override	Condition
Bis zur Eingabe des ersten Zeichens oder Abbruchtaste	15 s min 15 s max 30 s	1 st WT DO specifying timeout <t1>	Start: Nach Darstellung der ersten Eingabeaufforderung
Until input of 1 st character or cancel key			Starts after presentation of the 1 st Display message
Bis zur Eingabe des jeweils nächsten Zeichens oder Abbruchtaste oder Bestätigungstaste	5 s min 5 s max 30 s	2 nd WT DO specifying timeout <t2>	Start: Nach jedem Tastendruck
Until input of subsequent characters or cancel key or enter key			Starts after each key stroke
Gesamtwartezeit für die PIN-Eingabe	180 s min 35 s max 255 s	3 rd WT DO specifying overall timeout <t3>	Start: Nach der Darstellung der ersten Eingabeaufforderung
Overall-Time for the complete PIN-entry			Starts after presentation of the 1 st Display message

Hat der Karteninhaber die Eingabe vorgenommen, muss die Eingabephase enden mit

- 1) dem Erreichen der vorgegeben Anzahl Eingabezeichen und dem Ablauf des Timeouts <t2> oder <t3>, falls die Bestätigungstaste nicht angefordert wurde,
- 2) jeder Betätigung der Bestätigungstaste innerhalb der vorgegebenen Zeitspanne oder
- 3) dem Abbruch, in dem der Karteninhaber entweder die Karte dem SICCT-Kartenterminal entzieht oder die Abbruchtaste verwendet.

Wird das Kommando mit variabler PIN-Länge verwendet so muss folgendes gelten:

- 1) Die PIN-Eingabe muss mit der Bestätigungstaste beendet werden können.
- 2) Wird nach der Eingabe der letzten PIN-Stelle der Timeout <t2> oder <t3> überschritten, so können folgende SICCT-Terminalaktionen standardkonform ausgeführt werden:
 - a) Es muss die Display Message 12 (Standard: "Abbruch") angezeigt, die PIN-Eingabe abgebrochen und der entsprechende Rückgabewert generiert werden.
 - b) Es muss die Display Message 10 (Standard: "Bitte Eingabe bestätigen") angezeigt und erneut die eingestellte Timeoutzeit <t2> gesetzt sowie diese Zeitspanne auf die Bestätigungstaste gewartet werden.
Wird die Bestätigungstaste innerhalb der Timeout <t2> oder <t3> nicht gedrückt, muss die PIN-Eingabe wie unter 1. beschrieben abgebrochen werden.

SICCT-Terminals, welche die Variante 1. bei der variablen PIN-Länge unterstützen, dürfen die Display Message 10 (Standard: "Bitte Eingabe bestätigen") nicht verwenden. Die Display Message 10 darf nur in Variante 2 erscheinen.

Sollen jedoch die Standard-Messages durch Datenobjekte im Datenteil des Kommandos überschrieben werden, so muss auch bei SICCT-Terminals der Variante 1. die Reihenfolge der zu überschreibenden Displaytexte wie in Tabelle 22 beschrieben berücksichtigt werden und bei Änderung aller Standard Display Texte auch die Display Message 10 mit angegeben werden!

Zum Ende der Eingabephase muss das SICCT-Terminal die Benutzereingabe intern in das eingestellte PIN-Format umwandeln, und diese in das APDU- Kommando einfügen.

SICCTs mit biometrischem Sensor (z.B. Fingerprint-Sensor) müssen eine Erfassung und entsprechende Aufbereitung der biometrischen Messdaten sowie deren Integration in das Chipkartenkommando erlauben. Auch in diesem Fall muss der Abbruch per Abbruchtaste oder Kartenentnahme während der Eingabephase möglich sein.

Nach der Eingabephase muss das SICCT-Terminal das aufbereitete Chipkartenkommando (Ausführungsphase) senden, sofern sich die adressierte Chipkarte bzw. das RFID-Token (PICC) im entsprechend aktivierten Betriebszustand befindet. Anderenfalls muss das SICCT-Terminal die Ausführung abbrechen, das Kartenkommando verwerfen und einen entsprechenden Fehlerstatus an den Aufrufer senden. Die Ausführungsphase des Command-to-Perform muss mit dem Empfang des von der Chipkarte generierten Statusworts (SW1SW2) enden.

Während der Nachbereitungsphase muss ein SICCT-Terminal mit Display dem Bediener (Karteninhaber) den Status bzw. den Ausgang der Operation signalisieren, indem es entweder eine Standardmeldung (Standardmessage Nr. 5 oder 6) oder eine übergebene Meldung am adressierten Display darstellen muss.

No	Message	Phase	Condition
5	"Aktion erfolgreich"	Ausführungsphase	Chipcard Response equals '90 00'
		Stage 2	
6	"Geheimzahl falsch/gesperrt"	Nachbereitungsphase	Chipcard Response unequal to '90 00'
		Stage 3	

Anschließend muss das SICCT-Terminal eine R-APDU (SW1SW2) an den Aufrufer senden.

Abbruchbedingung

Das Kommando SICCT PERFORM VERIFICATION darf nur dann eine direkte Auswirkung auf den Benutzer haben, sofern ein erkennbarer Statuswechsel an der Mensch-Maschine-Schnittstelle entsteht. Die Ausführung, d.h. eine Anforderung einer Nutzereingabe, kann durch eine Anwenderinteraktion am SICCT-Kartenterminal beeinflusst oder nach einem erreichten Timeout (s. timer <t1>, >t2> und <t3>) abgebrochen werden.

Zulässige Interaktionen durch den Benutzer können während der Eingabephase sein:

Eingabe der angeforderten Identifikationsdaten innerhalb eines spezifizierten Zeitrahmens <t3>	Successful Operation
Keine oder unvollständige Eingabe der angeforderten Identifikationsdaten innerhalb eines spezifizierten Zeitintervalls bzw. Gesamtzeitrahmens <t1> bzw. <t2> oder <t3>	Abort condition
Entnahme der Chipkarte oder des RFID-Tokens (PICC) aus dem Wirkbereich des SICCT-Kartenterminals vor und während der Eingabephase,	
Betätigung der Abbruchtaste am Keypad des SICCT-Kartenterminals	
Keine Betätigung der Bestätigungstaste, falls diese als Abschluss der Eingabe angefordert wurde.	

Nach Auftreten eines Timeouts oder nach Abbruch per Abbruchtaste muss das SICCT-Terminal die Standardmeldung Nr. 12 oder die mit dem Kommando übergebene Textmeldung des fünften APPL DO bzw. SMTBD DO ausgeben.

Die Ausführung des Kommando SICCT PERFORM VERIFICATION kann durch die steuernde Entität während der Eingabephase abgebrochen werden.

SICCT PERFORM VERIFICATION							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
BCS	SICCT			Stage			
					1	2	3
no	✓	✓	✓		✓	no	No

5.19.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	Lc	Data	[Le]
SICCT PERFORM VERIFICATION	'80'	'18'	Functional Unit	Command Qualifier	Length Command Data	Command Data	Length Requested Data

	- CLA = Class - INS = Instruction - P1, P2 = Parameter 1 and 2 - Lc = Length of command data field - Le = Length of expected - SW1, SW2 = Status Bytes	Case 3 (cmd data, no rsp data): Lc=1-255 Bytes or extended Lc no Le
		Case 4 (cmd data, rsp data): Lc=1-255 Bytes or extended Lc Le=1-256 Bytes

Specification C-APDU		
CLA	'80'	Cardterminal Command Class
INS	'18'	SICCT PERFORM VERIFICATION

P1	Functional Unit			
	bit8...bit1	Referenced Coding		
		'FF'	Escape: Signal Referenced Coding of P1 Functional Unit(s) referenced by Functional Unit Index Data Object (FUI DO) or Functional Unit Context Data Object (FU CON DO). Either one FUI DO or one FU CON DO contained within Command Data Field.	
	bit8...bit5	Direct Coding (mandatory)		
		'0'	Address contact bound chipcard interface or cardterminal	
		'1'	Address contactless chipcard interface (CL-IF)	
		'x'	x <> '0', '1': other values RFU	
	bit4...bit1	'0'	RFU	
		'1' : 'E'	bit 8 ... bit 5 = '0'	1 st ICC-Interface ... 14 th ICC-Interface
			bit 8 ... bit 5 = '1'	1 st CL-IF ... 14 th CL-IF

P2	Command Qualifier specifies the Functional Units which performs input data					
	Referenced Coding					
	bit7...bit1	'7F'	Functional Unit which performs input and / or output data contained within FU CON DO			
	bit8	0	Input does not need Confirmation Key			
		1	Input needs Confirmation Key			
	Direct Coding (mandatory)					
	bit7...bit1	'5x'	User authentication via PIN-pad			
			'0' <= x <= "E'	'50'	Standard Keypad	
				:	1 st ... 14 th additional Keypad	
						'51'
						'5E'
		'7x'	User authentication via biometric sensor			
			'0' <= x <= "E'	'70'	Standard Biometric Sensor Fingerprint Sensor	
				'7x'	Other Biometric Sensor Types see 5.5.9	
	bit8				In case P2 <> '7F' and P2 addresses Keypad or Biometric Sensor	
		0	Input does not need Confirmation Key			
		1	Input needs Confirmation Key			

Lc	Length of Command Data Nc			
	variable length	Length of Command Data Nc (no. of bytes contained in Data field)		
		Lc short '01' <= Lc <= 'FF'	1 <= Nc <= 255	
		Lc extended 3 Byte Coding '000001' <= Lc <= '00FFFF'	1 <= Nc <= 65535	

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	APPL DO	Application Label Data Object
	SMTBD DO	SICCT Message-To-Be-Displayed Data Object
	WT DO	Waiting Time Data Object
	CMD DO	Command-To-Perform Data Object
	In case of Referenced Coding of 'P1'	
	FUI DO	Functional Unit Index Data Object
	FU CON DO	Functional Unit Context Data Object
	APPL DO	Application Label Data Object
	SMTBD DO	SICCT Message-To-Be-Displayed Data Object
	WT DO	Waiting Time Data Object
	CMD DO	Command-To-Perform Data Object

Le	Length of Requested Data Ne		
	Return up to Ne bytes of requested information		
	Empty	Le absent	No information requested Ne = 0
	variable length	Le short '01 <= Le <= 'FF'	1 <= Ne <= 255
		Le short Le = '00'	Ne = 256:

5.19.4 Data Objects

Das SICCT PERFORM VERIFICATION kann mit den folgenden Daten Objekten arbeiten.

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	Functional Unit IndexData Object	see 5.5.10.9
FU CON DO	CMD	Functional Unit Context Data Object	see 5.5.10.10
		Functional Unit Index Data Object(s) for command execution (Referenced Coded FU)	
		Selection of ▪ Keypad ▪ Display, ▪ Biometric Sensor Unit ▪ ICC or RFID.	
APPL DO	CMD	APPLICATION LABEL Data Object	see 5.5.10.18 see command description 1 st to 5 th SMTBD DO
		Text / display message(s).	
		Up to five (5) Application Label Data Objects (in case no SICCT Message-to-be-Displayed Dos given)	
SMTBD DO	CMD	SICCT Message To Be Displayed Data Object	see 5.5.10.20 see command description 1 st to 5 th SMTBD DO
		Text / display message(s).	
		Constructed TLV-DO containing one Character Set and one Application Label DO (in case no Application Label DOs given).	
		Up to five (5) SICCT Message-To-be-Displayed Data Objects in a fixed order.	
WT DO	CMD	Waiting Time Data Object	see 5.5.10.21 see command description 1 st to 3 rd WT DO
		Max. Waiting Time(s) in seconds	
		Up to three (3) Waiting Time Data Objects in fixed order specifying timeout periods in seconds <t1>, >t2> and <t3>	
CMD DO	CMD	Command To Perform Data Object	see 5.5.10.22
		APDU to be sent to the card	

Erlaubte Kombinationen aus P1, P2 und Data Objects				
P1	P2	FU CON DO	FUI DO	Bemerkung
Direct	Direct	No	No	▪ P1 adressiert direkt den Slot. ▪ P2 adressiert direkt die Eingabeeinheit (Keypad oder Bio-Sensor) sowie implizit das Standard-Display.
Referenced	Direct	No	Yes	▪ P1 - Referenced Coding: P1 verweist auf FUI DO mit Slotangabe. ▪ P2 adressiert direkt die Eingabeeinheit (Keypad oder Bio-Sensor) sowie implizit das Standard-Display.
Referenced	Referenced	Yes	No	▪ P1 Referenced Coding: P1 verweist auf ein Datenobjekt im Datenteil. Da P2 ebenfalls Referenced Coding verwendet, verweist P1 auf ein gemeinsames FU CON DO mit der ▪ Slotangabe und min. der Eingabeeinheit (Keypad oder Bio-Sensor) oder ▪ Slotangabe und der Eingabeeinheit (Keypad oder Bio-Sensor) sowie Ausgabeeinheit (Display). ▪ P2 - Referenced Coding: P2 verweist auf ein Datenobjekt im Datenteil.
Anmerkung: - Weitere Permutationen müssen jeweils einen Fehlerfall darstellen, in denen DO(e) fehlen oder zuviel übergeben werden. Als Fehlerfälle müssen auch FU CON DO bewertet werden, die keinen zulässigen oder eindeutigen Kontext für PIN-Eingaben (min. 2 FUs: min. ein Slot, min. ein Keypad, optional: ein Display) ergeben.				

5.19.5 Response Structure

Wurde vom SICCT-Terminal ein Kommando an die Chipkarte gesendet (Command To Perform), muss das SICCT-Terminal das von der Chipkarte zurückgegebene Statuswort (SW1SW2) als Ergebnis im Trailer zurückgeben.

SICCT PERFORM VERIFICATION	Kodierung R-APDU				
	[Body]			Trailer	
	[Requested Data / Information]			Status Byte 1	Status Byte 2
	Empty	in case no requested information		SW1	SW2
		in case invalid parameter 'P1' / 'P2'			
		in case Lc was invalid			
		in case Le was invalid or too small			
		in case of error			
		in case of valid command and error free operation			
	ICC returned R-APDU	[ICC returned information on performed command]		ICC returned status code	
SW1				SW2	

5.19.6 Status-Codes SW1SW2

Nach erfolgreicher Ausführung des SICCT PERFORM VERIFICATION - Kommandos muss das erste Byte des Statusworts SW1 den Wert '90' enthalten. Das Statusbyte SW2 kann weitere Umstände in Abhängigkeit zu Kommandoparametern qualifizieren. Generell muss gelten: Wurde vom SICCT-Terminal ein Kommando an die Chipkarte gesendet (Command To Perform), muss das SICCT-Terminal das von der Chipkarte zurückgegebene Statuswort (SW1SW2) als Ergebnis im Trailer zurückgeben.

SW1SW2	Addressed Functional Unit		Priority Priorität	Specification	Meaning / Bedeutung
	P1				
	Of Type				
	ICC<n>	CL- IF<n>			
'63Cx'	x	x		Verification unsuccessful. x = number of possible re-tries	Note: Chipcard generated status word in case the PIN verification failed.
'6400'	x	x	12	No or incomplete input in time	Abort condition raised by timeout during stage 1 to stage 3.
'6401'	x	x	12	Process aborted by pressing of CANCEL key	Abort: User canceled input operation by pressing of CANCEL key. Abbruch - Der Benutzer beendete die Eingabe via ABBRUCH-Taste.
'6402'	x	x	12	Process unsuccessful, new PIN not identical	Fail: The confirmation of the PIN entry was unsuccessful. Fehler: Die Bestätigung der PIN war nicht erfolgreich.
'64A1'	x	x	11	No Card present	Verification not performed, because no card present addressed ICC Interface / slot did not contain or contact a chipcard / RFID token.

'64A2'			11	Card not activated	Verification not performed, because no activated card.
'6700'	x	x	3	Wrong (command) length	- Wrong Lc: Inconsistent command body or no Command Data field supported. - Message too long.
'6900'	x	x	9	Command not allowed.	- The terminal cannot perform the command because there is no valid (open) CT Session - TLS connection
'6930'	x	x	8	Command with timer not supported.	Terminal does not support the timer option. / Das Terminal unterstützt die Option Zeitüberwachung nicht.
'6940'	x	x	8	Command with Display not supported.	Terminal does not support the display option. / Das Kommando unterstützt die Option Display (Anzeige) nicht.
'6941'	x	x	9	Functional Unit (FU) busy / not available.	The addressed FU (display, slot, ...) is busy at the moment or not available. / Die adressierte Funktionseinheit (Anzeige, Konatktiereinheit, ...) ist im Moment belegt und ist nicht verfügbar.
'6941'	x	x	10	Functional Unit busy / not available	At the moment FU_CT is busy / unavailable. Funktionseinheit FU_CT ist z.Zt. belegt / nicht verfügbar
'6942'	x	x	8	Selected Character Set not supported.	The addressed display does not support the selected character set. / Die adressierte Anzeige unterstützt den gewählten Zeichensatz nicht.
'6983'	x	x	12	Verification Method Blocked.	Note: Chipcard generated status word in case the PIN verification failed, because the authentication method is blocked.
'6A00'	x	x	5	Wrong parameters P1, P2	- P1 addresses invalid Functional Unit. - In case of Referenced Coding – Invalid / ungültiges FUI DO referenced by P1 (command data). - P2 specifies not supported value - Invalid combination of P1 and P2
'6A80'	x	x	6	Incorrect parameters in the command data field or Invalid Data Object	- Incorrect data objects / parameters in the command data field - General data error - In case of CMD DO e.g. non-allowed 'INS' code. - In case of Referenced Coding – The data field contains an invalid DO / Das Datenfeld enthält ein unzulässiges DO. - Invalid FUI DO / Unzulässiges FUI DO - Invalid FU CON DO / Unzulässiges FU CON DO
'6A88'	x	x	7	Missing Data Object	In case of Referenced Coding – The data field does not contain a FUI DO or FU CON DO/ Das Datenfeld enthält kein FUI DO oder FU CON DO
'6A89'	x	x	8	Too many Data Objects	Too many Data Objects (CMD DO, FUI DO, FU CON DO, APPL DO, SMTBD DO, WT DO) of same type. Es sind zu viele Datenobjekte gleichen Typs (CMD DO, FUI DO, FU CON DO, APPL DO, SMTBD DO, WT DO) enthalten

'6C00'	x	x	4	Wrong Length Le	- Wrong Le / falscher Wert Le: - If Le exists, here it has to be set to '00'.
'6D00'	x	x	2	Wrong Instruction	Wrong / unbekanntes Instruction-Byte
'6E00'	x	x	1	Class not supported	Falsches / unbekanntes Class-Byte
'6F00'	x	x	11	Communication with addressed FU not possible	- In case - the terminal detects general protocol errors or - the terminal detects errors within CT Session or TLS connection
'9000'	x	x	12	Command successful	Note: Chipcard generated status word in case the PIN verification was successful.

Note: In case the SICCT-Terminal has successfully generated and sent an APDU (command to perform) to the chipcard, the returned status word has been taken from the chipcards' response. Due to a variety of cards there might be more corresponding values (SW1SW2) than '9000', '6983' and '63Cx'.

5.20 Command SICCT MODIFY VERIFICATION DATA

5.20.1 Funktion

Das Kommando SICCT MODIFY VERIFICATION DATA muss einen Spezialfall zur Änderung der Kartenauthentifikationsdaten darstellen, mit welchem eine PIN, ein Kartenpasswort oder ein biometrisches Merkmal (PIN, Passwort, biometrisches Merkmal) abgefragt und geändert werden kann.

Das Kommando SICCT MODIFY VERIFICATION DATA muss ebenfalls eine PIN-basierte oder biometrische Benutzerauthentisierung des Kartenhalters über Funktionseinheiten des SICCT-Kartenterminals durchführen. Zusätzlich muss nach der Eingabe des bisherigen (alten) Merkmals das jeweilige neue Merkmal erfragt, und in ein übergebenes Command-To-Perform eingebettet werden.

Das Kommando muss dieselben Anwendungsbedingungen und prinzipiell gleiche Funktionalitäten wie das SICCT PERFORM VERIFICATION Kommando bieten, bis auf die Tatsache, dass weitere Dialoge (Meldungen) und Benutzereingaben zur Eingabe des neuen Merkmals notwendig sein können. Die Eingabe des neuen Merkmals muss nach der ersten Eingabe wiederholt werden, damit das SICCT-Terminal einen internen Vergleich vornehmen kann.

Bei diesem Kommando muss die Interaktion mit der Chipkarte oder dem RFID Token aus diesen Teilschritten bestehen

- 1) Senden des im Datenfeld des SICCT MODIFY VERIFICATION DATA -Kommandos übergebenen Kommandos (Data Object Command-To-Perform, see 5.5.10.22), nachdem die vom SICCT-Terminal erfasste alte und neue PIN, oder Resetting Code oder ein vom SICCT-Terminal aufbereiteter biometrischer Messdatensatz in das im DO Command-To-Perform angegebene Chipkartenkommando (APDU) an entsprechender Stelle (Insertion Position) eingesetzt wurde, und
- 2) Empfang und der Rückgabe des von der Chipkarte oder RFID-Token (PICC) generierten Ergebnisses.

Das Kommando muss den Status (Erfolg, Abbruch) bzw. das Ergebnis der Verifikation (Erfolgreich, Fehlerkodierung der Chipkarte) zurückgeben.

Wurde ein Kommando an die adressierte Chipkarte bzw. RFID-Token (PICC) gesendet, muss das Kommando den Ausgang der Chipkartenoperation mit der Antwort zurückliefern.

Tabelle 24: Ausführungsphasen SICCT MODIFY VERIFICATION DATA

	Phase	Stage	Option	Description
Preprocessing phase	Vorbereitungsphase	1		▪ display message ▪ perform user dialog ▪ wait for user input or abort via keypad
Processing phase	Ausführungsphase	2		▪ process user input or abort via keypad (input phase) ▪ build ICC APDU (command to perform) ▪ perform ICC / PICC command
Postprocessing phase	Nachbereitungsphase	3		▪ display message ▪ send return value

5.20.2 Anwendungsbedingungen

Das Kommando darf generell nur zulässig sein, wenn das SICCT-Kartenterminal über entsprechende Eingabefunktionseinheiten, d.h. die Functional Units Keypad oder eine biometrische Sensorik, verfügt.

Die Ausführung von Benutzerdialogen darf nur möglich sein, sofern das Terminal über mindestens ein vom Terminal kontrolliertes Display verfügt. Die Anzeige von Benutzerdialogen kann durch die steuernde Entität generell gesteuert werden. Im Terminal vorgehaltene Standard-Meldungen müssen zur Verfügung stehen oder können für die Ausführungszeit des Kommandos individuell überschrieben werden, indem der Aufrufer entsprechende Datenobjekte im Datenteil des Kommandos übergibt. Verfügt das SICCT-Terminal über multiple Ausgabeeinrichtungen (Displays) kann zurzeit nur eine angesprochen werden.

Anmerkung: Eine generelle Beschreibung von Vorgaben zum Aufbau und zur Behandlung von Display-Nachrichten, insbesondere mit PIN-Eingabe-Prompt, findet sich in 5.6.1.

Numerische bzw. alphanumerische Eingaben entsprechend den nachfolgend aufgeführten PIN-/ Passwort-Formaten dürfen nur möglich sein, sofern das SICCT-Terminal über mindestens ein vom SICCT-Terminal gesteuertes Keypad verfügt. Verfügt das SICCT-Terminal über mehrere Eingabeeinrichtungen (Keypads) kann zurzeit nur eine zur Eingabe angefordert werden.

Das SICCT-Kartenterminal muss standardmäßig das erste Keypad und Display sowie Standardmeldungen verwenden. Ein hiervon abweichender Ausführungskontext kann mit dem Kommando in Form des 'P2'-Parameters und entsprechender Datenobjekte im Datenteil übergeben werden.

Tabelle 25: Standardausgabertext und Änderungsmöglichkeiten

Standardausgabertext		Darstellung in	Änderungsmöglichkeit	
Default Message		Displayed at	Override	
4	"Bitte Geheimzahl eingeben"	Eingabephase	1 st APPL DO	1 st SMTBD DO
	"Please enter PIN"	Stage 1		
5	"Aktion erfolgreich"	Ausführungsphase	2 nd APPL DO	2 nd SMTBD DO
	"Action successful"	Stage 2		
6	"Geheimzahl falsch/gesperrt"	Nachbereitungsphase	3 rd APPL DO	3 rd SMTBD DO
	"PIN wrong or blocked"	Stage 3		
7	"Neue Geheimzahl Eingeben"	Eingabephase	4 th APPL DO	4 th SMTBD DO
	"Please enter new PIN"	Stage 1		
8	"Eingabe wiederholen"	Eingabephase	5 th APPL DO	5 th SMTBD DO
	"Repeat input"	Stage 1		
9	"Geheimzahl nicht gleich. Abbruch"	Eingabephase	6 th APPL DO	6 th SMTBD DO
	"PIN not identical. Abort"	Stage 1		
10	"Bitte Eingabe bestätigen"	Ende der Eingabephase bei variabler PIN-Länge	7 th APPL DO	7 th SMTBD DO
	"Please confirm input"	End of stage 1 and in case of indefinite PIN-Length		
12	"Abbruch"	Abbruch durch ▪ Abbruchtaste vor oder während der Eingabephase ▪ Entzug der Karte ▪ PIN-Eingabe mit inkorrekturer Länge	8 th APPL DO	8 th SMTBD DO
	"Abort"	Abort by ▪ pressing of Cancel Key before or during stage 1 ▪ card withdraw ▪ PIN entry too long/short		

Für eine Benutzerauthentisierung (CHV, Card Holder Verification) nach ISO 9564-1 [STD17], [ISO/IEC 7816-4] oder CEN 726-3 [STD19] (VERIFY-Kommando) müssen folgende PIN-Formate zur Verfügung stehen:

Format 2 PIN Block	see 5.5.10.22
Binary Coded Digit (BCD)	
ASCII, d.h. Zeichen nach T.50-Standard	

Die Anwahl des PIN-Formats, eine APDU-Bytesequenz, die PIN-Länge sowie eine Einfügeposition (Insertion Position), müssen dem SICCT-Terminal durch ein übergebenes Command-To-Perform Data Object (CMD DO) vorgegeben werden. Zusätzlich muss über die Angabe des Control-Bytes im CMD DO über die Länge der einzugebenden PINs bestimmt werden.

Wenn in den Bits b4...b3 des Control Bytes im CMD DO (s. 5.5.10.22) der Wert '0' (b4=0, b3=0) kodiert ist, so kann in den Bits b8...b5 die exakte Länge der einzugebenden alten und neuen PIN vorgegeben werden. Dabei muss eine Länge von 0 als variable Eingabelänge interpretiert werden.

Wenn in den Bits b4...b3 des Control Bytes des CMD DO der Wert '1' (b4=0, b3=1) codiert ist, so wird in den Bits b8...b5 die Mindestlänge der einzugebenden neuen PIN vorgegeben.

In diesem Fall gilt für die alte PIN eine variable Eingabelänge.

Das SICCT-Terminal soll auch bei Format2PIN Block formatierten PINs die vom Control Byte vorgegebenen Längenangaben nutzen, solange die vorgegebene Länge im zulässigen Bereich von 4 bis 12 Zeichen liegt.

Sonderfall Null-PIN-Verfahren:

Zur Initiierung einer sog. NULL-PIN-Abfrage muss das CMD DO in allen Datenobjekten vollständig besetzt übergeben werden. Die erste 'Insert Position' muss als Indikator den Wert Null ('00'), nicht benötigte Display-Nachrichten müssen als notwendige Platzhalter in Form von 'leeren' Datenobjekten, bestehend aus Tag und Value, übergeben werden. Die Kommandoausführung muss die erste PIN-Eingabe übergehen und sich dann auf die Eingabe und Bestätigung der (zweiten) PIN beschränken, die an die Position der '2nd INSERTION Position' in das Karten-APDU eingetragen werden muss. Der (Null)-Inhalt, bzw. der Wert, der ersten PIN muss im Karten-APDU vom Aufrufer im CMD DO mit übergeben werden.

Die Eingabephase muss mit der Ausgabe einer ersten Meldung über das gewählte Display beginnen, welche den Kartenhalter zur Eingabe auffordert. Der Standardtext Nr. 4 muss dargestellt werden, sofern kein Application Label oder SICCT Message-To-Be-Displayed Data Object übergeben wurde. Den zeitlichen Ablauf der Eingabe kann das SICCT-Terminal über folgende vorgegebene Timer-Einstellungen überwachen, die ebenfalls über WaitingTime Data Objects überschrieben werden können.

Tabelle 26: Aktionen SICCT MODIFY VERIFICATION DATA

Aktion	Standard	Änderungsmöglichkeit	Bedingung
Action	Default-/min / max-Value	Override	Condition
Bis zur Eingabe des ersten Zeichens oder Abbruchtaste	15 s min 15 s max 30 s	1 st WT DO specifying timeout <t1>	Start: Nach Darstellung der ersten Eingabeaufforderung
Until input of 1 st character or cancel key			Starts after presentation of the 1 st Display message
Bis zur Eingabe des jeweils nächsten Zeichens oder Abbruchtaste oder Bestätigungstaste	5 s min 5 s max 30 s	2 nd WT DO specifying timeout <t2>	Start: Nach jedem Tastendruck
Until input of subsequent characters or cancel key or enter key			Starts after each key stroke
Gesamtwartezeit für jede PIN-Eingabe	180 s min 35 s max 255 s	3 rd WT DO specifying overall timeout <t3>	Start: Nach der Darstellung der jeder Eingabeaufforderung
Overall-Time for the each complete PIN-entry			Starts after presentation of the each Display message

Hat der Karteninhaber die Eingabe vorgenommen, muss die Eingabephase des einzugebenden Merkmals (alte PIN, neue Pin und Wiederholung der neuen PIN) jeweils enden mit

- 1) dem Erreichen der vorgegeben Anzahl Eingabezeichen und dem Ablauf des Timeouts <t2> oder <t3>,
- 2) jeder Betätigung der Bestätigungstaste innerhalb der vorgegebenen Zeitspanne oder
- 3) dem Abbruch, in dem der Karteninhaber entweder die Karte dem SICCT-Kartenterminal entzieht oder die Abbruchtaste verwendet.

Wird das Kommando mit Variabler PIN-Länge verwendet so muss folgendes gelten:

- 1) Die PIN-Eingabe muss mit der Bestätigungstaste beendet werden.
- 2) Wird nach der Eingabe der letzten PIN-Stelle der Timeout <t2> oder <t3> überschritten, so können folgende SICCT-Terminal-Aktionen standardkonform gelten:
 - a) Die Display Message 12 (Standard: "Abbruch") muss angezeigt, die PIN-Eingabe abgebrochen und der entsprechende Rückgabewert generiert werden.
 - b) Es muss die Display Message 10 (Standard: "Bitte Eingabe bestätigen") angezeigt und erneut die eingestellte Timeoutzeit <t2> gesetzt sowie diese Zeitspanne auf die Bestätigungstaste gewartet werden.
 Wird die Bestätigungstaste innerhalb der Timeout <t2> oder <t3> nicht gedrückt, muss die PIN-Eingabe wie unter 1. beschrieben abgebrochen werden.

SICCT-Terminals, die Variante 1. bei der variablen PIN-Länge unterstützen, können optional die Display Message 10 (Standard: "Bitte Eingabe bestätigen") verwenden. Sollen jedoch die Standard Messages

durch Datenobjekte im Datenteil des Kommandos überschrieben werden, so muss auch bei SICCT-Terminals der Variante 1. die Reihenfolge der zu überschreibenden Displaytexte, wie in der weiter oben aufgeführten Tabelle der Standardtexte beschrieben, berücksichtigt werden, und bei Änderung aller Standard Display Texte auch die Display Message 10 mit angegeben werden.

Zum Ende der gesamten Eingabephase muss das SICCT-Terminal die Benutzereingaben intern in das eingestellte PIN-Format umwandeln, und diese in das APDU- Kommando einfügen.

SICCTs mit biometrischem Sensor (z.B. Fingerprint-Sensor) müssen eine Erfassung und entsprechende Aufbereitung der biometrischer Messdaten sowie Integration in das Chipkartenkommando erlauben. Auch in diesem Fall muss der Abbruch per Abbruchtaste oder Kartenentnahme während der gesamten Eingabephase möglich sein.

Anderenfalls muss das SICCT-Terminal die Ausführung abbrechen, das Kartenkommando verwerfen und einen entsprechenden Fehlerstatus an den Aufrufer senden. Die Ausführungsphase des Command-to-Perform muss mit dem Empfang des von der Chipkarte generierten Statusworts (SW1SW2) enden.

Während der Nachbereitungsphase muss das SICCT-Terminal dem Karteninhaber den Status bzw. den Ausgang der Operation signalisieren, indem entweder eine Standardmeldung (Standardmessage Nr. 5 oder 6) oder eine übergebene Meldung am eingestellten Display dargestellt werden muss.

No	Message	Phase	Condition
5	"Aktion erfolgreich"	Ausführungsphase	Chipcard Response equals '90 00'
		Stage 2	
6	" Geheimezahl falsch/gesperrt"	Nachbereitungsphase	Chipcard Response unequal to '90 00'
		Stage 3	

Anschließend muss das SICCT-Terminal eine R-APDU (SW1SW2) an den Aufrufer senden.

Abbruchbedingung

Das Kommando SICCT MODIFY VERIFICATION DATA darf nur dann eine direkte Auswirkung auf den Benutzer haben, sofern ein erkennbarer Statuswechsel an der Mensch-Maschine-Schnittstelle entsteht. Die Ausführung, d.h. eine Anforderung einer Nutzereingabe, kann durch eine Anwenderinteraktion am SICCT-Kartenterminal beeinflusst oder nach einem erreichten Timeout (s. timer <t1>, >t2> und <t3>) abgebrochen werden.

Zulässige Interaktionen durch den Benutzer können während der Eingabephase sein:

Eingabe der angeforderten Identifikationsdaten innerhalb eines spezifizierten Zeitrahmens	Successful Operation
Keine oder unvollständige Eingabe der angeforderten Identifikationsdaten innerhalb eines spezifizierten Zeitintervalls bzw. Gesamtzeitrahmens <t1> bzw. <t2> oder <t3>	Abort condition
Entnahme der Chipkarte oder des RFID-Tokens (PICC) aus dem Wirkungsbereich des SICCT-Kartenterminals vor und während der Eingabephase,	
Betätigung der Abbruchtaste am Keypad des Kartenterminals	
Keine Betätigung der Bestätigungstaste, falls diese als Abschluss der Eingabe angefordert wurde.	

Nach Abbruch per Abbruchtaste muss das SICCT-Terminal Standardmeldung Nr. 12 oder die mit dem Kommando übergebene Textmeldung des fünften APPL DO bzw. SMTBD DO ausgeben.

Die Ausführung des Kommando SICCT MODIFY VERIFICATION DATA kann durch die steuernde Entität während der Eingabephase abgebrochen werden.

SICCT MODIFY VERIFICATION DATA							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
BCS	SICCT				Stage		
					1	2	3
no	✓	✓	✓		✓	no	no

5.20.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	Lc	Data	[Le]
SICCT MODIFY VERIFICATION DATA	'80'	'19'	Functional Unit	Command Qualifier	Length Command Data	Command Data	Length Re- quested Data
	▪ CLA = Class ▪ INS = Instruction ▪ P1, P2 = Parameter 1 and 2 ▪ Lc = Length of command data field ▪ Le = Length of expected ▪ SW1, SW2 = Status Bytes				Case 3 (cmd data, no rsp data:) Lc=1-255 or extended Lc no Le		
					Case 4 (cmd data, rsp data): Lc=1-255 Bytes or extended Lc Le=1-256 Bytes		

Specification C-APDU		
CLA	'80'	Cardterminal Command Class
INS	'19'	SICCT MODIFY VERIFICATION DATA

P1	Functional Unit		
	bit8...bit1	Referenced Coding	
		'FF'	Escape: Signal Referenced Coding of P1 Functional Unit(s) referenced by Functional Unit Index Data Object (FUI DO) or Functional Unit Context Data Object (FU CON DO). Either one FUI DO or one FU CON DO contained within Command Data Field.
	bit8...bit5	Direct Coding (mandatory)	
		'0'	Address contact bound chipcard interface or cardterminal
		'1'	Address contactless chipcard interface (CL-IF)
		'x'	x <> '0', '1': other values RFU
	bit4...bit1	'0'	RFU
		'1'	bit 8 ... bit 5 = '0' 1 st ICC-Interface ... 14 th ICC-Interface
		'E'	bit 8 ... bit 5 = '1' 1 st CL-IF ... 14 th CL-IF

P2	Command Qualifier specifies the Functional Units which performs input data		
	Referenced Coding		
	bit7...bit1	'7F'	Functional Unit which performs input and / or output data contained within FU CON DO
	bit8	0	Input does not need Confirmation Key
		1	Input needs Confirmation Key
	Direct Coding (mandatory)		
	bit7...bit1	User authentication via PIN-pad	
		'5x'	'50' Standard Keypad
			'51' : 1 st ... 14 th additional Keypad
			'5E'
		User authentication via biometric sensor	
	'7x'	'0' <= x <= "E'	'70' Standard Biometric Sensor
			Fingerprint Sensor
			'7x' Other Biometric Sensor Types see 5.5.9
	bit8	In case P2 <> '7F' and P2 addresses Keypad or Biometric Sensor	
		0	Input does not need Confirmation Key
		1	Input needs Confirmation Key

Lc	Length of Command Data Nc		
	variable length	Length of Command Data Nc (no. of bytes contained in Data field)	
		Lc short '01' <= Lc <= 'FF'	1 <= Nc <= 255
		Lc extended 3 Byte Coding '000001' <= Lc <= '00FFFF'	1 <= Nc <= 65535

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	Empty	non-existent: in case Nc = '00': no Command Data provided
	APPL DO	Application Label Data Object
	SMTBD DO	SICCT Message-To-Be-Displayed Data Object
	WT DO	Waiting Time Data Object
	CMD DO	Command-To-Perform Data Object
	In case of Referenced Coding of 'P1'	
	FUI DO	Functional Unit Index Data Object
	FU CON DO	Functional Unit Context Data Object
	APPL DO	Application Label Data Object
	SMTBD DO	SICCT Message-To-Be-Displayed Data Object
	WT DO	Waiting Time Data Object
	CMD DO	Command-To-Perform Data Object

Le	Length of Requested Data Ne Return up to Ne bytes of requested information		
	Empty	Le absent	No information requested Ne = 0
	variable length	Le short '01' <= Le <= 'FF'	1 <= Ne <= 255
		Le short Le = '00'	Ne = 256:

5.20.4 Data Objects

Das SICCT MODIFY VERIFICATION DATA arbeitet mit den folgenden Daten Objekten.

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	Functional Unit IndexData Object	see 5.5.10.9
FU CON DO	CMD	Functional Unit Context Data Object	see 5.5.10.10
		Functional Unit Index Data Object(s) for command execution (Referenced Coded FU)	
		Selection of - Keypad - Display, - Biometric Sensor Unit - ICC or RFID.	
APPL DO	CMD	APPLICATION LABEL Data Object	see 5.5.10.18
		Text / display message(s).	
		Up to eight Application Label DataObjects (in case no SICCT Message-to-be-Displayed Dos given)	
SMTBD DO	CMD	SICCT Message To Be Displayed Data Object	see 5.5.10.20
		Text / display message(s).	
		Constructed TLV-DO containing one Character Set and one Application Label DO (in case no Application Label DOs given).	
		Up to eight SICCT Message-To-be-Displayed DataObjects	
WT DO	CMD	Waiting Time Data Object	see 5.5.10.21
		Max. Waiting Time(s) in seconds	see command de-
		Up to three (3) Waiting Time Data Objects in fixed order specifying timeout periods in seconds <t1>, >t2> and <t3>	scription 1 st to 3 rd WT DO
CMD DO	CMD	Command To Perform Data Object	see 5.5.10.22
		APDU to be sent to the card	

Erlaubte Kombunationen aus P1, P2 und Data Objects				
P1	P2	FU CON DO	FUI DO	Bemerkung
Direct	Direct	No	No	P1 adressiert direkt den Slot. P2 adressiert direkt die Eingabeeinheit (Keypad oder Bio-Sensor) sowie implizit das Standard-Display.
Referenced	Direct	No	Yes	P1 - Referenced Coding: P1 verweist auf FUI DO mit Slotangabe. P2 adressiert direkt die Eingabeeinheit (Keypad oder Bio-Sensor) sowie implizit das Standard-Display.
Referenced	Referenced	Yes	No	P1 - Referenced Coding: P1 verweist auf ein Datenobjekt im Datenteil. Da P2 ebenfalls Referenced Coding verwendet, verweist P1 auf ein gemeinsames FU CON DO mit der - Slotangabe und min. der Eingabeeinheit (Keypad oder Bio-Sensor) oder - Slotangabe und der Eingabeeinheit (Keypad oder Bio-Sensor) sowie Ausgabeeinheit (Display). P2 - Referenced Coding: P2 verweist auf ein Datenobjekt im Datenteil.
Anmerkung: - Weitere Permutationen stellen müssen jeweils einen Fehlerfall darstellen, in denen DO(e) fehlen oder zuviel übergeben werden. Fehlerfälle müssen auch FU CON DO darstellen, welche keinen zulässigen oder eindeutigen Kontext für PIN-Eingaben (min. 2 FUs: min. ein Slot, min. ein Keypad, optional: ein Display) ergeben.				

5.20.5 Response Structure

Wurde vom SICCT-Terminal ein Kommando an die Chipkarte gesendet (Command To Perform), muss das SICCT-Terminal das von der Chipkarte zurückgegebene Statuswort (SW1SW2) als Ergebnis im Trailer zurückgeben.

SICCT MODIFY VERIFICATION DATA	Kodierung R-APDU				
	[Body]			Trailer	
	[Requested Data / Information]			Status Byte 1	Status Byte 2
	Empty	in case no requested information		SW1	SW2
		in case invalid parameter 'P1' / 'P2'			
		in case Lc was invalid			
		in case Le was invalid or too small			
		in case of error			
		in case of valid command and error free operation			
	ICC returned R-APDU	[ICC returned information on performed command]		ICC returned sta- tus code	
SW1				SW2	

5.20.6 Status-Codes SW1SW2

Nach erfolgreicher Ausführung des SICCT MODIFY VERIFICATION DATA - Kommandos muss das erste Byte des Statusworts SW1 den Wert '90' enthalten.

Das Statusbyte SW2 kann weitere Umstände in Abhängigkeit zu Kommandoparametern qualifizieren.

SW1SW2	Addressed Functional Unit		Priority Priorität	Specification	Meaning / Bedeutung
	P1				
	Of Type				
	ICC<n>	CL- IF<n>			
'63Cx'	x	x		Verification unsuccessful. x = number of possible re-tries	Note: Chipcard generated status word in case the PIN verification failed.
'6400'	x	x	12	No or incomplete input in time	Abort condition raised by timeout (during stage 1).
'6401'	x	x	12	Process aborted by pressing of CANCEL key	Abort: User canceled input operation by pressing of CANCEL key. / Abbruch - Der Benutzer beendete die Eingabe via ABBRUCH-Taste.
'6402'	x	x	12	Process unsuccessful, new PIN not identical	Fail: The confirmation of the PIN entry was unsuccessful. Fehler: Die Bestätigung der PIN war nicht erfolgreich.
'64A1'	x	x	11	No Card present	Verification not performed, because no card present Addressed ICC Interface / slot did not contain or contact a chipcard / RFID token.
'64A2'			11	Card not activated	Verification not performed, because no activated card.
'6700'	x	x	3	Wrong (command) length	- Wrong Lc: Inconsistent command body or no Command Data field supported. - Message too long.

'6900'	x	x	9	Command not allowed.	- The terminal cannot perform the command because there is no valid (open) - CT Session - TLS connection
'6930'	x	x	8	Command with timer not supported.	Terminal does not support the timer option. / Das Terminal unterstützt die Option Zeitüberwachung nicht.
'6940'	x	x	8	Command with Display not supported.	Terminal does not support the display option. / Das Kommando unterstützt die Option Display (Anzeige) nicht.
'6941'	x	x	9	Functional Unit (FU) busy / not available.	The addressed FU (display, slot, ...) is busy at the moment or not available. / Die adressierte Funktionseinheit (Anzeige, Konaktiereinheit, ...) ist im Moment belegt und ist nicht verfügbar.
'6941'	x	x	10	Functional Unit busy / not available	At the moment FU_CT is busy / unavailable. Funktionseinheit FU_CT ist z.Zt. belegt / nicht verfügbar
'6942'	x	x	8	Selected Character Set not supported.	The addressed display does not support the selected character set. / Die adressierte Anzeige unterstützt den gewählten Zeichensatz nicht.
'6983'	x	x	12	Verification Method Blocked.	Note: Chipcard generated status word in case the PIN verification failed, because the authentication method is blocked.
'6A00'	x	x	5	Wrong parameters P1, P2	- P1 addresses invalid Functional Unit. - In case of Referenced Coding – Invalid / ungültiges FUI DO referenced by P1 (command data). - P2 specifies not supported value - Invalid combination of P1 and P2
'6A80'	x	x	6	Incorrect parameters in the command data field or Invalid Data Object	- Incorrect data objects / parameters in the command data field - General data error - In case of CMD DO e.g. non-allowed 'INS' code. - In case of Referenced Coding – The data field contains an invalid DO / Das Datenfeld enthält ein unzulässiges DO. - Invalid FUI DO / Unzulässiges FUI DO Invalid FU CON DO / Unzulässiges FU CON DO
'6A88'	x	x	7	Missing Data Object	In case of Referenced Coding – The data field does not contain a FUI DO or FU CON DO/ Das Datenfeld enthält kein FUI DO oder FU CON DO
'6A89'	x	x	8	Too many Data Objects	Too many Data Objects (CMD DO, FUI DO, FU CON DO, APPL DO, SMTBD DO, WT DO) of same type. Es sind zu viele Datenobjekte gleichen Typs (CMD DO, FUI DO, FU CON DO, APPL DO, SMTBD DO, WT DO) enthalten
'6C00'	x	x	4	Wrong Length Le	- Wrong Le / falscher Wert Le: - If Le exists, here it has to be set to '00'.
'6D00'	x	x	2	Wrong Instruction	Wrong / unbekanntes Instruction-Byte

'6E00'	x	x	1	Class not supported	Falsches / unbekanntes Class-Byte
'6F00'	x	x	11	Communication with addressed FU not possible	- In case the terminal detects general protocol errors or - the terminal detects errors within CT Session or TLS connection
'9000'	x	x	12	Command successful	Note: Chipcard generated status word in case the change of Verification Data was successful.
Note: In case the SICCT-Terminal has successfully generated and sent an APDU (command to perform) to the chipcard, the returned status word has been taken from the chipcards' response. Due to a variety of cards there might be more corresponding values (SW1SW2) than '9000', '6983' and '63Cx'.					

5.21 Command SICCT COMFORT AUTHENTICATION

5.21.1 Funktion

Ein SICCT-Terminal kann optional über weitere Funktionen zur Benutzerauthensierung verfügen. Verschiedene Authentisierungsverfahren können zum Einsatz kommen:

- Biometrische Sensoreinheit
- RFID Token mit kryptographischen Funktionen
- Tastenkombination auf PIN-Pad

Das SICCT-Kartenterminal kann optional zwei weitere Kommandos zur Verfügung stellen:

- SICCT COMFORT ENROLL
- SICCT COMFORT AUTHENTICATION

SICCT COMFORT ENROLL kann eine Möglichkeit darstellen, Referenzdaten aufzunehmen, intern zu speichern und in Relation zu einem Chipkartenmerkmal (z.B. ICCSN) verwalten bzw. referenzieren zu können. Die Präsentation der Referenzdaten kann später als zusätzliches Auslösekriterium bestimmter Transaktionen an die referenzierte Chipkarte dienen.

SICCT COMFORT AUTHENTICATION kann den erforderlichen Präsentationsprozess, quasi als Bestätigung des berechtigten Karteninhabers, darstellen. Erst nach der Präsentation und Überprüfung des Referenzmerkmals (PIN, Passwort oder biometrisches Merkmal), kann ein angeforderter Kartenprozess (z.B. Auslösen eines Signaturprozesses) ausgelöst werden. Das SICCT-Kartenterminal oder eine externe Entität können die für den Vergleichsprozess entsprechend vorgehaltenen Referenzdaten bereitstellen.

Um ein entsprechend geschütztes Kommando ausführen zu können, muss das SICCT-Kartenterminal das Kommando SICCT COMFORT AUTHENTICATION mit dem eindeutigen Kartenidentifikator (z.B. Kartenseriennummer) erhalten. Das SICCT-Kartenterminal kann dann das entsprechende Benutzeridentifikationsmerkmal (PIN, Passwort oder biometrisches Merkmal) sowie die Anwesenheit des sicheren Objekts vor jeder einzelnen Kommandoausführung überprüfen.

Ausführungsphasen SICCT COMFORT AUTHENTICATION				
Phase		Stage	option	Description
preprocessing phase	Vorbereitungsphase	1		▪ display message ▪ perform user dialog ▪ wait for user input or abort via keypad
processing phase	Ausführungsphase	2		▪ process user input or abort via keypad
postprocessing phase	Nachbereitungsphase	3		▪ display message ▪ send return value

5.21.2 Anwendungsbedingungen

Abbruchbedingung:

Das Kommando SICCT COMFORT AUTHENTICATION kann durch die steuernde Entität in der Ausführung nicht abgebrochen werden.

Das Kommando SICCT COMFORT AUTHENTICATION darf nicht im Betriebsmodus BCS unterstützt werden.

SICCT COMFORT AUTHENTICATION							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
BCS	SICCT				Stage		
					1	2	3
no	✓	✓	✓	no	no	no	no

5.21.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	[Lc]	[Data]	[Le]
SICCT COMFORT AUTHENTICATION	'80'	'21'	Functional Unit	Command Qualifier	Length Command Data	Command Data	Length Re- quested Data
	▪ CLA = Class ▪ INS = Instruction ▪ P1, P2 = Parameter 1 and 2 ▪ Lc = Length of command data field ▪ Le = Length of expected ▪ SW1, SW2 = Status Bytes						

5.22 Command SICCT COMFORT ENROLL

5.22.1 Funktion

Mittels einer Anlernfunktion kann ein Referenzdatum, bestehend aus einem biometrischen Merkmal des Karteninhabers sowie einem eindeutigen Kartenmerkmal, z.B. die Kartenseriennummer (ICCSN), gebildet und im SICCT-Kartenterminal gespeichert werden.

Zur Erzeugung des Referenzdatums kann das SICCT-Kartenterminal ein eindeutiges Kartenmerkmal sowie ein Identitätsmerkmal des Karteninhabers einlesen. Optional kann zuvor eine Authentifizierung des Kartenanwenders erfolgen.

SICCTs mit biometrischem Sensor können - falls dies für das betreffende Verfahren sinnvoll ist - auch einen Wechsel der biometrischen Daten ermöglichen (z.B. statt Daumen soll der Zeigefinger bei der Fingerabdruck-Verifikation verwendet werden).

Das SICCT-Kartenterminal muss die angelernten und gespeicherten Referenzdaten vor externen Zugriffen schützen und sicher verwalten. Nach der Aufnahme von Referenzdaten muss das SICCT-Terminal die zugvorige Authentifizierung des Karteninhabers gegen das Referenzdatum als verpflichtende Ausführungsbedingung für bestimmte Karten- oder SICCT-Kartenterminalkommandos initiieren können.

Ausführungsphasen SICCT COMFORT ENROLL				
Phase		Stage	option	Description
preprocessing phase	Vorbereitungsphase	1		
processing phase	Ausführungsphase	2		
postprocessing phase	Nachbereitungsphase	3		▪ send return value

5.22.2 Anwendungsbedingungen

Abbruchbedingung:

Das Kommando SICCT COMFORT ENROLL kann durch die steuernde Entität in der Ausführung nicht abgebrochen werden.

Das Kommando SICCT COMFORT ENROLL darf nicht im Betriebsmodus BCS unterstützt werden.

SICCT COMFORT ENROLL							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
BCS	SICCT				Stage		
					1	2	3
no	✓	✓	no	no	no	no	no

5.22.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	[Lc]	[Data]	[Le]
SICCT COMFORT ENROLL	'80'	'22'	Functional Unit	Command Qualifier	Length Command Data	Command Data	Length Re- quested Data
	▪ CLA = Class ▪ INS = Instruction ▪ P1, P2 = Parameter 1 and 2 ▪ Lc = Length of command data field ▪ Le = Length of expected ▪ SW1, SW2 = Status Bytes						

5.23 Command SICCT DOWNLOAD INIT

5.23.1 Funktion

Die Kommandos SICCT DOWNLOAD INIT, SICCT DOWNLOAD DATA und SICCT DOWNLOAD FINISH müssen eine Updatemöglichkeit der SICCT-Kartenterminal-Firmware (FW) über die Kommandoschnittstelle implementieren.

Für den Download der Firmware eines SICCT-Terminals muss ein einfacher Transportmechanismus bereitstehen.

Downloadobjekte und Formate dürfen herstellerspezifisch ausfallen. Das Datenformat des Downloadobjekts muss nicht standardisiert werden, da Gerätesoftware und Downloadobjekt stets von demselben Hersteller geliefert werden.

Die Kommandoschnittstelle stellt Mechanismen bereit, um den FW-Download in aufeinander folgenden Transferblöcken gesteuert ablaufen zu lassen.

Da der Firmwaretausch für jedes Gerät ein kritischer Vorgang ist, muss der gesamte Downloadvorgang vom Absender zeitüberwacht werden.

Diese Zeitüberwachung muss variabel gestaltet sein, da z.B. die Lösch- und Programmierzeiten von Flashbausteinen und auch eine Signaturprüfung unterschiedlich lange Laufzeiten beinhalten können, die für den proprietären Ladealgorithmus im Gerät individuell gelten dürfen.

Im Falle eines Timeouts muss der Download abgebrochen werden. Wenn das Gerät nach dem Abbruch des Download-Vorgangs funktionsfähig geblieben ist, muss der Betrieb des Geräts ohne Update fortgesetzt werden können und der Download kann wiederholt werden.

5.23.1.1 Ablauf eines FW-Downloads

Folgende Abbildung zeigt den beispielhaften Ablauf eines FW-Downloads unter Nutzung der SICCT-Download-Kommandos sowie bei Verwendung einer optionalen Absicherung der Datenpakete durch eine Hashwertübermittlung und Prüfung im SICCT-Kartenterminal.

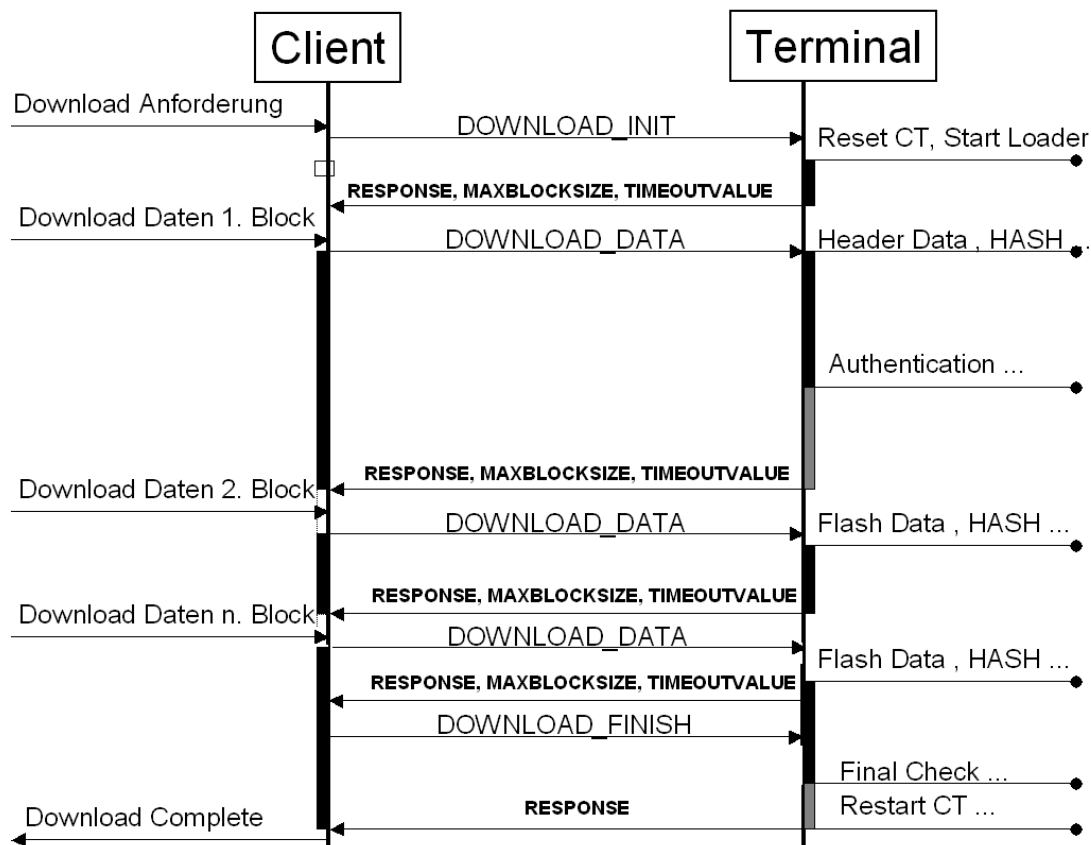


Abbildung 4: Beispielablauf des Firmware-Downloads

5.23.1.2 Download-Fehlerbehandlung

Folgende Abbildung zeigt den beispielhaften Ablauf einer Fehlerbehandlung eines FW-Downloads unter Nutzung der SICCT-Download-Kommandos sowie bei Verwendung einer optionalen Absicherung der Datenpakete durch eine Hashwertübermittlung und Prüfung im SICCT-Kartenterminal.

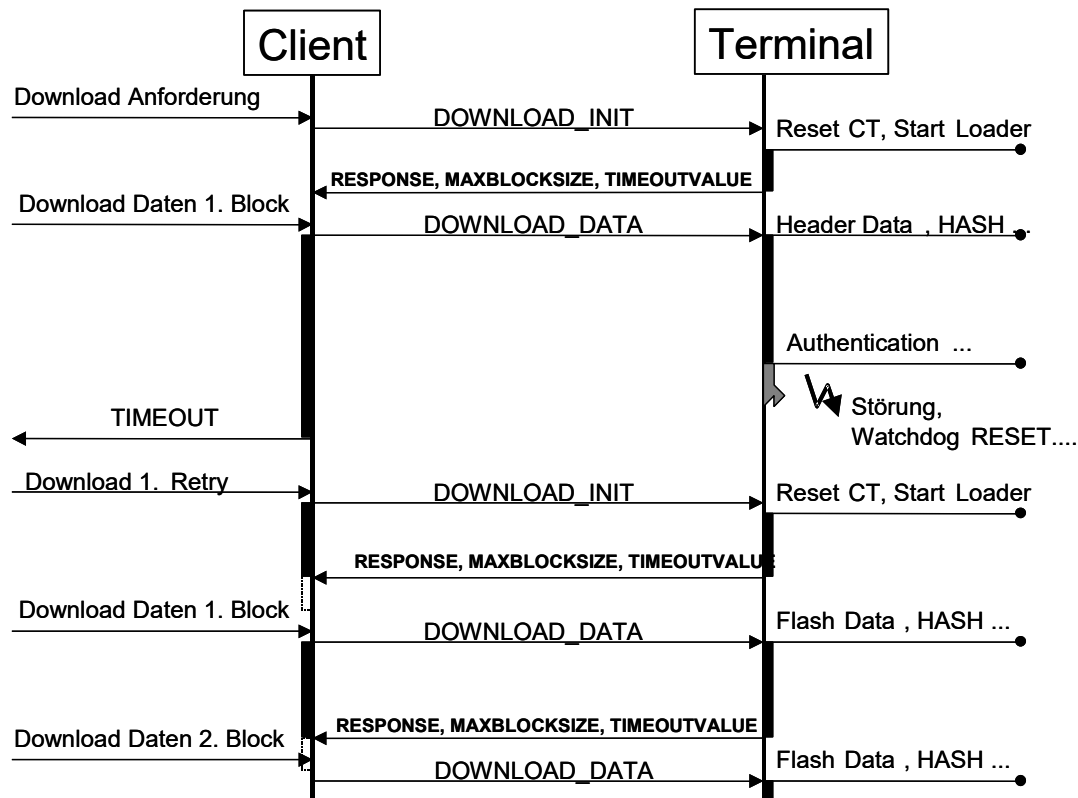


Abbildung 5: Beispielablauf der Fehlerbehandlung beim Firmware-Download

Das Kommando SICCT DOWNLOAD INIT kann einen Firmware-Download-Mechanismus am SICCT-Kartenterminal initiieren. Nach erfolgreicher Ausführung muss eine sog. Download-(DL) Session beginnen, welche bis zum SICCT CT DOWNLOAD FINISH Kommando, Fehlerfall oder Abbruch andauern muss. Innerhalb einer DOWNLOAD-Session darf das SICCT-Kartenterminal nur die Kommandos akzeptieren

- SICCT DOWNLOAD INIT (Wiederholung des Kommandos nach Eröffnung der DL-Session)
- SICCT DOWNLOAD DATA
- SICCT DOWNLOAD FINISH

Vor dem Eintritt in eine Download-Session muss das SICCT-Kartenterminal einen Geräte-Reset wie bei einer Ausführung eines SICCT RESET CT Kommandos herstellen, d.h. u.a. alle laufenden Transaktionen beenden sowie alle Chipkarten (RFID-Token, PICC) deaktivieren. Abschließend muss das SICCT-Terminal das Kommando mit einem Antwortdatensatz, bestehend aus den zurückgegebenen Download-Parametern 'RESPONSE, MAXBLOCKSIZE' und 'TIMEOUTVALUE', quittieren

Diese Parameter müssen dabei folgende Bedeutung und Codierung aufweisen:

RESPONSE wie bei allen Kommandos kodiert "OK, Fehler, Busy" auf das vorhergehende Kommando.

MAXBLOCKSIZE, unsigned integer 16 Bit.

Der Absender des Downloads darf maximal MAXBLOCKSIZE Bytes Downloaddaten im folgenden Block innerhalb des Elements „Download Data Package“ von DLDATA DO senden.

TIMEOUTVALUE, unsigned integer 16 Bit. Zeit in Sekunden.

Dieser Parameter wird für die Zeitüberwachung des folgenden Vorgangs vom Absender verwendet. Der Wert 0000 hält den Timeout Timer an.

Erst nach Initiierung der Download Session können Datenblöcke mittels des Kommandos DOWNLOAD_DATA an das SICCT-Terminal übertragen werden. Nach jedem Datenblock muss die Verarbeitung im SICCT-Terminal erfolgen.

Nach der Verarbeitung muss das SICCT-Terminal den erhaltenen Datenblock quittieren und eine RESPONSE an den Aufrufer senden sowie aktuelle Werte für MAXBLOCKSIZE und TIMEOUTVALUE setzen.

Das Ende der Datenübertragung muss mit dem Kommando DOWNLOAD_FINISH gekennzeichnet werden.

Daraufhin kann das Gerät mit dem Abschluss des Downloads mit ggf. einer finalen Gültigkeitsprüfung beginnen und den Abschluss der DL Session vornehmen.

Nach Ende der Prüfung des Downloadobjekts muss der anschließende Versand einer finalen RESPONSE erfolgen, die dem Aufrufer das Ende der DL Session anzeigen muss.

Ausführungsphasen SICCT DOWNLOAD INIT				
Phase		Stage	option	Description
preprocessing phase	Vorbereitungsphase	1		▪ display message
processing phase	Ausführungsphase	2		▪ init download session
postprocessing phase	Nachbereitungsphase	3		▪ display message ▪ send return value

5.23.2 Anwendungsbedingungen

SICCT DOWNLOAD INIT muss das erste Kommando sein und muss den Start der Download-Session einleiten, sofern das SICCT-Terminal dieses innerhalb einer CT ADMIN Session zulassen kann und nicht bereits anderweitig beschäftigt / belegt war.

SICCT DOWNLOAD INIT							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
BCS	SICCT			Stage			
				1	2	3	
no	✓	✓	no	no	no	no	no

5.23.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	[Lc]	[Data]	Le
SICCT DOWNLOAD INIT	'80'	'24'	Functional Unit	Command Qualifier	[Length Command Data]	[Command Data]	Length Re- quested Data
	▪ CLA = Class ▪ INS = Instruction ▪ P1, P2 = Parameter 1 and 2 ▪ Lc = Length of command data field ▪ Le = Length of expected ▪ SW1, SW2 = Status Bytes				Case 2 (no cmd data, rsp data): no Lc, Le=1-256 Bytes		
					Case 4 (cmd data, rsp data): Lc=1-255 Bytes or extended Lc Le=1-256 Bytes or extended Le		

Specification C-APDU			
CLA	'80'	Cardterminal Command Class	
INS	'24'	SICCT DOWNLOAD INIT	
P1	Functional Unit		
	bit8...bit1	Referenced Coding	
		'FF'	Escape: Signal Referenced Coding of P1 Functional Unit referenced by Functional Unit Index Data Object. FUI DO contained within Command Data Field.
	bit8...bit1	Direct Coding (mandatory)	
		'00'	Address Cardterminal
			other values RFU

P2	Command Qualifier		
	bit8...bit1	'00'	other values RFU

Lc	Length of Command Data Nc		
	Empty	Lc absent	no Command Data provided; Nc = 0
	variable length	Length of Command Data Nc (no. of bytes contained in Data field)	
		Lc short '01' <= Lc <= 'FF'	1 <= Nc <= 255
		Lc extended 3 Byte Coding '000001' <= Lc <= '00FFFF'	1 <= Nc <= 65535

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	Empty	non-existent: in case Nc = '0': no Command Data provided
	In case of Referenced Coding of 'P1'	
	FUI DO	Functional Unit Index Data Object

Le	Length of Requested Data Ne Return up to Ne bytes of requested information		
	variable length	Condition: Lc short Le short '01' <= Le <= 'FF'	1 <= Ne <= 255
		Condition: Lc short Le short Le = '00'	Ne = 256
		Condition: Lc absent Le extended '000001' <= Le <= '00FFFF'	1 <= Ne <= 65535
		Condition: Lc absent Le extended Le = '000000'	Ne = 65536
		Condition: Lc extended Le extended '0001' <= Le <= 'FFFF'	1 <= Ne <= 65535
		Condition: Lc extended Le extended Le = '0000'	Ne = 65536

5.23.4 Data Objects

Das SICCT DOWNLOAD INIT Kommando arbeitet mit den folgenden Daten Objekten.

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10.9
DLPARAM DO	RSP	Download Parameter Data Object	see 5.5.10.24

5.23.5 Response Structure

SICCT DOWNLOAD INIT	Kodierung R-APDU			
	[Body]			Trailer
	[Requested Data / Information]			Status Byte 1 Status Byte 2
	Empty	in case invalid parameter 'P1' / 'P2'	SW1	SW2
		in case Lc was invalid		
		in case Le was invalid or too small		
		in case of error		
	DLPARAM DO	Download Parameter Data Object		

5.23.6 Status-Codes Sw1SW2

SW1SW2	Addressed Functional Unit P1	Priority, Priorität	Specification	Meaning / Bedeutung
'6200'	CT	11	Warning -	Download - Session already started.
			Download - Session already started.	
'6400'		12	Reset not successful	- Command execution error: - Error occurred during cardterminal / FU reset. - State of terminal and functional units (FU) possibly unchanged.
'6501'		11	Memory Failure	- Terminal internal error: - Flash-Memory or memory capacity error
'6700'		3	Wrong (command) length	Wrong Lc: Inconsistent command body or Command Data field length error.
'6900'		9	Command not allowed.	- The terminal cannot perform the command because - No open / valid CT Session - Access Rights not fulfilled (no Admin CT Session) - No open / valid TLS connection - Cardterminal busy, - Download - Session cannot start.
				- At the moment FU_CT is busy / unavailable. - Funktionseinheit FU_CT ist z.Zt. belegt / nicht verfügbar
'6A00'		5	Wrong parameters P1, P2	- P1 addresses invalid Functional Unit. - In case of Referenced Coding – - Invalid / ungültiges FUI DO referenced by P1 (command data). - P2 specifies not supported value
'6A80'		6	Invalid Data Object	- Incorrect data objects / parameters in the command data field - In case of Referenced Coding – - The data field contains an invalid DO / - Das Datenfeld enthält ein unzulässiges DO. - Invalid FUI DO / Unzulässiges FUI DO
'6A88'		7	Missing Data Object	- In case of Referenced Coding – - The data field does not contain a FUI DO / - Das Datenfeld enthält kein FUI DO.
'6A89'		8	Too many Data Objects	- Too many Data Objects (FUI DO) of same type. - Es sind zu viele Datenobjekte gleichen Typs (FUI DO, ...) enthalten
'6C00'		4	Wrong Length Le	Wrong Le / falscher Wert Le:
'6D00'		2	Wrong Instruction	Wrong / unbekanntes Instruction-Byte
'6E00'		1	Class not supported	Falsches / unbekanntes Class-Byte
'6F00'		11	Communication with CT not possible	- In case - the terminal detects general protocol errors or - the terminal detects errors within CT Session or TLS connection
'9000'		12	Command successful	- Download Initialisation was successful. - Cardterminal and all functional units successfully set to reset state.

5.24 Command SICCT DOWNLOAD DATA

5.24.1 Funktion

Das SICCT DOWNLOAD DATA Kommando muss einzelne Download-Pakete während einer Download-Session an das SICCT-Kartenterminal übertragen können.

Ausführungsphasen SICCT DOWNLOAD DATA				
Phase		Stage	option	Description
preprocessing phase	Vorbereitungsphase	1		▪ display message
processing phase	Ausführungsphase	2		▪ process download data
postprocessing phase	Nachbereitungsphase	3		▪ display message ▪ send return value

5.24.2 Anwendungsbedingungen

Das SICCT DOWNLOAD DATA Kommando darf nur innerhalb einer Download-Session, d.h. nach einem initialen SICCT DOWNLOAD INIT Kommando akzeptiert werden.

SICCT DOWNLOAD DATA							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
BCS	SICCT			Stage			
				1	2	3	
no	✓	✓	no	no	no	no	no

5.24.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	Lc	Data	Le
SICCT DOWNLOAD DATA	'80'	'25'	Functional Unit	Command Qualifier	Length Command Data	Command Data	Length Requested Data
	- CLA = Class - INS = Instruction - P1, P2 = Parameter 1 and 2 - Lc = Length of command data field - Le = Length of expected - SW1, SW2 = Status Bytes				Case 4 (cmd data, rsp data): Lc=1-255 Bytes or extended Lc Le=1-256 Bytes or extended Le		

Specification C-APDU		
CLA	'80'	Cardterminal Command Class
INS	'25'	SICCT DOWNLOAD Data

P1	Functional Unit		
	bit8...bit1	Referenced Coding	
		'FF'	Escape: Signal Referenced Coding of P1 Functional Unit referenced by Functional Unit Index Data Object. FUI DO contained within Command Data Field.
	bit8...bit1	Direct Coding (mandatory)	
		'00'	Address Cardterminal other values RFU

P2	Command Qualifier		
	bit8...bit1	'00'	other values RFU

Lc	Length of Command Data Nc
----	---------------------------

	variable length	Length of Command Data Nc (no. of bytes contained in Data field)	
		Lc short '01' <= Lc <= 'FF'	1 <= Nc <= 255
		Lc extended 3 Byte Coding '000001' <= Lc <= '00FFFF'	1 <= Nc <= 65535

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	DLDATA DO	Download Data Object
	In case of Referenced Coding of 'P1'	
	FUI DO	Functional Unit Index Data Object
	DLDATA DO	Download Data Object

Le	variable length	Length of Requested Data Ne Return up to Ne bytes of requested information	
		Condition: Lc short Le short '01' <= Le <= 'FF'	1 <= Ne <= 255
		Condition: Lc short Le short Le = '00'	Ne = 256
		Condition: Lc extended Le extended '0001' <= Le <= 'FFFF'	1 <= Ne <= 65535
		Condition: Lc extended Le Extended Le = '0000'	Ne = 65536

5.24.4 Data Objects

Das SICCT DOWNLOAD DATA Kommando kann mit den folgenden Daten Objekten arbeiten.

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10.9
DLDATA DO	CMD	Download Data Object	see 5.5.10.25
DLPARAM DO	RSP	Download Parameter Data Object	see 5.5.10.24

5.24.5 Response Structure

SICCT DOWNLOAD DATA	Kodierung R-APDU				
	[Body]			Trailer	
	[Requested Data / Information]			Status Byte 1	Status Byte 2
	Empty	in case invalid parameter 'P1' / 'P2'		SW1	SW2
		in case Lc was invalid			
		in case Le was invalid or too small			
		in case of error			
DLPARAM DO	Download Parameter Data Object				

5.24.6 Status-Codes Sw1SW2

SW1SW2	Addressed Functional Unit	Priority Priorität	Specification	Meaning / Bedeutung
	P1			
'6400'		12	Command execution error	- Command execution error: - Error occurred during reception of DLDATA DO.
'6501'		11	Memory Failure	- Terminal internal error: - Flash-Memory or memory capacity error
'6700'		3	Wrong (command) length	Wrong Lc: Inconsistent command body or Command Data field length error.
'6900'		9	Command not allowed.	- The terminal cannot perform the command because has there is no valid (open) - No open / valid / pending CT Session - No open / valid TLS connection - Cardterminal busy, - Download - Session cannot start.
'6A00'		5	Wrong parameters P1, P2	- P1 addresses invalid Functional Unit. - In case of Referenced Coding – - Invalid / ungültiges FUI DO referenced by P1 (command data). - P2 specifies not supported value
'6A80'		6	Invalid Data Object	- Incorrect data objects / parameters in the command data field - In case of Referenced Coding – - The data field contains an invalid DO / Das Datenfeld enthält ein unzulässiges DO. - Invalid FUI DO / Unzulässiges FUI DO - Invalid DLDATA DO / Unzulässiges DLDATA DO
'6A88'		7	Missing Data Object	- In case of Referenced Coding – - The data field does not contain a FUI DO / Das Datenfeld enthält kein FUI DO. - Missing DLDATA DO / Fehlendes DLDATA DO
'6A89'		8	Too many Data Objects	- Too many Data Objects (FUI DO, DLDATA DO) of same type. - Es sind zu viele Datenobjekte gleichen Typs (FUI DO, DLDATA DO...) enthalten
'6C00'		4	Wrong Length Le	Wrong Le / falscher Wert Le:
'6D00'		2	Wrong Instruction	Wrong / unbekanntes Instruction-Byte
'6E00'		1	Class not supported	Falsches / unbekanntes Class-Byte
'6F00'		11	Communication with CT not possible	- In case - the terminal detects general protocol errors or - the terminal detects errors within CT Session or TLS connection
'9000'		12	Command successful	- Download of Download Data Object was successful. - Transmission / reception of DLDATA DO was successful.

5.25 Command SICCT DOWNLOAD FINISH

5.25.1 Funktion

Das SICCT DOWNLOAD FINISH Kommando kann eine bestehende Download-Session des SICCT-Kartenterminals beenden.

Ausführungsphasen SICCT DOWNLOAD FINISH				
Phase		Stage	option	Description
preprocessing phase	Vorbereitungsphase	1		▪ display message
processing phase	Ausführungsphase	2		▪ finish download process
postprocessing phase	Nachbereitungsphase	3		▪ display message ▪ send return value

5.25.2 Anwendungsbedingungen

Das SICCT CT DOWNLOAD FINISH Kommando darf nur innerhalb einer Download-Session, d.h. nach einem SICCT DOWNLOAD INIT bzw. SICCT DOWNLOAD DATA Kommando gegeben werden.

SICCT DOWNLOAD FINISH							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable by SICCT CONTROL COMMAND			
BCS	SICCT			Stage			
				1	2	3	
no	✓	✓	no	no	no	no	no

5.25.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	[Lc]	[Data]	Le
SICCT DOWNLOAD FINISH	'80'	'26'	Functional Unit	Command Qualifier	[Length Command Data]	[Length Command Data]	Length Requested Data
	▪ CLA = Class ▪ INS = Instruction ▪ P1, P2 = Parameter 1 and 2 ▪ Lc = Length of command data field ▪ Le = Length of expected ▪ SW1, SW2 = Status Bytes				Case 2 (no cmd data, rsp data): no Lc, Le=1-256 Bytes		
					Case 4 (cmd data, rsp data): Lc=1-255 Bytes or extended Lc Le=1-256 Bytes or extended Le		

Specification C-APDU		
CLA	'80'	Cardterminal Command Class
INS	'26'	SICCT DOWNLOAD Finish

P1	Functional Unit		
	bit8...bit1	Referenced Coding	
		'FF'	Escape: Signal Referenced Coding of P1 Functional Unit referenced by Functional Unit Index Data Object. FUI DO contained within Command Data Field.
	bit8...bit1	Direct Coding (mandatory)	
		'00'	Address Cardterminal other values RFU

P2	Command Qualifier		
	bit8...bit1	'00'	other values RFU

Lc	Length of Command Data Nc		
	Empty	Lc absent	no Command Data provided; Nc = 0
	Length of Command Data Nc		
	variable length	Length of Command Data Nc (no. of bytes contained in Data field)	
		Lc short '01' <= Lc <= 'FF'	1 <= Nc <= 255
		Lc extended 3 Byte Coding '000001' <= Lc <= '00FFFF'	1 <= Nc <= 65535

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	Empty	non-existent: in case Nc = '00': no Command Data provided
	In case of Referenced Coding of 'P1'	
	FUI DO	Functional Unit Index Data Object

Le	Length of Requested Data Ne Return up to Ne bytes of requested information		
	variable length	Condition: Lc short Le short '01' <= Le <= 'FF'	1 <= Ne <= 255
		Condition: Lc short Le short Le = '00'	Ne = 256
		Condition: Lc extended Le extended '0001' <= Le <= 'FFFF'	1 <= Ne <= 65535
		Condition: Lc extended Le Extended Le = '0000'	Ne = 65536

5.25.4 Data Objects

Das SICCT DOWNLOAD FINISH Kommando kann mit den folgenden Daten Objekten arbeiten.

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10.9
DLTERM DO	RSP	Download Termination Data Object	see 5.5.10.26

5.25.5 Response Structure

SICCT DOWNLOAD FINISH	Kodierung R-APDU			
	[Body]			Trailer
	[Requested Data / Information]			Status Byte 1 Status Byte 2
	Empty	in case invalid parameter 'P1' / 'P2'	SW1	SW2
		in case Lc was invalid		
		in case Le was invalid or too small		
		in case of error		
	DLTERM DO	Download Termination Data Object		

5.25.6 Status-Codes Sw1SW2

SW1SW2	Addressed Functional Unit	Priority Priorität	Specification	Meaning / Bedeutung
	P1			
'6400'		12	Command execution error	- Command execution error: - Error occurred during finishing / closing of DL-Session.
'6501'		11	Memory Failure	- Terminal internal error: - Flash-Memory or memory capacity error - Terminal internal error: - Flash-Memory or memory capacity error
'6700'		3	Wrong (command) length	- Wrong Lc: Inconsistent command body or Command Data field length error. - Message too long.
'6900'		9	Command not allowed.	- The terminal cannot perform the command because has there is no valid (open) - No open / valid / pending CT Session - No open / valid TLS connection - Cardterminal busy, - Download - Session cannot start.
'6A00'		5	Wrong parameters P1, P2	- P1 addresses invalid Functional Unit. - In case of Referenced Coding – - Invalid / ungültiges FUI DO referenced by P1 (command data). - P2 specifies not supported value
'6A80'		6	Invalid Data Object	- Incorrect data objects / parameters in the command data field - In case of Referenced Coding – - The data field contains an invalid DO / - Das Datenfeld enthält ein unzulässiges DO. - Invalid FUI DO / Unzulässiges FUI DO - Invalid DLDAPDO / Unzulässiges DLDAPDO
'6A88'		7	Missing Data Object	- In case of Referenced Coding – - The data field does not contain a FUI DO / - Das Datenfeld enthält kein FUI DO. - Missing DLDAPDO / Fehlendes DLDAPDO
'6A89'		8	Too many Data Objects	- Too many Data Objects (FUI DO, DLDAPDO) of same type. - Es sind zu viele Datenobjekte gleichen Typs (FUI DO, DLDAPDO...) enthalten
'6C00'		4	Wrong Length Le	Wrong Le / falscher Wert Le:
'6D00'		2	Wrong Instruction	Wrong / unbekanntes Instruction-Byte
'6E00'		1	Class not supported	Falsches / unbekanntes Class-Byte
'6F00'		11	Communication with CT not possible	- In case - the terminal detects general protocol errors or - the terminal detects errors within CT Session or TLS connection
'9000'		12	Command successful	- Download Termination was successful. - DL-Session closed.

6 Schnittstellenbeschreibung zum Host

6.1 Allgemeines

Die verschiedenen Anschlussmöglichkeiten eines SICCT-Terminals erfordern unterschiedliche Mechanismen zur Adressierung des SICCT-Terminals, dem Kommandotransport und dessen Sicherung in Hinblick auf Datenintegrität und -authentizität, sowie zur Bereitstellung administrativen Schnittstellen.

Nachfolgend werden die Anforderungen an den Betrieb eines SICCT-Terminals mit Schwerpunkt auf die Anschlussmöglichkeit Ethernet beschrieben, die ein SICCT-Terminal erfüllen muss. Andere Übertragungswege sind noch nicht spezifiziert.

6.2 Anschlussmöglichkeit Ethernet

Das Nutzen eines SICCT-Terminals in einem IP-basierten Netzwerk erfordert:

1. IP Adress- und DNS-Namensvergabe am SICCT-Terminal:
Das SICCT-Terminal erhält eine IP Adresse und kann optional auch über einen DNS Namen ansprechbar sein. Des Weiteren werden andere zur Kommunikation notwendige Netzwerkparameter gesetzt.
2. Dynamisches Auffinden von SICCT-Terminals durch eine SICCT konforme Applikation:
 - 1 Dienstanfrage (Service Discovery): Ein Broadcast basiertes Protokoll ermöglicht es Applikationen, die aktuellen IP Adressen und Namen der im Netz verfügbaren SICCT-Terminals zu erhalten. Dieser Schritt muss typischerweise vor der eigentlichen Kommunikation zwischen Anwendung und SICCT-Terminal mittels der im Command Set definierten Kommandos ausgeführt werden.
 - 2 Bekanntmachen eines Dienstes (Service Announcement): SICCT-Terminals können zusätzlich die Möglichkeit unterstützen, sich selbstständig bei interessierten Clients bekannt zu machen.
3. SICCT Protokoll Kommandos für den TCP Datenstrom. Diese kodieren, übertragen und dekodieren (C- und R-) Kommandos.
4. Ereignisse, welche am SICCT-Terminal aufgetretene Zustandsänderungen und Fehlerbedingungen via TCP an den Client signalisieren.

6.2.1 Netzwerk Protokoll Festlegungen

Das Protokoll muss auf den Ebenen 5 bis 7 (Sitzungs-, Darstellungsschicht, Anwendungsschicht) des ISO/OSI-7-Schichtenmodells [STD18] bzw. der Anwendungsschicht des TCP/IP-Referenzmodells [RFC1122] angesiedelt sein, um den Einsatz in möglichst jedem Netz zuzulassen.

Der Schwerpunkt liegt bei Ethernet/IP basierten Netzen; für diese gilt im Speziellen:

Übertragungsschicht:

Ethernet Adapter müssen folgende Anforderungen erfüllen:

- 10 Mbit verpflichtend
- 100 Mbit und/oder 1Gbit optional
- Auto-negotiation Mode.

Vermittlungsschicht:

Als Protokoll der Vermittlungsschicht muss IPv4 ([RFC791] und Erweiterungen) vorausgesetzt werden.

Die Adresse der Vermittlungsschicht bzw. der DNS Name des SICCT-Terminals dienen zum eindeutigen Auffinden eines SICCT-Terminals im Netz.

Transportschicht:

Die Protokoll-Suiten TCP und UDP dienen auf der Transportschicht dem Kommandotransport bzw. dem Service Discovery.

Anwendungsschicht:

Die Protokolle zum Auffinden eines SICCT-Terminals sowie zur Kommunikation mit einem SICCT-Terminal müssen in der Anwendungsschicht angesiedelt sein; dazu gehören:

DHCP/DNS – IP Adressen und Namensvergabe (siehe 6.2.3 Service Discovery)

Command Set – Das Protokoll muss die zu übertragenden Daten und Codes zur Beschreibung der auszuführenden Aktionen (siehe Kapitel 5 "Command Set") beinhalten.

Zeitkritische Kommandos/Timeouts und Fehlerbehandlung:

Die Einhaltung zeitlicher Grenzen bei der Kommunikation muss durch die jeweiligen Protokolle der durchwanderten Netzwerkschicht sichergestellt werden.

Die Behandlung von Verbindungs- und Übertragungsfehlern werden nach Möglichkeit von den Netzwerkprotokollen der Schichten 1 bis 4 behandelt.

Die Behandlung SICCT Protokoll spezifischer Fehler wird in 6.2.4.6 "Fehlerbehandlung" beschrieben.

6.2.2 Adress- und Namensvergabe

Ein SICCT-Terminal muss seine Netzwerkkonfigurationsdaten durch einen der folgenden Mechanismen erhalten können:

- dynamische Konfiguration
 - DHCP [RFC2131]
 - AUTO-IP [RFC3927]
- statische Konfiguration

Die Konfigurationsdaten müssen folgende Werte umfassen:

- 1) die IP Adresse des SICCT-Terminals
- 2) Subnetzmaske des lokalen Netzwerks
- 3) IP Adresse des Default Gateway
- 4) SICCT-Terminal Hostname (DNS Name)
- 5) optional: DNS Server IP Adresse(n)

ANMERKUNG: Die Broadcast Adresse des lokalen Netzwerks, wie sie für das Service Discovery Protokoll benötigt wird, kann aus der IP Adresse und der Subnetzmaske errechnet werden und muss daher nicht vorgegeben werden.

DHCP: Bei Verwendung eines DHCP Servers müssen dem SICCT-Terminal durch diesen alle nötigen Daten entweder aufgrund von statischen Listen oder dynamisch zugewiesen werden.

Auto-IP, das der konfigurationsfreien Vernetzung von Geräte in lokalen Rechnernetzen dient, beschreibt einen auf dem ARP Protokoll basierenden IP-Allokationsmechanismus [RFC3927], der ohne zentrale Instanz wie einem DHCP Server versucht, eine im Netzwerk eindeutige IP Adresse auszuwählen: Hierfür muss mit Hilfe eines Zufallszahlengenerators am SICCT-Terminal eine IP Adresse aus dem von der IANA reservierten Bereich 169.254.1.0 bis 169.254.254.255 [RFC 3330] ausgewählt werden. Der [RFC3927] legt auch die Anzahl und zeitliche Abfolge der Adressierungsversuche, sowie das Verhalten bei Adresskonflikten fest.

Statische Konfiguration: Die statische Adresskonfiguration muss über eine der vom SICCT-Terminal implementierten Managementschnittstellen durch einen Administrator vorgegeben werden. Die Oberfläche muss die Eingabe der oben genannten Konfigurationsdaten anbieten.

Namensauflösung: Die Erreichbarkeit eines SICCT-Terminals über seinen Hostnamen ist immer vom Vorhandensein einer entsprechenden Infrastruktur im Netzwerk abhängig. In Netzen mit DHCP und DNS Server kann der DNS Name statisch am DNS Server eingetragen sein, und optional mittels DHCP

an das SICCT-Terminal übermittelt werden. Bietet der DHCP Server auch die DNS-UPDATE [RFC2136] Erweiterungen an, so ist auch die umgekehrte Richtung möglich: das SICCT-Terminal selbst gibt seinen eigenen Hostname über das DHCP Protokoll an den DNS Server weiter. Bei statischer Konfiguration des SICCT-Terminals muss auch der Hostname am DNS vorkonfiguriert sein. Netze, in denen der Auto-IP Mechanismus zum Einsatz kommen muss, haben keine DHCP Server. DNS Server sind meist ebenso nicht vorhanden oder müssen ebenfalls vorkonfiguriert sein.

Ablauf der Netzwerkkonfiguration: Ein SICCT-Terminal muss entweder für die Verwendung der statischen Netzwerkkonfiguration oder für automatische Adressierung eingestellt sein. Welcher der beiden Mechanismen zur Anwendung kommt, muss über die Managementschnittstellen einstellbar sein. Im Kapitel 6.3.2 Managementschnittstellen sind die Standardwerte festgelegt.

Bei der statischen Konfiguration trägt der Administrator die Verantwortung für die Korrektheit der Einstellungen. Falsche Einträge können dazu führen, dass das SICCT-Terminal nicht mehr über das Netzwerk ansprechbar ist.

Bei dynamischer Konfiguration des SICCT-Terminals wird zuerst versucht, die Konfigurationsdaten über das DHCP Protokoll zu erhalten. Antwortet innerhalb einer bestimmten Zeit kein DHCP Server, so muss vorerst angenommen werden, dass keiner vorhanden ist. Als nächstes kann der Auto-IP Mechanismus eingesetzt werden, um selbständig die Konfigurationsdaten zu setzen. Dieser Mechanismus schreibt auch vor, dass in bestimmten Abständen wiederum versucht werden muss, per DHCP Adressdaten zu erhalten.

Die Fehlerbehandlung (vor allem die Behandlung von Adresskonflikten sowie HW Fehlern) erfolgt im TCP/IP Stack bzw. durch die Implementierungen des DHCP Protokolls und des Auto-IP Mechanismus.

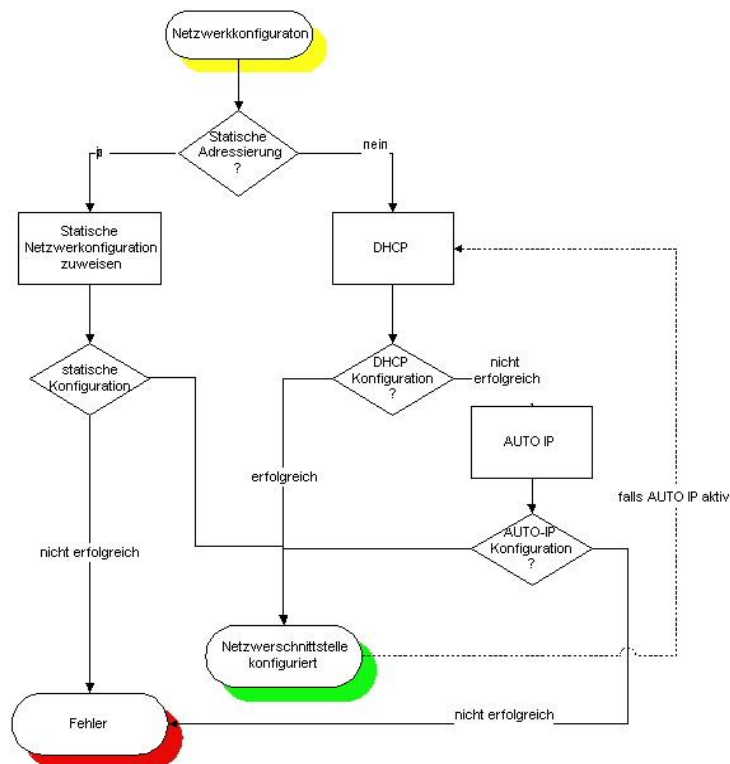


Abbildung 6 Netzwerkkonfiguration

6.2.3 Service Discovery/Announcement

Zum Auffinden eines SICCT-Terminals im Netz durch ein SICCT kompatibles Anwendungsprogramm muss ein UDP/IP ([RFC768] / [RFC 791]) basiertes Protokoll verwendet werden. Das SICCT-Terminal übernimmt hier die Rolle des Diensteanbieters (Server), das Anwendungsprogramm die Rolle des Anfragestellers (Client). Prinzipiell unterstützt das Protokoll die vom Client initiierte Suche nach SICCT-Terminals (Service Discovery), sowie die Bekanntmachung durch das Terminal selbst (Service An-

nouncement). Für das dynamische Auffinden von SICCT-Terminals muss das SICCT- Terminal den Service Discovery Mechanismus implementieren; die Unterstützung des Service Announcement ist optional.

Dieses Dokument beschreibt die Version 1.30 des Discovery-Protokolls.

6.2.3.1 Dienstanfrage (Service Discovery)

ANMERKUNG Dienstanfrage bezeichnet den Mechanismus, der es Clients ermöglicht im Netz verfügbare SICCT-Terminals dynamisch zu finden.

Versand der Dienstanfrage:

Die Dienstanfrage muss vom Client aus als UDP Broadcast an den Dienstanfrageport (Standard: UDP Port 4742) der SICCT-Terminals erfolgen. Der Broadcast kann entweder als gerichteter oder als lokaler Broadcast abgesetzt werden. Die Zieladresse bestimmt die Art des Broadcast:

- Für einen ungerichteten Broadcast im lokalen Subnetz muss die Adresse 255.255.255.255 verwendet.
- Für einen gerichteten Broadcast im lokalen Subnetz kann die Broadcastadresse aus der IP Adresse und der Subnetzmaske der Netzwerkschnittstelle des Clients berechnet werden.
- Beim Einsatz eines gerichteten Broadcast in ein anderes Subnetz muss die Broadcastadresse vorgegeben werden. Zusätzlich muss durch die Administration des Netzes sichergestellt werden, dass einerseits der Broadcast bis zum Zielnetzwerk geroutet wird, und andererseits Datenpakete antwortenden SICCT-Terminals retour gesendet werden können. Die entsprechende Konfiguration des Netzwerks und seiner Komponenten (Router und Firewall Konfiguration der am Weg liegenden Netze) ist nicht Gegenstand dieses Protokolls.

Bei der Verwendung vom Broadcast empfängt der Client üblicherweise das Broadcast-Paket selbst nicht mehr. Das Auffinden eines "lokal" am Client laufenden SICCT Interpreters muss bei diesem Mechanismus gesondert behandelt werden.

Datenformat des Dienstanfragepakets:

Die Daten des von der Client Anwendung versendeten Dienstanfragepakets müssen wie folgt codierte TLV Objekte sein:

Protokoll Version 1.30: Dienstanfragepaket					
Datenfeld		TAG (hex)	Datenlänge (Bytes)	zwingend er- forderlich	Beschreibung
Pakettyp		'A0'	m	ja	Dienstanfragepakettyp
m Bytes	Client IP Ad- resse	'81'	4	ja	IP Adresse des Clients in network-byte-order.
	Protokoll Ver- sion	'80'	2	ja	Versionsnummer der Discovery-Protokollspezifikation in network-byte-order.
	Client Port	'82'	2	ja	UDP Port am Client in network-byte-order an dem die Rückantwort der SICCT-Terminals erwartet wird (standardmäßig UDP Port 4742).

Tabelle 27 Dienstanfragepaket

- **Protokoll Version:**
Das Versionsfeld gibt Auskunft über die im Dienstanfragepaket vorhandenen Datenfelder, sowie über das Format des vom Client zurückerwarteten Dienstbeschreibungspakets.
- **Client IP Adresse/Client Port:**
Die für den Versand eines Dienstbeschreibungspakets nötige Adressinformation über den Anfragersteller.

Empfang des Dienstbeschreibungspakets:

Nach dem Versand einer Dienstanfrage muss der Client auf dem Dienstbeschreibungsport (Standard: UDP Port 4742) auf eingehende Dienstbeschreibungspakete der SICCT-Terminals warten. Der verwendete Port muss im Dienstanfragepaket immer angegeben werden. Es liegt im Ermessen des Host, wie lange auf Antwortpakete der SICCT-Terminals gewartet wird. Bei der Wahl der Wartezeit sind Geschwindigkeit und Auslastung der zu überbrückenden Netzwerke zu beachten. Für ein lokales 10Mbit Netzwerk sollte eine Wartezeit von 3 Sekunden ausreichen.

Jedes SICCT-Terminal, das ein Dienstanfragepaket empfängt und bereit ist, Anfragen über das SICCT Protokoll zu verarbeiten, sendet ein Dienstbeschreibungspaket mittels UDP/IP an den Client: Als Zieladresse und Zielpport müssen die im Dienstanfragepaket angegebenen Werte angegeben werden – für den UDP Port ist das standardmäßig 4742.

Datenformat des Dienstbeschreibungspakets:

Die Daten des von einem SICCT-Terminal versendeten Dienstbeschreibungspakets müssen wie folgt codierte TLV – Objekte sein:

Protokoll Version 1.30: Dienstbeschreibungspakets					
Datenfeld		TAG (hex)	Daten- länge (Bytes)	Datenfeld im Paket zwin- gend erfor- derlich	Beschreibung
Pakettyp		'A1'	m	ja	Dienstbeschreibungspakettyp
m Bytes	Protokoll Ver- sion	'80'	2	ja	Versionsnummer der Protokollspezifikation in network-byte-order.
	SICCT- Terminal IP Adresse	'81'	4	ja	IP Adresse des SICCT-Terminals in network-byte-order.
	SICCT- Terminal MAC Adresse	'83'	6	ja	MAC Adresse des SICCT-Terminals
	SICCT Terminalname	'84'	max. 32	ja	SICCT-Terminalname (ASCII Zeichenkette)
	SICCT Kommando- interpreter Port	'82'	2	ja	TCP/IP Port an dem der SICCT Kommandointerpreter auf eingehende SSL Verbindungen wartet (standardmäßig TCP Port 4742).
	Sicherheits- Protokoll	'A3'	n	nein	TLV codierte Beschreibung des zu verwendenden Sicherheitsprotokolls.

Tabelle 28 Dienstbeschreibungspaket

- 1) Protokoll Version:
Das Versionsfeld gibt Auskunft über die im Dienstbeschreibungspaket vorhandenen Datenfelder.
- 2) SICCT-Terminal IP Adresse:
IP Adresse des SICCT-Terminals auf dem der SICCT Kommandointerpreterdienst erreichbar ist.
- 3) SICCT-Terminal MAC Adresse:
MAC Adresse des SICCT-Terminals auf dem der SICCT Kommandointerpreterdienst erreichbar ist. Da eine korrekt vergebene MAC Adresse eindeutig sein muss, kann diese zur Wiedererkennung von SICCT-Terminals verwendet werden. Es gilt aber zu beachten, dass es keine Garantie für die Authentizität der gesendeten MAC Adresse gibt.
- 4) SICCT-Terminalname:
Der Hostname des SICCT-Terminals.
- 5) SICCT Kommandointerpreter Port:
Der TCP Port an dem der SICCT Interpreter auf eingehende Verbindungen wartet, muss immer im Feld "SICCT Kommandointerpreter Port" des Dienstbeschreibungspakets angegeben werden. Der SICCT Kommandointerpreter wartet standardmäßig auf dem TCP Port 4742 auf eingehende Clientverbindungen.
- 6) Sicherheitsprotokoll (optional):
Bei Vorhandensein muss das Feld die Beschreibung des zu verwendenden sicheren Protokolls enthalten. Wird kein Protokoll aufgelistet, so darf nur eine "unverschlüsselte" Verbindung unterstützt werden: Ein SICCT-Terminal, das unverschlüsselte Kommunikation nicht zuässt, listet dementsprechend nur das unterstützte sichere Protokoll. Ein SICCT-Terminal, das nur unverschlüsselte Verbindungen unterstützt, darf kein "Sicherheitsprotokoll listen".

Sicherheitsprotokolle:					
Protokoll	TAG (hex)	Datenlänge (Bytes)	Daten	Wert (hex)	Beschreibung
TLS	'8A'	1	Unterstützte Protokollversion (1 Byte)	'30'	TLS 1.2 gemäß [RFC5246]
				'40'	TLS 1.3 gemäß [RFC8446]
...	...	...	...	...	...

7) "Unverschlüsselt":

Eine "unverschlüsselte" Verbindung entspricht einer TCP/IP Verbindung ohne Anwendung weiterer Mechanismen zur Identifikation oder Authentifikation der Kommunikationspartner, sowie zur verschlüsselten Übertragung der Protokolldaten (SICCT Kommandos und Ereignisse).

8) TLS:

Wird das optionale Sicherheitsprotokoll TLS unterstützt, dann muss TLS 1.2 [RFC5246] unterstützt werden. TLS1.3 [RFC8446] kann optional unterstützt werden. Ältere TLS Cipher Suites dürfen nicht mehr unterstützt werden. Die Aushandlung der Cipher Suite sowie deren Parameter muss im Rahmen des TLS Handshake Verfahrens während des Aufbaus der SICCT Kommandointerpreterverbindung erfolgen. Für weitere Festlegungen siehe Kapitel 6.4.1.1 "TLS".

6.2.3.2 Bekanntmachung eines Dienstes (Service Announcement):

Ein SICCT-Terminal kann zusätzlich auch die Möglichkeit, sich selbst im Netzwerk bekannt zu machen, anbieten: Zu diesem Zweck sendet es ein Dienstbeschreibungspaket (siehe 6.2.3.1 "Diensteanfrage (Service Discovery)") mittels einem UDP/IP Broadcast an den Dienstankündigungsport (Service Announcement Port; Standard: UDP Port 4742).

Die Adresswahl/-berechnung für den Broadcast muss wie beim Versand einer Diensteanfrage erfolgen.

An Service Announcements interessierte Clients müssen am Service Announcement Port auf eingehende Broadcasts warten.

Das SICCT-Terminal erwartet keine Rückantwort auf gesendete Service Announcements von den Clients.

Das SICCT-Terminal darf Service Announcements zu jedem beliebigen Zeitpunkt versenden, im Speziellen aber immer nach dem Wirksamwerden relevanter Terminaleinstellungsänderungen (siehe Kapitel 6.3 "Terminal Managementverfahren"). Relevante Änderungen umfassen:

- Wechsel der IP-Adresse
- Wechsel des SICCT Kommandointerpreter Ports
- Änderung des SICCT-Terminal Hostnamens (DNS)

Die Änderung dieser Parameter muss durch entsprechende administrative Aktionen über eine der Managementschnittstellen erfolgen.

Das selbständige Versenden von Dienstbeschreibungspaketen durch die SICCT-Terminals ermöglicht es den Clients, schneller von der Anwesenheit des SICCT-Terminals bzw. den Änderungen am SICCT-Terminal zu erfahren, ohne selbst das Intervall der Service Discovery Broadcasts erhöhen zu müssen. Vom periodischen Versenden dieser Nachricht sollte abgesehen werden.

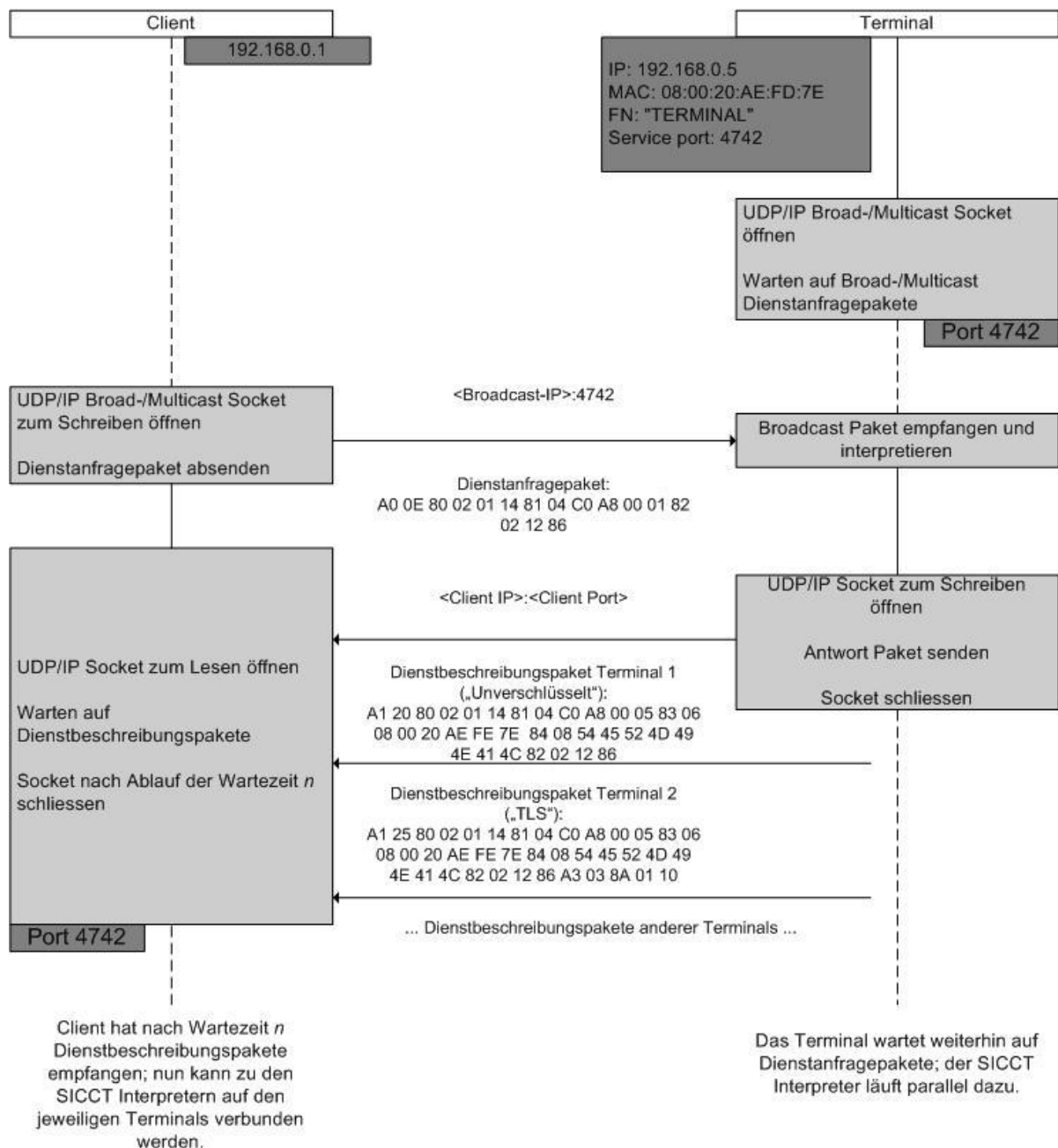


Abbildung 7 Service Discovery Broadcast

6.2.3.3 Konventionen des Dienst anfrage- und Dienstbeschreibungspakets

Die beim Protokoll übertragenen Daten müssen als TLV Objekte (siehe Tabellen) codiert werden.

Der Empfänger einer Nachricht durchsucht ein empfangenes Datenpaket sequentiell nach den TLV Objekten. Das "Protokoll Versions"-Feld muss das erste TLV Objekt in Datenpaket sein, da es so dem Empfänger ermöglicht Annahmen über die Anzahl und Art der folgenden Datenfelder zu treffen. Die Reihenfolge der anderen TLV Objekte im Datenpaket ist nicht relevant. Felder mit nicht aufgelisteten Tag Werten müssen ignoriert werden, um Datenpakete zukünftiger um optionale Felder erweiterte Protokollversionen noch verarbeiten zu können.

Format des SICCT-Terminal MAC Adressen Feldes:

Es ist üblich MAC Adressen in hexadezimaler Schreibweise in der Form anzugeben, dass man mit dem höherwertigen Bits (Bits 47-39) beginnt.

MAC Adresse (6 Byte)					
80:00:20:AE:FD:7E oder 800020AEFD7E					
47	... Bit ...				0
80	00	20	AE	FD	7E

Dementsprechend muss im Discoveryprotokoll das Bit 47 der MAC Adresse auch zuerst übertragen werden.

Format des Protokoll Version Feldes:

Die Versionsnummer der Protokollspezifikation wird als zwei Bytewert definiert, wobei das höherwertige Byte die Hauptversionsnummer, das niederwertige Byte die Nebenversionsnummer angibt. Die Übertragung muss wiederum in network-byte-order erfolgen, d.h. die Hauptversionsnummer muss in Bytestrom zuerst übertragen werden.

Protokollversionen mit gleicher Hauptversionsnummer sind zueinander kompatibel: D.h. sie enthalten einen gemeinsamen Satz an Datenfeldern. Die Nebenversionsnummer gibt an ob und welche optionale Datenfelder hinzugekommen sind.

Protokoll Version 1.30							
Hauptversionsnummer			Nebenversionsnummer				
15	...	Bit ...	8	7	...	Bit ...	0
0000 0001			0001 1110				

Tabelle 29 Protokoll Versionsfeld

Anmerkung: Die network-byte-order entspricht dem Big-Endian Format (das höchstwertige Byte an der niedrigsten Speicheradresse); die Netzwerkimplementierungen der gängigen Betriebssysteme bieten entsprechende Funktionen zur Umwandlung der Daten in die host-byte-order, also der am jeweiligen Prozessor und/oder Betriebssystem gängigen Byte-Reihenfolge.

6.2.4 Kommandotransport und Namensvergabe

6.2.4.1 Adressierung

Ist die IP Adresse und der Port, an dem der SICCT Kommandointerpreter an einem SICCT-Terminal wartet, durch Service Discovery bzw. Announcement oder statische Konfiguration des Clients bekannt, kann dieser die für diese Kommunikation nötige TCP/IP Verbindung zum jeweiligen Terminal öffnen. Steht im Netzwerk DNS zur Verfügung, so können die SICCT-Terminals auch über ihren DNS Namen angesprochen werden. Vor allem bei Benutzerinteraktion sollten DNS Namen den IP Adressen vorgezogen werden, da sie für den Benutzer leichter merkbar sind.

Die Verbindung mit einem SICCT-Terminal kann eine verschlüsselte TCP/IP Verbindung mit ein- oder zweiseitiger Authentifizierung (siehe Kapitel 6.4 "Sicherer Kanal") sein: Signalisiert ein SICCT-Terminal die Unterstützung für verschlüsselte Kommunikation, so müssen die zum Aufbau der verschlüsselten Verbindung nötigen Schritte sofort nach erfolgreichem Aufbau der zugrunde liegenden TCP/IP Verbin-

dung, aber noch vor der SICCT Protokoll basierten Kommunikation durchgeführt werden. Die Aushandlung der für die verschlüsselte Kommunikation erforderlichen Parameter (Algorithmen, Schlüssellängen, etc.) muss über die im jeweiligen sicheren Protokoll vorgesehenen Mechanismen erfolgen.

Der SICCT Kommandointerpreter stellt den Endpunkt der Netzwerkverbindung dar.

Ein SICCT Kommandointerpreter bedient typischerweise genau einen Client zu einer Zeit. Bei Unterstützung konkurrierender Verbindungen muss jede Art von Ressourcenkonflikten vom SICCT-Terminal selbst verwaltet werden.

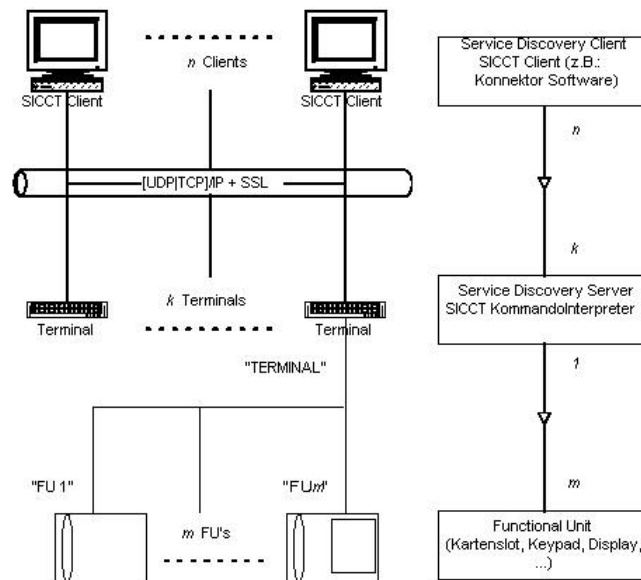


Abbildung 8 SICCT Netzwerkmodell

Ein SICCT-Terminal "exportiert" über den Kommandointerpreter eine oder auch mehrere Functional Units (FU). Folgende Einheiten entsprechen einer FU:

- Terminal
- Schnittstelle für ICC
- Keypad
- Display
- Biometrisches Modul
- Tongeber
- weitere

Das SICCT-Terminal selbst ist eine FU, die immer vorhanden sein muss.

Der Dispatcher-Teil des SICCT Kommando Interpreters am SICCT-Terminal muss die eintreffenden SICCT Kommandos interpretieren: Kommando gelangen daraufhin entweder sofort zur Ausführung oder werden an die im Kommando angegebene FU weitergeleitet.

Die Adressierung und Benennung der FU eines SICCT-Terminals erfolgt über das im Kapitel 5.5.9.2

Ressourcentabelle definierte Schema. Unterschiedliche Anwendungsszenarien erfordern unterschiedliche Benennungen der FU's: Ein Client kann über den GET STATUS Befehl des Command Sets die aktuelle Zuordnung der FU Nummern zum jeweiligen Friendlyname am SICCT-Terminal erfragen.

6.2.4.2 Envelope

Allen SICCT Kommandos (C- und den R-Kommandos, sowie Ereignisnachrichten) muss ein 10-Byte großer Header vorangestellt werden. Im Weiteren als "Envelope" bezeichnet, der folgende Felder enthalten muss:

SICCT Envelope: C- und R- Kommandos/Ereignisnachricht				
Offset	Feld	Datenlänge (Bytes)	Wert (hex)	Beschreibung
0	bMessageType	1	'6B'	C-Kommando
			'83'	R-Kommando
			'50'	Ereignisnachricht
1	wSrcOrDesAddr	2		Falls C-Kommando ('6B'): Zieladresse in network-byte-order
			'0000'	Falls SICCT Kommando APDU
			'0001' – '00FF'	Falls ISO-7816 APDU: ICC 1 – ICC 255
			'1001' – '10FF'	CL-IF 1 – CL-IF 255
				Falls R-Kommando ('83'): Quelladresse in network-byte-order
			'0000'	Falls SICCT Kommando APDU
				Falls ISO-7816 APDU: ICC 1 – ICC 255
			'1001' – '10FF'	CL-IF 1 – CL-IF 255
				Falls Ereignisnachricht ('50'): Quelladresse in network-byte-order
			'0000'	SICCT-Terminal Ereignis
3	wSeq	2		Sequenznummer des Kommandos in network-byte-order
			'0000' - 'FCFF'	SICCT C- und R-Kommando
			'FD00' - 'FFFF'	SICCT Ereignis
5	abRFU	1		Reserved for Future Use
6	dwLength	4	'00000000' - 'FFFFFFFF'	Länge des abCmd Feldes in network-byte-order
10	abCmd	Byte array		Falls C- bzw. R-Kommando: - SICCT Kommando APDU - ISO 7816-3 Karten APDU
				Falls Ereignisnachricht: TLV codierte Ereignisse

Tabelle 30 SICCT Envelope

Nachfolgend werden die Felder des Envelope beschrieben:

- 1) bMessageType:
Das Feld bMessageType gibt die Art der im abCmd Feld enthaltenen Daten an.
- 2) wSrcOrDesAddr:
Das Feld wSrcOrDesAddr spezifiziert die Absender- bzw Empfänger-FU des jeweiligen Kommandos; die Kodierung muss über das in der "Ressourcentabelle" im Kapitel 5.5.9.2 Schema erfolgen. Die Adresse muss in network-byte-order übertragen werden.

- 3) wSeq:
Die Sequenznummer wSeq wird vom Client aus dem in Kapitel 5.5.10.23 beschriebenen Wertebereichschema für jede abgesandte Nachricht vergeben: Der vergebene Wert muss dem in Tabelle 30 SICCT Envelope beschriebenen Wertebereichschema entsprechen. Hierbei sind die unterschiedlichen Wertebereiche für Kommandos und Ereignisse zu beachten. Der Wert muss in network-byte-order übertragen werden. Der Envelope des zugehörigen R-Kommandos des SICCT-Terminals muss wiederum die gleiche Sequenznummer enthalten. So ist es dem Client möglich den C-Kommandos ihre R-Kommandos zuzuordnen.
- Der Client muss selbst dafür sorgen, dass die verwendeten Sequenznummern eindeutig sind: D.h. es muss sichergestellt werden, dass eine Sequenznummer erst dann wieder verwendet wird, wenn auch das zugehörige Antwortkommando mit der gleichen Sequenznummer bereits wieder empfangen wurde. Wurde das Kommando mittels SICCT CONTROL abgebrochen, so wird dessen Sequenznummer mit Empfang der positiven Abarbeitung des CONTROL Kommandos wieder frei.
- Das SICCT-Terminal muss die gleichzeitige Abarbeitung von Kommandos mit gleicher Sequenznummer innerhalb einer Clientverbindung verweigern.
- 4) abRFU:
Das Feld abRFU ist für zukünftige Verwendung frei gehalten.
- 5) dwLength:
Das Feld dwLength muss die Länge der im Feld abCmd gesendeten Daten angeben. Der Längenswert muss in network-byte-order übertragen werden. Die Länge des Envelope ist in dieser Längeninformation nicht enthalten. Diese Längenangabe dient dem korrekten Einlesen von Kommandos über das Netzwerk.

6.2.4.3 Kommandoabarbeitung

Nach dem erfolgreichen Verbindungsaufbau befindet sich das SICCT-Terminal im Ausgangszustand; zu dieser Zeit ist nur das SICCT Kommando INIT CT SESSION zulässig. Mit diesem Kommando kann eine Session geöffnet werden, deren zulässiger Befehlsatz vom authentifizierten Benutzer abhängt: Prinzipiell werden zwei Rollen unterschieden:

- Administrator: darf alle SICCT Kommandos und im Speziellen auch administrative Kommandos, die Einstellungen des SICCT-Terminals manipulieren, ausführen.
- Benutzer: der Benutzer darf entsprechend alle SICCT Kommandos ausführen, bis auf jene, die SICCT-Termineinstellungen ändern. Kommandos die nur der Abfrage von SICCT-Termineinstellungen dienen sind jedoch auch für den Benutzer zulässig.

Sessions sind nicht kaskadierbar, d.h. das in einer Kommandointerpreterverbindung zu einer Zeit immer maximal eine Session aktiv sein darf. Um z.B. von einer User Session in eine Administrator Session zu wechseln, muss zuerst die User Session mittels SICCT CLOSE CT SESSION geschlossen werden, um danach die Administrator Session mittels SICCT INIT CT SESSION zu öffnen.

Das SICCT-Terminal muss in der Lage sein SICCT Kommandos empfangen zu können, obwohl ein früheres SICCT Kommando noch in Abarbeitung ist (Multithreading): Jedes Kommando mit Ausnahme von GET STATUS und SICCT CONTROL darf durch das SICCT-Terminal mit dem Fehlercode `TERMINAL_BUSY` (SW1/2 0x6941) abgelehnt und verworfen werden. Es ist dem SICCT-Terminal aber auch erlaubt, jedes SICCT Kommando entweder parallel zu bearbeiten oder intern zu queueen.

Die Empfänger FU des SICCT-Kommandos muss im Envelope angegeben werden:

- Ein SICCT Kommando APDU muss immer an das SICCT-Terminal selbst gerichtet werden. Zur Kommandoausführung müssen die in der SICCT Kommando APDU spezifizierte(n) FU(s) verwendet werden. Die Codierung der "Empfänger FU's" muss mittels "direct bzw. referenced coding" im P1 Feld des SICCT Kommandos erfolgen. Ob die Kommandoabarbeitung vollständig im SICCT-Terminal oder aufgeteilt zwischen SICCT-Terminal und FU abläuft, ist von der Architektur des SICCT-Terminals und der Art des Kommandos abhängig. Der Typ des SICCT Kommandos bzw. die enthaltenen Daten entscheiden darüber, welche Aktionen an den beteiligten FU's ausgeführt werden müssen.
- ISO 7816 Karten APDU's müssen an der im Envelope angegebene Kontaktiereinheit ausgeführt werden.

Der Client muss nach dem Versand eines SICCT-Kommandos an eine der FU's des SICCT-Terminals auf eine entsprechend codierte Rückantwort warten: Der Envelope des zugehörigen R-Kommandos muss die gleiche Adresse der FU sowie die Sequenznummer wie das zugehörige C-Kommando enthalten.

Der Client kann das GET STATUS Kommando verwenden, um abzufragen, ob das Kommando noch in Abarbeitung ist. Um die Netzwerkbelastung gering zu halten wird empfohlen, dass der Client das GET STATUS Kommando nicht häufiger als einmal innerhalb von 500ms wiederholen darf.

Jede Art von Ressourcenkonflikten muss das SICCT-Terminal selbst verwalten.

Kommandos zum Abfragen und zur Manipulation von Einstellungen / Daten des Terminals selbst müssen immer an die FU "Terminal" ('0000') gesendet werden. Zu diesen Kommandos gehören GET STATUS und CONTROL. Die Befehle GET STATUS und CONTROL dürfen nicht kaskadiert werden: D.h., dass der Zustand eines noch in Ausführung befindlichen GET STATUS Kommandos nicht durch ein weiteres GET STATUS Kommando erfragt werden darf. Genauso darf ein CONTROL Kommando nicht durch ein weiteres CONTROL Kommando abgebrochen werden.

Eine Kommandointerpreter Sitzung (Session) muss typischerweise clientseitig durch das CLOSE SESSION SICCT Kommando beendet werden. Unter bestimmten Umständen, wie nach Konfigurationsänderungen und bei Auftreten bestimmter Fehler wie in Kapitel 6.2.4.6 "Fehlerbehandlung" beschrieben, kann die Sitzung auch terminal-seitig beendet werden. In jedem Fall müssen die gesteckten Karten deaktiviert, und die Initialzustände der FU's und des SICCT-Terminals wiederhergestellt werden.

6.2.4.4 Ereignisbenachrichtigung

Die Netzwerkverbindung, die zum Austausch der Command Set Kommandos dient, wird auch verwendet um den Client über Ereignisse am SICCT-Terminal oder einer seiner FU's zu informieren.

Das SICCT-Terminal kann Ereignisnachrichten nur versenden, wenn eine Session geöffnet ist. Die Ereignisnachrichten dürfen aber nur unidirektional vom SICCT-Terminal zum Client versandt werden. Das SICCT-Terminal erwartet auf versandte Ereignisnachrichten keine Antwort vom Client.

Der Absender einer Ereignisnachricht muss immer die FU "Terminal" ('0000') sein. Die Sequenznummer wSeq muss durch das SICCT-Terminal verwaltet werden, da die Nummerierung der Ereignisnachrichten von den Kommandos unabhängig ist. Zwei aufeinander folgende Ereignisnachrichten müssen unterschiedliche Sequenznummern aufweisen.

ANMERKUNG Diese Nachrichten dienen dem SICCT-Terminal dazu, die verbundenen Clients möglichst rasch über Statusänderungen und Fehlerzustände zu informieren. Unabhängig davon kann ein Client jederzeit Kommandos zur Statusabfrage an das SICCT-Terminal absetzen.

Nachfolgend wird das Datenformat des Ereignisbenachrichtigungspakets beschrieben:

Die über diesen Kanal versandten Ereignisse sind:

- 1) Protokollereignisse;
 - a) KEEP ALIVE: dient der terminalseitigen Detektierung des Netzwerkverbindungsstatus (optional)
 - b) TERMINAL SIGN OFF: Ankündigung eines kontrollierten, terminalseitig initiierten Verbindungsabbruchs.
 - c) PROTOCOL ERROR: Fehler im SICCT Nachrichten Aufbau den Envelope betreffend. Die Fehler sind in der Tabelle Tabelle 31 Ereignisbenachrichtigung mit aufsteigender Priorität angegeben: z.B. bei erfolgreich eingelesenen Envelope wird zuerst der Nachrichtentyp, dann die Verfügbarkeit der angegebenen FU und schließlich die Gültigkeit der Sequenznummer überprüft.
- 2) FU- und Kartenereignisse:
 - a) Hinzukommen und Entfernen von beliebigen FU's
 - b) Slotstatusänderung: Einführen einer Karte in einen Slot bzw. Erkennung einer RFID Karte, sowie deren Entfernung.
- 3) Kommandospezifische Ereignisse:
 - a) Tastaturereignisse

Die Daten der vom SICCT-Terminal versendeten Ereignisnachrichten müssen wie folgt codierte TLV Objekte sein:

Datenfeld	TAG (hex)	Datenlänge (Bytes)	Beschreibung
Protokollereignisse			
KEEP ALIVE	'80'	2	Daten: FU Nummer des SICCT-Terminals Paket zur Detektierung der Funktionstüchtigkeit der Netzwerkverbindung (optional) (siehe 6.2.4.5)
TERMINAL SIGN OFF	'81'	2	Daten: FU Nummer des SICCT-Terminals Ankündigung der terminalseitigen Beendigung der Kommandointerpreterverbindung. Das SICCT-Terminal schließt die Netzwerkverbindung sofort nach Versand der Ereignisnachricht.
PROTOCOL ERROR	'86'	1	Daten: Error Code (hexadezimal)
			'00' Timeout beim Lesen des Envelope
			'01' Timeout beim Lesen der im letzten Envelope angegebenen Anzahl Datenbytes
			'10' Nicht unterstützter SICCT Envelope Nachrichten Typ
			'11' Im Envelope angegebene Zieladresse (FU) nicht vorhanden
			'12' falsche/unerwartete Nachrichtensequenznummer
FU- und Kartenereignisse			
FU Ereignis – FU hinzugekommen	'82'	2	Daten: FU Nummer (entspricht Type und Index)
FU Ereignis – FU entfernt	'83'	2	Daten: FU-Nummer (entspricht Type und Index)
Slotereignis – Karte eingesteckt	'84'	2	Daten: FU-Nummer (entspricht Type und Index)
Slotereignis – Karte entfernt	'85'	2	Daten: FU-Nummer (entspricht Type und Index)
Kommandospezifische Ereignisse			
Tastaturereignis	'87'	3	Daten: FU-Nummer und Tastencode SICCT Kommandokontext: VERIFY, INPUT
			Länge (Bytes)
			2 FU-Nummer der ereignisgenerierenden Tastatur
			1 Tastencode (siehe Tabelle 32 Tastencodes)

Tabelle 31 Ereignisbenachrichtigung

Ein SICCT-Terminal muss alle Protokoll-, sowie FU- und Kartenereignisnachrichten aus "Tabelle 31 Ereignisbenachrichtigung" unterstützen. Die kommandospezifischen Ereignisse sind je nach Unterstützung der jeweiligen Kommandos und Terminalausprägung zu unterstützen. Die Kodierung der FU Nummern erfolgt über das in der Ressourcentabelle im Kapitel 5.5.9.2 definierte Schema.

Ereignisfelder gleichen Typs dürfen hier beliebig oft und in beliebiger Reihenfolge vorkommen.
Ein Ereignisfeld darf immer nur die Daten für ein Ereignis enthalten.

Beispiele (hexadezimal, ohne Envelope):

Ein Slot "Slot1" mit der FU-Nummer 0x0001 hinzugekommen (die Nummernzuordnung ist hersteller-spezifisch und kann mit dem SICCT-Kommando GET STATUS ausgelesen werden):

82 02 00 01

Ein Slot "Slot1" mit der FU-Nummer '0001' entfernt:

83 02 00 01

Ein Slot "Slot1" mit FU-Nummer '0001' hinzugekommen und ein Slot "Slot2" mit FU-Nummer '0005' entfernt:

82 02 00 01 83 02 00 05

oder

83 02 00 05 82 02 00 01

Nachfolgend wird die Handhabung der Ereignisbehandlung (FU- und Kartenereignisse) logischer Slots von CL-IF Ressourcen beschrieben:

Es werden kontaktbehaftete und kontaktlose Slots gleich behandelt. Wird eine Karte in Reichweite eines CL-IF gebracht, so wird diese aktiviert und das Slotereignis „Karte eingesteckt“ mit der entsprechenden FU-Nummer des CL-IF geschickt. Wird die Karte aus der Reichweite entfernt, so muss das Slotereignis „Karte entfernt“ mit der entsprechenden FU-Nummer geschickt werden.

Nachfolgend werden kommandospezifische Ereignisse/Tastaturereignisse beschrieben:

Die Unterstützung von Tastaturereignissen ist für SICCT-Terminals mit Display optional, wird jedoch für SICCT-Terminals ohne Display empfohlen.

Ein SICCT-Terminal mit Tastaturereignis-Unterstützung muss in der Lage mindestens die als nicht optional gekennzeichneten Tastaturereignisse und Tastencodes aus untenstehender Tabelle (Tabelle 32 Tastencodes) senden können.

Tastaturereignisse dürfen nur während der Abarbeitung der SICCT Kommandos VERIFY und INPUT erzeugt werden.

Die Tastaturcodes müssen den im PCSC 2.0 Standard (Part 10 Kapitel 2.10 [PCSC20]) für das SPE Kommando FEATURE_GET_KEY_PRESSED definierten Codes entsprechen: Der Tastencode '00' (Kein Taste gedrückt seit letzter Abfrage) ist im Zusammenhang mit ereignisbasierter Benachrichtigung in SICCT nicht sinnvoll, und darf deshalb nicht in der Liste aufgeführt werden.

Tastencode (hex)	Beschreibung
'2B'	,0' – ,9' Taste gedrückt
'1B'	Abbruch Taste gedrückt
'08'	Korrektur/Lösch Taste gedrückt
'0D'	Eingabe/OK Taste gedrückt
'0A'	Alle Zeichen gelöscht (Backspace) (optional)
'40'	PIN Operation abgebrochen
'0E'	SPE Abbruch durch Zeitüberschreitung

Tabelle 32 Tastencodes

6.2.4.5 Keep-Alive

Mechanismen des TCP Stacks können bei Netzwerkfehler unter Umständen dazu führen, dass lange Wartezeiten entstehen bis ein entsprechender Fehler an höhere Schichten gemeldet werden. Generell ist es schneller möglich solche Fehler beim Versenden als beim Empfangen von Daten zu erkennen.

Um dem SICCT Client bzw. SICCT-Terminal zu ermöglichen solche Fehler der Netzwerkschichten frühest möglich zu erkennen, dürfen beide Kommunikationspartner zu beliebigen Zeiten bestimmte Nachrichten als "Heartbeat" versenden:

- Der Client versendet GET STATUS Nachrichten.
- Das SICCT-Terminal versendet KEEP ALIVE Ereignisse.

Der Status des Versendens eines solchen SICCT Kommandos bzw. Ereignisses gibt Auskunft über die Lebendigkeit der Verbindung.

Der Client sollte frühestens drei Sekunden nach Abschluss der letzten Kommunikation (SICCT Kommando oder Ereignisnachricht) ein GET STATUS Kommando senden. Das Client GET STATUS Kommando sollte nicht öfter als einmal alle 500ms, jedoch mindestens alle 10 Sekunden einmal abgesetzt werden.

Selbige Empfehlungen gelten für das SICCT-Terminal beim Versenden von KEEP ALIVE Ereignissen. Ein größeres Timeout auf SICCT-Terminalseite begünstigt allerdings die Erkennung des Verbindungsstatus durch den clientseitigen Keep-Alive Mechanismus, und entlastet so den Kommunikationskanal. Werden gerade SICCT Kommandos abgearbeitet, und damit auch Daten über das Netzwerk bidirektional versandt, kann aus dem Status dieser Kommunikation auf die Lebendigkeit der Netzwerkverbindung geschlossen werden. Somit darf auf das Versenden von GET STATUS Kommandos und KEEP ALIVE Ereignisses zur Prüfung des Netzwerkstatus zu solchen Zeitpunkten verzichtet werden.

Werden vom SICCT-Terminal KEEP ALIVE Events versandt, so muss das SICCT-Terminal die Netzwerkverbindung selbständig abbauen, wenn es vom Client mindestens 120 Sekunden lang keinerlei Kommandos empfangen hat. Dem Verbindungsabbau muss ein TERMINAL SIGN OFF Ereignis vorangehen.

6.2.4.6 Fehlerbehandlung

Nachfolgend wird die Behandlung von SICCT APDU Fehlern, SICCT Protokollfehlern und Netzwerkfehlern behandelt:

SICCT APDU Fehler:

Fehlercodes der jeweiligen SICCT Kommando APDU müssen in den SW1/2 Feldern des R-Kommando APDU codiert werden.

SICCT Protokollfehler:

Sendet der Client unvollständige Daten, also entweder nur einen Teil des SICCT Envelope oder weniger Daten als im SICCT Envelope Feld dwLength angegeben, so muss der Kommandointerpreter das Einlesen abbrechen, wenn nach Ablauf einer bestimmten Zeit keine der fehlenden Daten gelesen werden konnten (Block Read Timeout) oder der Lesevorgang insgesamt zu lange dauert (Message Timeout).

Mit jedem erfolgreichen Einlesen von zumindest einem Teil der fehlenden Daten muss das Block Read Timeout wieder zurückgesetzt werden.

Die maximale Zeit zum Empfang einer kompletten SICCT Nachricht (Message Timeout) soll Werte von mehreren Minuten annehmen, da ansonsten das Empfangen größerer Datenmengen (z.B. beim Firmwareupload) nicht möglich wäre.

Das Read Timeout sollte einen Wert von 5 Sekunden, sowie das Message Timeout einen Wert von 5 Minuten nicht überschreiten.

Die effektiv benötigte Zeit zum Einlesen einer kompletten SICCT Nachricht ist von der Implementierung des Clients und des Kommandointerpreters, sowie von verschiedenen Netzwerkparametern abhängig. Timeout Werte sollten entweder entsprechend groß vorgewählt oder über die SICCT-Terminalkonfiguration einstellbar gemacht werden.

Läuft eines dieser Timeouts ab, so muss der Kommandointerpreter ein PROTOCOL ERROR Ereignis mit zutreffenden Error Code (siehe "Tabelle 31 Ereignisbenachrichtigung": `00`, `01`) senden. Da hier

ein Wiederaufsetzen nicht möglich ist, muss die Netzwerkverbindung selbständig geschlossen werden. Vor dem Schließen der Verbindung muss noch ein SIGN OFF Ereignis abgesetzt werden.

Ein mit falschen Werten befüllter SICCT Envelope muss mit einem PROTOCOL ERROR Ereignis mit dem zutreffenden Error Code (siehe "Tabelle 31 Ereignisbenachrichtigung": `10`, `11`, `12`) beantwortet werden. Da bei einem solchen Fehler ein Wiederaufsatz möglich ist, muss das SICCT-Terminal einen solchen SICCT Envelope und so viele nachfolgende Bytes, wie es das Feld dwLength dieses Envelope's angibt, verwerfen. Die nachfolgenden Daten müssen wieder als neue SICCT Nachrichten interpretiert werden. Es dürfen maximal 5 derartige Fehler in Folge auftreten. Bei Überschreiten dieser Anzahl sollte das SICCT-Terminal die Netzwerkverbindung nach dem Absetzen eines SIGN OFF Ereignisses selbständig schließen.

Ist einer der hier beschriebenen Empfangsfehler auf eine fehlerhafte Netzwerkverbindung zurückzuführen, die im Empfangsmodus des Kommandointerpreters nicht detektierbar ist, so wird das Absetzen des zugehörigen PROTOCOL ERROR Ereignisses diesen Fehler erkennen und fehlschlagen. Der Kommandointerpreter muss in so einem Fall ebenfalls die Netzwerkverbindung unverzüglich schließen, darf aber auf ein vorheriges Absetzen eines SIGN OFF Ereignisses verzichten.

Protokoll Standardwerte	
Block Read Timeout	5 Sekunden
Message Read Timeout	5 Minuten
Max. Anzahl Protokollfehler	5

Tabelle 33 Protokoll Standardwerte

Netzwerkfehler:

Fehler, die in der Sicherheitsprotokoll Schicht oder darunter liegenden Netzwerkschichten auftreten, müssen primär durch deren Mechanismen behandelt werden: Da ein Fehler in einer der Netzwerkschichten zur Folge hat, dass der SICCT Kommandokanal nicht mehr funktional ist, erübrigt sich die Definition von entsprechenden Fehlernachrichten im Protokoll selbst. Die Fehlerbehandlung muss in den Funktionsschnittstellen der Anwendungsprogramme und der Kommandointerpreter Implementierungen vorgesehen werden. Bei Detektion eines solchen Fehlers muss der Kommandointerpreter die Verbindung schließen und anschließend erneuert auf eingehende Verbindungen warten.

6.2.4.7 Reset und Wiederaufsatz

Auf Änderungen der Einstellungen zur Adressierung und Identifikation des SICCT-Terminals, sowie bei gewissen Fehlerzuständen, muss ein Beenden bestehender SICCT Kommunikationsverbindungen, sowie das Zurücksetzen terminalinterner Stati erfolgen.

Zu diesen Einstellungsänderungen gehören:

- Ändern der IP Adresse
- Ändern der Sicherheitsprotokolleinstellungen

Zu diesen Fehlerzuständen gehören:

- Detektierung des Entferns des Netzkabels am SICCT-Terminal
- Fehler beim Senden eines SICCT Kommandos oder Ereignisses (Nachrichten des Keep Alive Mechanismus miteingeschlossen)

Wurden die Einstellungsänderungen über eine der administrativen Schnittstellen ausgelöst, so muss die Verbindung ordnungsgemäß beendet werden: Das SICCT-Terminal muss ein SIGN OFF Ereignis versenden und es erfolgt ein regulärer Verbindungsabbau entsprechend dem verwendeten Sicherheits-

bzw. Netzwerkprotokolls. Beiden Kommunikationspartnern ist es so möglich die Verbindung "sauber" zu beenden. Nachdem das SICCT-Terminal die Verbindung beendet hat, werden die neuen Einstellungen aktiviert. Danach kann das Auffinden des SICCT-Terminals durch Clients erneut über den Service Discovery bzw. Announcement Mechanismus stattfinden. Um Clients schneller davon zu informieren, dass das SICCT-Terminal wieder verfügbar ist sollte es sich mittels Service Announcement im Netzwerk "melden".

Wurde einer der Fehlerzustände erkannt muss terminalseitig die Beendigung der Verbindung und das Zurücksetzen der internen Stati erfolgen. Der Client und das SICCT-Terminal haben nun keine Möglichkeit Informationen auszutauschen. Es ist beiden erst wieder nach Behebung der Probleme des Netzwerks möglich in Kontakt zu treten:

Wurde z.B. das Netzkabel am SICCT-Terminal selbst entfernt so kann sich dieses, sobald es wieder eine Netzwerkverbindung feststellt, mittels Service Announcement am Netz "melden".

Wurde das Kabel am Client gezogen, so kann dieser bei wiederhergestellter Konnektivität mittels Service Discovery verfügbare SICCT-Terminals finden.

Bei anderen Netzwerkfehlern muss der Client darauf warten, dass seine Service Discovery Nachrichten wieder vom SICCT-Terminal empfangen und entsprechend beantwortet werden können.

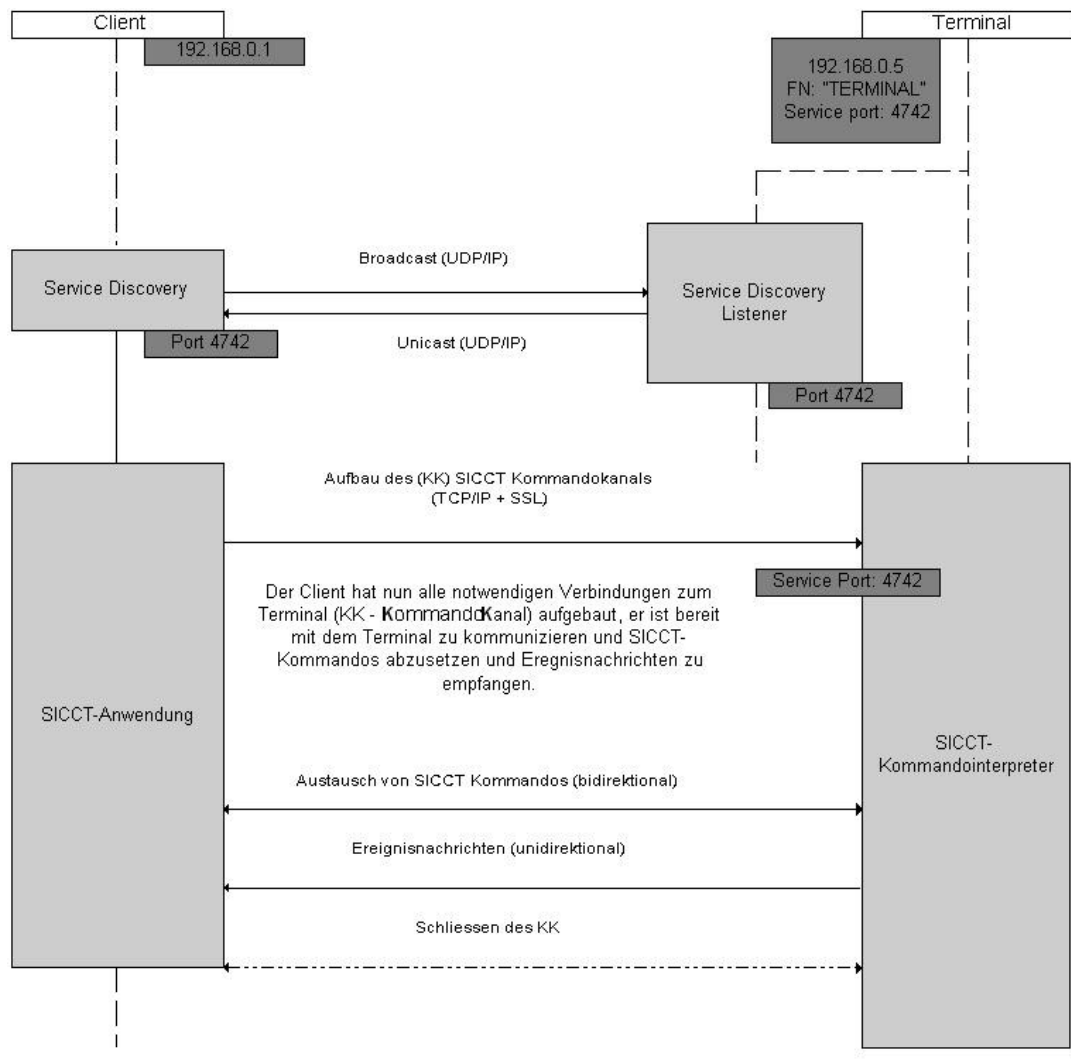


Abbildung 9 SICCT Netzwerkverbindungen

6.2.4.8 Timing

Nachfolgend werden die Anforderungen an das Timing des SICCT-Terminals (Server) und des Clients beschrieben.

- 1) Jede Art von Timingverwaltung für die SICCT Kommandoabarbeitung entfällt auf Terminalseite.
- 2) Der Client muss von einem minimalen Timeout von 3 Sekunden für jedes SICCT Kommando ausgehen. Falls ein längeres Timeout für ein SICCT Kommando notwendig ist, so muss der Client selbst abhängig vom Kommando die Wartezeit erhöhen.
- 3) Der Client kann zu jeder Zeit mit dem GET STATUS Kommando den aktuellen Status der Kommandoabarbeitung am SICCT-Terminal erfragen.
- 4) Der Client kann Kommandos mit dem CONTROL Kommando abbrechen. Das Abbrechen eines Kommandos kann notwendig werden, wenn z.B. ...
 - a) die Bearbeitungszeit des Kommandos länger dauert als der Client dafür vorgesehen hat
 - b) ein Fehler- oder eine Abbruchbedingung auf Clientseite aufgetreten ist

6.2.5 Auslieferungszustand

Nachfolgend werden die Parameter beschrieben, die ein SICCT-Terminal im Auslieferungszustand erfüllen muss. Der Auslieferungszustand eines SICCT-Terminals muss umfassen:

- 1) Standardwerte für die Netzwerkkonfiguration
 - a) DHCP ist aktiv

- b) IP Adresse und Netzmaske undefiniert oder mit Werten aus einem nicht route-baren Adressbereich [RFC1918] vorbelegt.
- c) SICCT-Terminal Name (DNS)
- 2) Standardwerte für das Service Discovery:
 - a) Dienstanfrageport:
 - i) Port: 4742 (UDP, Broadcast)
 - b) Dienstankündigungsport (Service Announcement Port):
 - i) Port: 4742 (UDP, Broadcast)
- 3) Standardwerte für den SICCT Kommandointerpreter:
 - a) SICCT Kommandointerpreter Port: 4742 (TCP)
 - b) Hostname des SICCT-Terminals (DNS)
 - c) Namen für die FU's entsprechend einer hersteller- oder fachspezifischen Sicht
- 4) SICCT-Terminal Zertifikat / Schlüssel

6.3 Terminal Managementverfahren

6.3.1 Allgemeine Anforderungen an den Ablauf des Terminalmanagement

Die Ausführung administrativer Tätigkeiten über andere Schnittstellen als die Kommandointerpreterverbindung dürfen den laufenden Betrieb des SICCT-Terminals so lange wie möglich nicht stören oder beeinflussen. Die Möglichkeit, parallel SICCT Kommandos abzuarbeiten und zeitgleich administrative Tätigkeiten auszuführen, hängt von der jeweils verwendeten Administrationsschnittstelle und den Fähigkeiten des Geräts ab. Werden bei einer solchen parallelen Abarbeitung gleiche Ressourcen des SICCT-Terminals beansprucht, so muss die ausführende Logik im SICCT-Terminal dafür sorgen, diese Ressourcen exklusiv bis zur Beendigung der Aktionen zu erhalten. Eine konkurrierend zugreifende Logik muss warten bis der Vorgänger die Ressource wieder freigibt. Wird in einem SICCT Kommando eine bereits verwendete Ressource angefordert, so muss das Kommando mit dem Statuswort (SW1/SW2) `6941` "Funktionseinheit belegt/nicht verfügbar" beantwortet werden.

SICCT-Terminals mit Anschlussmöglichkeit "Ethernet":

Zur Information eines verbundenen Clients über einen terminalseitig initiierten Abbau der Kommandointerpreterverbindung, muss das TERMINAL SIGN OFF Ereignis (siehe 6.2.4.4) gesendet werden. Ein solches Ereignis ist beispielsweise vor dem effektiven Ändern von SICCT-Terminaleinstellungen, welche den Zustand der Kommandointerpreterverbindung beeinflussen, nötig.

6.3.2 Managementschnittstellen

Die administrativen Schnittstellen eines SICCT-Terminals müssen umfassen:

1. SICCT-Kommandointerpreterverbindung (Administrative SICCT Kommandos)
2. Web-basiertes Management Interface. ("Ethernet")
3. Weitere administrative Schnittstellen dürfen angeboten werden, sind aber herstellerspezifisch.

Für alle Varianten müssen entsprechende Methoden zur sicheren Übertragung der Daten benutzt werden (siehe 6.4 "Sicherer Kanal").

Nachfolgend werden die in den jeweiligen Managementschnittstellen zulässigen Aktionen aufgelistet:

Managementaktionen	Modus	(1)	(2)	(3)
Anzeigen der aktuellen Netzwerkkonfiguration	U/A	-	opt.	opt.
Ändern des Display Standard Texts	A	opt.	opt.	opt.
Ändern der Netzwerkkonfiguration				
Verwendung von DHCP oder statischer Konfiguration	A	-	opt.	opt.
Standard IP Adresse, Netzwerkmaske, Gateway	A	-	opt.	opt.
Nameserver Adresse(n)	A	-	opt.	opt.
DNS Name des SICCT-Terminals	A	-	opt.	opt.
Dienstanfrage und Service Announcement:				
Ändern der Dienstanfrageports	A	-	opt.	opt.
Ändern der Dienstankündigungsports	A	-	opt.	opt.
Zertifikats- und Schlüsselmanagement				opt.
Zertifikate/Schlüssel Informationen anzeigen	U/A	-	opt.	opt.
Zertifikate/Schlüssel austauschen/erzeugen/löschen	A	-	opt.	opt.
SICCT Kommandointerpreter Konfiguration				
Ändern des Ports des Kommandointerpreters	A	-	opt.	opt.
Vergabe logischer Namen der FU's des SICCT-Terminals	A	mand.	mand.	opt.
Firmware Update	A	mand.	opt.	opt.
Zurücksetzen der Konfiguration in den Werkszustand	A	-	opt.	opt.
"-" ... nicht zulässig "mand." ... mandatory "opt." ... optional	Modi: U ... User Modus A ... Admin Modus			

Tabelle 34 Managementaktionen

Die Managementschnittstellen müssen sicherstellen, dass Abfragen und Änderungen der Konfiguration erst nach erfolgreicher Authentifizierung erfolgen können: Der Benutzer ist nur berechtigt, seine aktuellen Einstellungen einzusehen und sein eigenes Kennwort zu ändern. Das Einsehen der aktuellen Konfiguration muss geschützt werden, da die Daten der SICCT-Terminalkonfiguration Informationen enthalten können, die für DoS und ähnliche Attacks benutzt werden können. Dem Administrator muss es ermöglicht sein, Einstellungen zur Benutzerverwaltung, Netzwerkkonfiguration (bei Anschlussmöglichkeit "Ethernet"), den SICCT-Terminal und Slotnamen sowie Zertifikate zu ändern.

6.3.2.1 Administrative SICCT Kommandos

Parameter aus der Liste der Managementaktionen dürfen auch über Kommandos des SICCT Protokolls gelesen und verändert werden. So kann eine client-seitige Applikation Befehle aus dem Command Set benutzen, um SICCT-Terminals zu konfigurieren.

Für die Veränderung dieser Daten muss, so wie bei anderen Managementschnittstellen auch, eine vorangehende erfolgreiche Authentifizierung erfolgt sein: Die Authentifizierung erfolgt mittels des INIT SESSION SICCT Kommandos. Nach dem Verbindungsaufbau ist noch keine Session aktiv. Das INIT SESSION Kommando muss bei erfolgreicher Authentisierung in den mit dem Benutzername assoziierten Modus wechseln. Konnte eine Administrator Session erfolgreich geöffnet werden, können mit den entsprechenden Befehlen (wie SICCT SET STATUS, SICCT DOWLOAD [INIT] DATA[FINISH]) Änderungen durchgeführt werden. Wurde nur eine User Session geöffnet, dürfen die Befehle für die administrativen Aktionen nicht zu Verfügung stehen, aber muss die Möglichkeit der Statusabfrage sowie müssen die weiteren unprivilegierten SICCT Befehle bereitstehen. Das SICCT Kommando CLOSE SESSION schließt die Session wieder und das SICCT-Terminal wechselt in den Ausgangszustand zurück. Sessions dürfen nicht kaskadiert werden. Innerhalb einer Kommandointerpreterverbindung darf immer maximal eine Session zu einer Zeit aktiv sein.

6.3.2.2 Web-basierte Managementschnittstelle ("Ethernet")

Die Organisation und Präsentation der verschiedenen Konfigurationsdaten eines SICCT-Terminals ist herstellerspezifisch und orientiert sich an den technischen Möglichkeiten und Kapazitäten der Terminal-hard- und software (Webserver). Die Webmanagementschnittstelle ist nicht für eine tool-unterstützte automatisierte Geräteverwaltung vorgesehen; dafür sind geeignete SICCT-Terminal-Kommandos im Command-Set (siehe Kapitel 5 "Command Set") definiert. Grundlegende Richtlinien an eine solche Benutzerschnittstelle sind in Kapitel 7 "Schnittstelle zum Benutzer" zu definieren.

6.3.2.3 Herstellerspezifische Managementschnittstellen

Die Beschreibung der Ausprägung herstellerspezifischer Managementschnittstellen ist nicht Teil dieses Standards. Solche Schnittstellen können beispielsweise mit Hilfe der am SICCT-Terminal vorhandenen Ein-/Ausgabemöglichkeiten realisiert werden, oder auch andere dem Gerät entsprechende Schnittstellen zur Benutzerinteraktion nutzen.

Bei Verwendung von Einheiten des SICCT-Terminals, die auch über die SICCT Kommandos der Kommandointerpreterverbindung gesteuert werden können (wie Keyboard und Display), müssen die Anforderungen bzgl. der Handhabung von Ressourcenkonflikten aus Kapitel 6.3.1 beachtet werden.

6.3.3 Firmware Download

Das SICCT-Terminal sollte die Möglichkeit bieten, den SICCT-Kommandointerpreter, sowie andere "Systemkomponenten" zu aktualisieren. Die Form der dafür benötigten Daten hängt von der konkreten Implementierung des SICCT-Terminals ab, und kann somit einer Firmware für einen bestimmten Mikroprozessor, als auch einer PC-ähnlichen Softwarearchitektur entsprechen. Je nach Ausbau des SICCT-Terminals kann die Aktualisierung der Software einen vollständigen oder auch einen komponentenweisen Austausch bedeuten. Das Format, in dem die Daten bereitgestellt werden, ist herstellerspezifisch. Bei der Übertragung selbst müssen die Daten und eine angehängte Signatur in das jeweils verwendete Protokoll eingebettet werden.

6.3.3.1 Vorgeschriebener Download Mechanismus

Jedes SICCT-Terminal muss die Kommandos des Command-Sets zum Firmware Download unterstützen.

Optionale Download Mechanismen dürfen zusätzlich vom Hersteller implementiert werden um die Firmware z.B. über Ethernet per http, über RS-232 oder USB zu transportieren.

6.3.3.2 Integrität und Authentizität der Daten

Die Dateien, die ein Firmware Download ausmachen, müssen vor der Programmierung ins Gerät auf Integrität und Authentizität überprüft werden, um falsche oder böswillig veränderte Daten vorzeitig zu erkennen: Die zur Sicherstellung der Integrität und Authentizität verwendeten Verfahren sind herstellerspezifisch.

6.4 Sicherer Kanal

6.4.1 Sicherheitsprotokolle

Das SICCT-Terminal kann über das Feld "Sicherheitsprotokolle" im Dienstbeschreibungspaket dem Client die Unterstützung von Sicherheitsprotokollen anzeigen.

Das SICCT-Terminal erwartet, das beim Verbindungsaufbau versucht wird, das Protokoll mit der höchsten Priorität zuerst zu benutzen. Schlägt eine entsprechende Einrichtung der Verbindung fehl, so muss das Protokoll mit der nächstniedrigeren Priorität versucht werden. Diese Schritte werden so lange wiederholt, bis entweder ein Protokoll erfolgreich eingerichtet werden konnte, oder keine weiteren Protokolle mehr zur Verfügung stehen. Konnte kein Protokoll eingerichtet werden so muss die zu Grunde liegende TCP/IP Verbindung geschlossen werden.

6.4.1.1 TLS

Für die Verschlüsselung der Kommunikation mittels TLS (Transport Layer Security) muss mindestens die Protokollversion 1.2 [RFC5246] angeboten werden. TLS1.3 [RFC8446] kann optional unterstützt werden. Die älteren TLS Cipher Suites TLS 1.0 und TLS1.1 dürfen nicht mehr unterstützt werden.

Wird TLS implementiert und gibt es keine fachspezifischen Vorgaben bzgl. der zu unterstützenden Cipher Suites, so muss zumindest die Cipher Suite TLS_RSA_WITH_AES_128_CBC_SHA implementiert sein, um die Mindestanforderungen des TLS 1.2 Standards zu erfüllen (siehe [RFC5246#9, Mandatory Cipher Suites]).

Die Aushandlung der konkret zu verwendenden Cipher Suite und ihrer Parameter erfolgt im Rahmen des im TLS Standard definierten Handshakeverfahrens.

Beim Einsatz von TLS muss eine einseitige Authentifizierung des SICCT-Terminals gegenüber dem Client verpflichtend unterstützt werden. Das SICCT-Terminal darf optional auch eine Authentifizierung des Clients vornehmen (zweiseitige Authentifizierung). Dementsprechend soll es dem SICCT-Terminal möglich sein, die Gültigkeit des Client-Zertifikats zu prüfen.

Beim Einsatz von TLS muss die Installation passender Zertifikate am SICCT-Terminal und Client möglich sein.

6.4.2 Zertifikate

Es muss möglich sein, ein nach dem X.509v3 Standard [RFC3280] ausgestelltes Zertifikat auf dem SICCT-Terminal abzulegen. Das verwendete Dateiformat zum Aufbringen bzw. zur Speicherung des Zertifikates am Terminal ist dem Terminalhersteller überlassen.

Die Standardfelder spezifizieren den Terminalhersteller näher: Herstellername, Supportadresse, Gerätetypbezeichnung.

Das Zertifikat muss einen der folgenden Algorithmen verwenden:

- 1) RSASSA-PSS mit einer Moduluslänge von mindestens 3072 bit verwenden, oder
- 2) ECDSA mit der elliptischen Kurve ansix9p256r1, oder
- 3) ECDSA mit der elliptischen Kurve brainpoolP256r1.

Für die Webmanagementoberfläche und die Verbindung des Command Sets muss ein gemeinsames Zertifikat verwendet werden. Dementsprechend muss das Zertifikat für Webserverauthentifizierung (x.509v3 Standard Extensions Key Usage) verwendbar sein.

Ein Terminalzertifikat darf nicht zur Signierung anderer Zertifikate verwendet werden (Standard Extensions Basic Constraints).

Optional darf es möglich sein, das Zertifikat am SICCT-Terminal auszutauschen, um die jeweilige Vertrauensstellung an die Einsatzumgebung des SICCT-Terminals anzupassen. Das Verändern bzw. das Tauschen des Zertifikats darf nur durch den "Administrator" (siehe Benutzerrollen in Terminal Managementverfahren) erfolgen: Dementsprechend muss eine erfolgreiche Authentifizierung an der Webmanagementoberfläche bzw. im Command Set erfolgen, um die entsprechenden Aktionen zu erlauben.

SICCT konformes x.509 Zertifikat	
Feld	Beschreibung
Version	Zertifikatsstandardversion: 3
Seriennummer	Seriennummer des Zertifikats
Signatur Algorithmus	Signaturalgorithmus der ausstellenden Instanz (CA)
Aussteller	Terminal Root CA Distinguished name (DN)
Gültigkeit	Gültigkeit (Start und Enddatum)
Subjekt	Terminalherstellereinstellungen (DN der mit dem Zertifikat assoziierten Instanz)
Subject Public Key Info	Subject Public Key Algorithmus und Subject Public Key
Erweiterungen	KeyUsage, BasicConstraints und X509v3 Subject Alternative Name
Zertifikat Signaturalgorithmus	Signatur der ausstellenden Instanz (CA).
Zertifikat Signatur	

Tabelle 35 SICCT konformes x.509 Zertifikat

6.4.3 Sicherung der netzwerkbasierten Verbindungen

6.4.3.1 Sicherung des Service Discovery/Announcement

Die Kommandos zum Auffinden der SICCT-Terminals (Service Discovery/Announcement) werden über unverschlüsselte Verbindungen gesendet.

6.4.3.2 Sicherung des SICCT Kommunikationskanals

Wird der SICCT Kommunikationskanal mit einem der zur Verfügung stehenden Sicherheitsprotokolle abgesichert, so darf erst mittels SICCT Kommandos kommuniziert werden, nachdem das Sicherheitsprotokoll erfolgreich eingerichtet wurde.

6.4.3.3 Sicherung der Managementschnittstellen

Nachfolgend werden die Mechanismen zur Sicherung der Managementschnittstellen beschrieben.

SICCT Kommandos:

Administrative SICCT Kommandos sollten nur bei verschlüsseltem SICCT Kommunikationskanal angewandt werden. Ansonsten gelten die Vorgaben aus Kapitel 6.4 Sicherer Kanal.

Webmanagementschnittstelle:

Für den Zugriff auf das SICCT-Terminal mittels Webmanagementschnittstelle darf TLS verwendet werden. Der Aufbau der TLS Verbindung erfolgt nach der im HTTPS Protokoll definierten Vorgehensweise. Für Einsatzszenarien, die eine verschlüsselte SICCT Verbindung voraussetzen, sollte der Zugriff auf die Webmanagementschnittstelle nur verschlüsselt zugelassen werden.

Herstellerspezifische Managementschnittstellen:

Die Sicherung der Daten darf in einer der Art der Schnittstelle angepassten Weise erfolgen.

7 Schnittstellenbeschreibung zum Benutzer

7.1 Tastatur

Der Chipkartenleser kann eine Tastatur besitzen. Falls eine Eingabetastatur vorhanden ist, müssen folgende Regelungen beachtet werden:

Es sollten die Tasten "0" bis "9", eine Abbruch und eine Bestätigungstaste vorhanden sein. Eine Korrekturtaste wird optional empfohlen.

Es können zusätzliche Tasten vorhanden sein, z.B. zusätzliche Funktionstasten.

Auf eine ergonomisch günstige Ausprägung der Tastatur sollte geachtet werden.

Die Anordnung der Tasten muss in Anlehnung an CEN EN 1332-3 [STD20] "Identifikationskartensysteme – Schnittstelle Mensch Maschine – Teil 3: Tastenfelder" erfolgen.

Bei Tastaturen mit eingebautem Chipkartenleser obliegt die Anordnung der Tasten wegen der Vielzahl unterschiedlichen Anforderungen dem Tastaturhersteller.

7.2 Display/Anzeige

Falls ein Display vorgesehen ist, muss die Größe von mindestens 2 Zeilen mit je minimal 16 Zeichen zur Darstellung bereitgestellt werden. Bei einem virtuellem SICCT-Terminal (z.B. ein PC mit einem Chipkartenleser in der Tastatur) muss der Software-Proxi für eine Emulation des Displays auf dem Monitor sorgen. Außer der deutschen Sprachanzeige können weitere Sprachen zur Anzeige von Meldetexten implementiert werden. Weitere Symbole und Zeichen zur Benutzerführung (z. B. Sicherheitsmodus) können verwendet werden. Bei Anzeigetexten mit nachfolgender Tastatur-Eingabe muss ein blinkendes Cursor-Zeichen die Position des Cursors anzeigen. Für den oben beschriebenen Sicherheitsmodus müssen Standardtexte im Chipkartenleser vorgehalten werden.

Folgende Standardtexte werden festgelegt:

Nr.	Text		Supported in Mode	
	German	English	BCS	SICCT
1	Bitte Karte einführen	Please insert card	✓	✓
2	Bitte Karte entnehmen	Please remove card	✓	✓
3 ¹³	Karte unlesbar. Falsche Lage?	Card illegible. Wrong position?	✓	RFU
4	Bitte Geheimzahl eingeben	Please enter PIN	✓	✓
5	Aktion erfolgreich	Action successful	✓	✓
6	Geheimzahl falsch/gesperrt	PIN wrong or blocked	✓	✓
7	Neue Geheimzahl eingeben	Please enter new PIN	✓	✓
8	Eingabe wiederholen	Repeat input	✓	✓
9	Geheimzahl nicht gleich. Abbruch	PIN not identical. Abort	✓	✓
10	Bitte Eingabe bestätigen	Please confirm input	✓	✓
11	Bitte Dateneingabe	Please enter data	✓	✓
12	Abbruch	Abort	✓	✓

Tabelle 36 Standard-Anzeigetexte für SICCT- und BCS Betriebsmode.

CT Idle Message

Zusätzlich zu den dargestellten Standardtexten verfügt ein SICCT-Terminal über eine sog. CT Idle-Message, welche herstellerspezifisch ausfallen und über die Mechanismen der Geräteadministration angepasst werden kann. Die Darstellung der Idle-Message muss nach dem Kommando, die ein Rücksetzen des SICCT-Terminals bedeuten, oder per Option des SICCT OUTPUT Befehls erfolgen.

7.2.1 Transparente Displayansteuerung

Durch die Kommandos OUTPUT (BCS Mode) und SICCT OUTPUT (SICCT Mode) kann der Host Ausgaben auf dem Display vornehmen.

Die Kommandodaten müssen ein TLV Objekt beinhalten.

Kommandos und Daten Objekte sind im Kapitel Command Set definiert.

¹³ Standard-Anzeigetext Nr. 3 entfällt im SICCT Betriebsmode.

7.2.1.1 Transparente Displayansteuerung "BCS mode"

Bei Verwendung des Kommandos OUTPUT (BCS kompatibel) wird davon ausgegangen, dass ein zweizeiliges Display mit je 16 Zeichen zur Verfügung steht. Als Zeichenvorrat müssen Groß- und Kleinbuchstaben inklusive Umlaute sowie die Sonderzeichen entsprechend DIN 66003, insbesondere inklusive "Stern", unterstützt werden. Als Steuerzeichen darf in einem Anzeigetext CR (0Dh, Wechsel in die zweite Zeile) verwendet werden.

Falls eine Zeile mit weniger als 16 Zeichen beschrieben wird, müssen die restlichen Zeichen der Zeile gelöscht werden.

Als Kommandodaten darf nur das Application Label Data Object verwendet werden.

7.2.1.2 Transparente Displayansteuerung "SICCT mode"

Bei Verwendung des Kommandos SICCT OUTPUT (SICCT kompatibel) kann vorher mit SICCT GET / SET STATUS abgefragt werden, welche Displayart vorhanden ist und welcher Zeichensatz unterstützt wird.

Als Objekt für SICCT Output kann das "SICCT Message To Be Displayed DO" verwendet werden. Dieses Objekt besteht aus dem "Character Set Data Object:" und dem "Application Label Data Object"

Bei Character Set Data Object Value '00' (DIN 66003) wird davon ausgegangen, dass ein zweizeiliges Display mit je 16 Zeichen zur Verfügung steht. Als Zeichenvorrat müssen Groß- und Kleinbuchstaben inklusive Umlaute sowie die Sonderzeichen entsprechend DIN 66003, insbesondere inklusive "Stern", unterstützt werden. Als Steuerzeichen darf in einem Anzeigetext CR (0Dh, Wechsel in die zweite Zeile) und BEL (07h, Ausgabe eines Beep) enthalten sein.

Falls eine Zeile mit weniger als 16 Zeichen beschrieben wird, müssen die restlichen Zeichen der Zeile gelöscht werden.

Wenn mehr druckbare Zeichen ausgegeben werden sollen, als das Display in einer Zeile darstellen kann, dürfen überzählige Zeichen nicht bzw. erst durch eine Scrollfunktion dargestellt werden.

7.2.2 Leuchtdioden

Optional können Informationen über den Betriebszustand des Chipkartenleser durch Leuchtdioden signalisiert werden.

Eine Leuchtdiode in einer ersten Farbe (vorzugsweise grün) kann den Zustand nach einer korrekten Initialisierung des Chipkartenlesers nach Anlegen der Versorgungsspannung signalisieren. Der Betriebszustand nach Aktivierung (kontaktbehaftet) oder Selektierung der (kontaktlosen) Chipkarte kann durch eine Leuchtdiode in einer zweiten Farbe (vorzugsweise gelb) angezeigt werden. Eine blinkende Leuchtdiode in der zweiten Farbe oder einer dritten Farbe kann den Fehlerfall signalisieren.

Falls nur eine Leuchtdioden-Anzeige vorhanden ist, muss diese anzeigen, wenn eine Chipkarte aktiviert oder selektiert ist. Die Bereitschaft des Lesers sollte ebenfalls dem Bediener angezeigt werden.

Sollten die Betriebszustände nicht über Leuchtdioden ersichtlich sein, so dürfen alternativ geeignete Ausgaben z.B. über ein Display den jeweiligen Betriebszustand anzeigen.

7.3 Tongeber

Der Chipkartenleser kann mit einem Tongeber ausgestattet sein. Dieser kann z.B. einen Fehler signalisieren.

7.4 Biometrische Einheiten

Der Chipkartenleser kann zusätzlich einen oder auch mehrere biometrische Sensoren besitzen. Möglichkeiten sind beispielsweise Fingerabdruck, Unterschrifts- und Spracherkennung oder Irisabtastung zur Identifikation biometrischer Merkmale. Die biometrischen Daten dürfen dazu nicht in die unsichere Umgebung des Hosts gelangen.

7.5 Benutzerauthentisierung bei der Komfortsignatur

Eine Authentisierung eines Signaturkarteninhabers muss über die Verifikation einer PIN erfolgen.

Zur Optimierung von Bedienabläufen kann die Komfortsignatur verwendet werden.

Dabei kann die Signaturkarte nach erfolgreicher Verifikation der PIN für eine begrenzte Anzahl von Signaturen eingesetzt werden. Vor jeder Signatur kann dann eine vereinfachte Authentisierung z.B. mittels biometrischer Einheiten (siehe Kapitel 7.4) oder sicheren Objekten (siehe Kapitel 7.5) erfolgen.

Die Sicherheitsanforderungen für die Komfortsignatur muss für jede Anwendung individuell bewertet und spezifiziert werden.

7.5.1 Anlernfunktion

Das SICCT-Terminal kann über eine Anlernfunktion für die vereinfachte Authentisierung verfügen.

Dabei wird z.B. die Kartennummer aus der Signaturkarte ausgelesen und zusammen mit einem biometrischen Merkmal des Signaturkarteninhabers oder einer eindeutigen Information aus dem sicheren Objekt im SICCT-Terminal gespeichert.

Wenn dann eine Komfortsignatur durchgeführt werden soll, erhält das SICCT-Kartenterminal vom Host das Kommando SICCT COMFORT AUTHENTICATION mit der Kartennummer der Signaturkarte. Das SICCT-Kartenterminal muss dann das biometrische Merkmal bzw. die Anwesenheit des sicheren Objekts verifizieren.

7.6 Sicherer Modus

Sicherheitstechnische Applikationen erfordern authentische Ein- und Ausgaben. So muss beispielsweise dem Benutzer signalisiert werden, dass derzeit die über die Tastatur des Chipkartenlesers eingegebene Geheimzahl nicht in die unsichere Umgebung des Personalcomputers oder über offene Übertragungswege an den Host gelangt. Auch Ausgaben die zum Beispiel bei einer digitalen Signatur oder einem Bezahlvorgang auftreten, können eine authentische Anzeige voraussetzen. Um zu informieren, dass der Chipkartenleser sich im gesicherten Modus befindet, muss er dieses dem Anwender eindeutig signalisieren.

Dazu müssen akustische, optische und/oder andere deutlich wahrzunehmenden Signale zur Verfügung gestellt werden. Zusatzleuchten und Symbole in Displays sind derzeit gängige Anzeigen. Es sollte beachtet werden, dass eine behindertengerechte Unterstützung der Funktion, z. B. durch vergrößerte Symbole oder eine Kombination von optischen und akustischen Anzeigen, gewährleistet ist. Dabei muss sichergestellt werden, dass das Signal nicht unbefugt ansteuerbar sein darf, und nur von der Firmware des Chipkartenlesers bedient werden kann.

Die Nutzung der Signalisierung muss dem Benutzer in der Dokumentation eindeutig dargestellt werden.

Eine weitere Möglichkeit, dem Benutzer zu signalisieren, dass sich der Chipkartenleser im gesicherten Modus befindet, ist die Anzeige einer "Display Message". Diese Display Message ist eine in der Chipkarte gespeicherte Nachricht, die vom Karteninhaber geändert werden kann. Z.B. nach der erfolgreichen Etablierung eines Trusted Channels zur Chipkarte kann die Display Message aus der Chipkarte gelesen und auf dem Chipkartenleser angezeigt werden.

Der sichere Modus muss für den Benutzer einwandfrei ersichtlich sein, er darf nicht vorgetäuscht werden können.

8 Sicherheit

8.1 Allgemeine Betrachtungen

Nachfolgend werden allgemeine Anforderungen beschrieben, die ein SICCT-Terminal erfüllen muss.

- SICCT_R1:** Die wichtigste Anforderung ist der fehlerfreie und unversehrte Betrieb der Chipkarte(n). Das SICCT-Kartenterminal muss die Chipkarte(n) elektrisch und mechanisch schützen und einen stabilen Betrieb gewährleisten.
- SICCT_R2:** Die vorrangige Anforderung ist die Bereitstellung einer normgerechten Funktionalität an allen elektrischen Anschlüssen (Interfaces) zur Außenwelt. Dies bezieht die Chipkarten-Interfaces (Kontaktiereinheiten) sowie Anschlüsse und Kommunikationsports (z. B. RS-232, USB, Ethernet 802.3) zu anderen IT-Systemen mit ein. Es gilt die ISO 7816.
- SICCT_R3:** Anmerkung: Das SICCT-Kartenterminal stellt eine sichere Komponente für die Verwendung von Chipkarten bereit und signalisiert dem Kartenanwender einen erkennbar sicheren und vertrauenswürdigen Betriebszustand.
- SICCT_R4:** Das SICCT-Kartenterminal muss technische Funktionen zur Anwenderauthentifizierung (zum Beispiel Keypad, biometrische Sensorik) anbieten. Authentisierungsdaten dürfen vom SICCT-Kartenterminal nicht an das Host-System weitergegeben, intern nicht gespeichert und fallbezogen angewendet werden.
- SICCT_R5:** Das SICCT-Kartenterminal muss für Dauerbetrieb geeignet sein, sowie einem Betreiber / Nutzer eine hohe Verfügbarkeit (z. B. geringe Ausfallquote, zugesicherte mean-time-between-failure) bieten.
- SICCT_R6:** Das SICCT-Kartenterminal muss hinsichtlich der genannten Anforderungen und entsprechend dem Einsatzumfeld und der Einsatzumgebung durch akkreditierte Prüf- oder Zulassungsstellen validiert werden.
- SICCT_R7:** Die Geräte-Firmware kann über die Lebensdauer des SICCT-Kartenterminals aktualisiert werden, um
- die Einführung neuer Funktionen
 - die Beseitigung von erkannten Sicherheitsrisiken
 - eine generelle Fehlerbehebung vorzunehmen.
- Voraussetzung hierfür ist, dass das SICCT-Terminal mit der neuen Firmware bereits validiert und zugelassen ist.
- SICCT_R8:** Das SICCT-Kartenterminal muss sich in Verbindung mit entsprechender Software als Teil einer Signaturanwendungskomponente (HW und SW) entsprechend den Anforderungen aus SigG und SigV zur Erstellung von qualifizierten elektronischen Signaturen eignen.
- SICCT_R9:** Das SICCT-Kartenterminal muss auf bestehenden Industriestandards basieren, die additiv für den Betrieb in konkreten Anwendungen ggf. eingeschränkt oder erweitert werden müssen. Die Standardisierungsanforderung orientiert sich an:
- elektrischen Eigenschaften
 - mechanischen Eigenschaften
 - bestehenden Sicherheitsanforderungen im Gesundheitswesen und avisierten Chipkartenanwendungen (qualifizierte Signatur).
- SICCT_R10:** Das SICCT-Kartenterminal soll konform zu einer gut verständlichen technischen Spezifikation gehalten werden, welche u.a. die Sicherheitsanforderungen beschreibt.

8.2 Sicherheitsziele

Nachfolgend werden Sicherheitsziele beschrieben, die ein SICCT-Terminal erfüllen muss.

SICCT_O1: Schutz des Datenverkehrs zwischen SICCT-Terminal und Host

Anhand von Analysen der Anwendungen und Anwendungsumgebungen ist festzulegen, ob und mit welchen Methoden der Datenverkehr zwischen SICCT-Terminal und Host bezüglich Ausspähen und Manipulation zu schützen ist.

Hohe sicherheitstechnische Anforderungen ergeben sich bei Einsatzszenarien in nicht kontrollierten Umgebungen und vor allem beim Transfer von sensiblen Daten (qualifizierte Signatur, personenbezogene medizinische und Sozialdaten).

SICCT_O2: Schutz der Authentifikationsdaten (PIN)

Mittels der sicheren PIN Eingabe kann das SICCT-Terminal Identifikationsdaten (PIN) erfassen und sie auf sicherem Weg an sichere Signaturerstellungseinheiten (Signatur-Chipkarten) gemäß SigG §2 Nummer 10 übermitteln.

SICCT_O3: Schutz der Authentizität und Integrität der Firmware

Wenn das SICCT-Terminal die Funktionalität eines Firmware-Updates anbietet, dann muss mit geeigneten Methoden die Authentizität und Integrität der neuen Firmware gewährleistet werden. Hierzu sind Verfahren gemäß der aktuellen Richtlinien des BSI zu verwenden.

SICCT_O3: Schutz vor Vortäuschen einer falschen Identität des Kommunikationspartners

Wenn das SICCT-Terminal die Funktionalität einer sicheren Identität anbietet, da die Infrastruktur dies fordert, ist diese so zu gestalten, dass Präsentationen falscher Identitäten ausgeschlossen werden.

8.3 Bedrohungen

Nachfolgend werden Bedrohungen aufgelistet, die ein SICCT-Terminal abwehren muss.

- SICCT_B1:** Mitlesen der Kommunikation zwischen SICCT-Kartenterminal und Host
- SICCT_B2:** Angriff auf eine Kommunikation durch Unterschieben nicht autorisierter Daten
- SICCT_B3:** Umlenkung der Kommunikation über einen unautorisierten Dritten ("man in the middle")
- SICCT_B4:** Nutzung eines manipulierten SICCT-Terminals
- SICCT_B5:** Unautorisierte Erzeugung einer Terminalidentität
- SICCT_B6:** Diebstahl der Terminalidentität und Nutzung in einem anderen SICCT-Kartenterminal
- SICCT_B7:** Diebstahl eines SICCT-Terminals und Wiedereinbringung mit kompromittierter Funktion
- SICCT_B8:** Unautorisierte Registrierung eines SICCT-Terminals
- SICCT_B9:** Ausspionieren von PINs
- SICCT_B10:** Ändern von PINs ohne sie zu kennen
- SICCT_B11:** Laden einer modifizierten bzw. unautorisierten Firmware
- SICCT_B12:** Manipulation der Hardware zum Angreifen der Kommunikation zwischen SICCT-Kartenterminal und Host oder SICCT-Kartenterminal und Chipkarten

8.4 Generische Maßnahmen

Nachfolgend werden generische Maßnahmen dargestellt, die ein SICCT-Terminal abbilden muss.

- SICCT_M1:** Als Schutz vor Ausspähen von Daten wird eine verschlüsselte Verbindung (z.B. TLS) genutzt.
- SICCT_M2:** Als Schutz der Firmware und Daten im SICCT-Kartenterminal wird das Gehäuse in geeigneter Form versiegelt. (z.B. Versiegelung mit fälschungssicherem Sicherheitsaufkleber, welcher sich bei Entfernung zerstört und damit nur einmal verwendbar ist).
- SICCT_M3:** Als Schutz vor Vortäuschung einer falschen Identität des Kommunikationspartners wird eine fälschungssichere kryptografische Identität verwendet und logisch mit dem zu sichernden SICCT-Kartenterminal verbunden.
- SICCT_M4:** Als Schutz vor Manipulation am Träger der kryptografischen Identität wird diese oder die entsprechende Gehäusekomponente in geeigneter Form versiegelt.

8.5 Annahmen

Nachfolgend werden Annahmen beschrieben, die das SICCT-Terminal umsetzen muss bzw. bei der Nutzung eines SICCT-Terminals beachtet werden müssen.

- SICCT_A1:** Die Inbetriebnahme bzw. der Einsatz des SICCT-Kartenterminals erfolgt generell in einer kontrollierten Umgebung.
- SICCT_A2:** Die Integration bzw. das Laden der kryptografischen Identität erfolgt in einer kontrollierten Umgebung.
- SICCT_A3:** Der Betrieb des SICCT-Kartenterminals erfolgt über eine ansteuernde Host-Instanz.
- SICCT_A4:** Das SICCT-Kartenterminal verfügt über die Möglichkeit zur Transportsicherung (z.B. auf Basis SSL/TLS).
- SICCT_A5:** Das SICCT-Kartenterminal kann so konfiguriert werden, dass nur gesicherte Kommunikationen zwischen Host und SICCT-Terminal möglich sind.
- SICCT_A6:** Das SICCT-Kartenterminal verfügt eine einmalige interne Gerätenummer sowie über eine einmalige von außen sichtbare Seriennummer.
- SICCT_A7:** Die elektronische Komponente des Trägers der kryptografischen Identität verfügt über eine sichtbar aufgebrachten einmalige Serien- oder Identifikationsnummer.
- SICCT_A8:** Es wird eine einmalige Kartenterminalidentität genutzt.
- SICCT_A9:** SICCT-Kartenterminal und ansteuernde Host-Instanz werden gepaart, so dass die Kartenterminalidentität an eine Hardwarekombination gebunden ist.
- SICCT_A10:** Für den Aufbau der sicheren Verbindung zwischen Host und SICCT-Terminal werden sowohl auf Host- als auch auf Terminalseite Zertifikate (z.B. X509) benötigt.
- SICCT_A11:** Die ansteuernde Host-Instanz verfügt über eine Whitelist von zugelassenen Zertifikaten.
- SICCT_A12:** Der Nutzer kontrolliert die Unversehrtheit des SICCT-Kartenterminals.
- SICCT_A13:** Der Nutzer kontrolliert die Unversehrtheit des bzw. der Sicherheitssiegel.
- SICCT_A14:** Der Nutzer gibt die PIN unbeobachtet am Keypad des SICCT-Kartenterminals ein.
- SICCT_A15:** Der Nutzer kontrolliert während der PIN Eingabe die Anzeige des Sicherheitsmodus (z.B. LED oder Schlosssymbole im Display).

8.6 Sicherheitsmechanismen

Nachfolgend werden Sicherheitsmechanismen beschrieben, die das SICCT-Terminal umsetzen muss bzw. bei der Nutzung eines SICCT-Terminals sichergestellt werden müssen.

- SICCT_SM1:** Verschlüsselung der Kommunikation zwischen SICCT-Kartenterminal und ansteuernder Host-Instanz.
- SICCT_SM2:** Einseitige Authentisierung des SICCT-Terminals durch ansteuernde Host-Instanz durch Nutzung einer kryptographischen Terminalidentität.
- SICCT_SM3:** Authentische Kommunikation zwischen SICCT-Kartenterminal und Host.
- SICCT_SM4:** Versiegelung der Gehäuse.
- SICCT_SM5:** Erzeugung eines Kartenterminal-individuellen Authentisierungsdatensatzes im ansteuernden Host aus Kryptografischer Terminalidentität und Terminalkennung die individuell, nicht vorhersagbar und eindeutig ist.
- SICCT_SM6:** Authentisierungsdaten bestehen aus einer kryptographischen Identität des SICCT-Terminals und einmaligen Daten der beteiligten Komponenten.
- SICCT_SM7:** Aufbau einer verschlüsselten Verbindung (/z.B. TLS) zur verschlüsselten Übertragung der Authentisierungsdaten.
- SICCT_SM8:** Inbetriebnahme und Registrierung des SICCT-Terminals am Host durch einen Administrator.
- SICCT_SM9:** Die Registrierung eines SICCT-Terminals am Host erfolgt in einer überschaubar sicheren Umgebung. So ist z.B. administrativ sicherzustellen, dass in dieser Zeit kein Zugang von nicht autorisiertem Personal stattfindet.
- SICCT_SM10:** Abmeldung (Deregistrierung) defekter oder entwendeter SICCT-Terminals in Host-Instanz durch einen Administrator.
- SICCT_SM11:** Automatisches Abmelden des Terminals in Host-Instanz bei fehlerhafter SICCT-Terminalidentität mit Notwendigkeit der Neuansmeldung.
- SICCT_SM12:** Die Firmware garantiert, dass die Kommandos zum Verifizieren und Modifizieren der PIN erkannt werden und durch Sicherheitsfunktionen zur "Sicheren PIN-Eingabe" bearbeitet werden. Diese Sicherheitsfunktionen gewährleisten, dass die PIN-Eingabe über Das Keypad des SICCT-Chipkartenterminals sicher an die Chipkarte weitergeleitet wird, ohne vom Host ausgespäht werden zu können. Die Speicherbereiche der PIN-Daten werden so durch die Firmware aufbereitet, dass kein Angriff auf gespeicherte Daten möglich ist. Das SICCT-Terminal erkennt die von der Host-Software übermittelten Kommandos zur PIN-Eingabe und fügt die über das Keypad des SICCT-Terminals eingegebenen Nummern als PIN an die entsprechenden Stellen des Kommandos an die Chipkarte ein. Die PIN selbst verlässt das SICCT-Terminal nie in Richtung Host.
- Die in nachfolgender Liste aufgeführten zur sicheren PIN-Eingabe unterstützten Instruction-Bytes sind von den Applikationen zu verwenden und von den Chipkarten spezifikationsgemäß zu unterstützen bzw. bei Nicht-Unterstützung mit einer geeigneten Fehlermeldung abzulehnen:
- VERIFY (ISO/IEC 7816-4): INS=0x20
 - CHANGE REFERENCE DATA (ISO/IEC 7816-4): INS=0x24
 - ENABLE VERIFICATION REQUIREMENT (ISO/IEC 7816-4): INS=0x28
 - DISABLE VERIFICATION REQUIREMENT (ISO/IEC 7816-4): INS=0x26
 - RESET RETRY COUNTER (ISO/IEC 7816-4): INS=0x2C
 - PERFORM SECURITY OPERATION: INS=0x2A

- SICCT_SM13:** Der Mode der sicheren PIN Eingabe wird dem Anwender optisch (LED oder Display) angezeigt.
- SICCT_SM14:** Die Anzeige des Sicherheitsmodus (z.B. LED oder Schlosssymbole im Display für die sichere PIN Eingabe) kann nicht von außen simuliert werden. Nur die dafür vorgesehenen Befehle des SICCT Command Sets können die Anzeige des Sicherheitsmodus bewirken.

9 Anmerkungen zum Dokumentenstand

9.1 Dokumentenstand und Haftungsausschluss

Die vorliegende Version der SICCT-Spezifikation gibt den momentanen Stand der Arbeiten wieder, aus dem Zielsetzung und Lösungsansätze ersichtlich sind. Das Dokument in der jetzigen Form kann als Basis für andere Spezifizierungsarbeiten oder technische Entwicklungen dienen.

Für die Richtigkeit, Vollständigkeit und Aktualität der Inhalte können die Autoren keine Gewähr übernehmen.

Die folgenden Auflistungen fokussieren einige wichtige Punkte, die als unerledigt oder unentschieden bekannt sind. Die Listen erheben keinerlei Anspruch auf Vollständigkeit mit der Konsequenz, dass das Dokument an jeder Stelle grundlegend oder im Detail noch späteren Erweiterungen unterworfen sein kann.

9.2 Errata-Beschreibung

Das vorliegende Dokument beinhaltet die SICCT-Basispezifikation [SICCT_130] sowie alle bis zum Ausgabedatum bekannten und von der TeleTrust-AG "SICCT" bearbeiteten Errata-Korrekturen zu der Spezifikation "Secure Interoperable Chipcard Terminal". Bisher wurden Errata-Korrekturen von der TeleTrust-AG "SICCT" in einem separaten Dokument und als Errata Sheet z.B. [[SICCT_121_Errata_1_0_2]] vom 28.10.2015 als Ergänzung zur SICCT Spezifikation 1.21 veröffentlicht. Ggf. zukünftige ERRATA-Dokumente werden im Dateinamen einen Versionsbezug der SICCT-Basispezifikation (z.B. [SICCT_130_Errata_X_y_z]) erhalten.

9.3 Zu bearbeitende Punkte

10 Abkürzungen

AM	Access Modes
APDU	Application Protocol Data Unit
API	Application Programming Interface
ATR	Answer to Reset
CRT	Control Reference Template
CT-API	CardTerminal Application Programming Interface
CT-BCS	CardTerminal Basic Command Set
DF	Dedicated File
EF	Elementary File
EMV	Europay/Master/Visa Kartenspezifikation
GUID	Global Unique Identifier
ICC	Integrated Circuit Card
ICCSP	Integrated Circuit Card Service Provider
IEC	International Electrotechnical Commission
IFD	Interface Device
ISO	International Standardization Organisation
KVK	Krankenversichertenkarte / German Health Insurance Card
OID	Object Identifier
PC	Personal Computer
PC/SC	Interoperability Specification for ICCs and PC Systems (References)
PIN	Personal Identification Number

POS	Point of Sale
SAM	Security Application Module
SC	SmartCard
CRT	Control Reference Template
SP	Service Provider
TLV	Tag-Length-Value

11 Referenzen

[ISO/IEC 7810] [STD1]	Identification cards — Physical characteristics, 4 th edition, 2019-12
	Amendment 1: Additional requirements for integrated circuit cards with contacts, 2024-01
[ISO/IEC 7816-2] [STD4]	Identification Cards – Integrated circuit(s) Part 2: Cards with contacts –Dimensions and location of the contacts, 2 nd edition, 2007-10
[ISO/IEC 7816-3] [STD6]	Identification Cards – Integrated circuit cards - Part 3: Cards with contacts – Electrical interface and transmission protocols, 3 rd edition, 2006-11
[ISO/IEC 7816-4] [STD8]	Identification Cards – Integrated circuit(s) cards with contacts – Part 4: Organization, security and commands for interchange, 4 th edition, 2020-05
[ISO/IEC 7816-10] [STD9]	Identification Cards – Integrated circuit(s) cards with contacts – Part 10: Electronic signals and answer to reset for synchronous cards, 1 st edition, 1999-11, corrected version 2000-02
[ISO/IEC 7816-12]	Identification cards - Integrated circuit cards, Part 12: Cards with contacts — USB electrical interface and operating procedures, 1 st edition, 2005-10
	Technical Corrigendum 1, 2014-03
[ISO/IEC 10646]	Information technology — Universal coded character set (UCS) 6 th edition, 2020-12
[ISO/IEC 14443-1] [STD10]	Cards and security devices for personal identification — Contactless proximity objects, Part 1: Physical characteristics, 4 th edition, 2018-04
[ISO/IEC 14443-2] [STD11]	Cards and security devices for personal identification — Contactless proximity objects, Part 2: Radio frequency power and signal interface, 4 th edition, 2020-07
	Technical Corrigendum 1, 2021-06
	Technical Corrigendum 2, 2023-09
[ISO/IEC 14443-3] [STD13]	Cards and security devices for personal identification — Contactless proximity objects, Part 3: Initialization and anticollision, 4 th edition, 20218-07
	Amendment 2: Enhancements for harmonization, 2020-08
[ISO/IEC 14443-4] [STD15]	Identification Cards – Contactless integrated circuit(s) cards – Proximity cards, Part 4: Transmission protocol, 4 th edition, 2018-06
	Amendment 2: Enhancements for harmonization, 2020-08
[ISO/IEC 16022]	Information technology — Automatic identification and data capture techniques — Data Matrix bar code symbology specification, 3 rd edition, 2024-05
[ISO/IEC 18004]	Information technology — Automatic identification and data capture techniques — QR code bar code symbology specification, 4 th edition, 2024-08
[STD16]	CEN ENV 1375-1: 1994, Identification card systems - Intersector integrate circuit(s) card additional formats – Part 1: ID-000 card size and physical characteristics

[STD17]	ISO 9564-1:2017, Financial services — Personal Identification Number (PIN) management and security, Part 1: Basic principles and requirements for PINs in card-based systems, Published (Edition 4, 2017) https://www.iso.org/standard/68669.html
[STD18]	ISO standard 7498-1:1994; http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip .
[STD19]	CEN, EN 726-3:1994(Main), Identification card systems - Telecommunications integrated circuit(s) cards and terminals - Part 3: Application independent card requirements https://standards.iteh.ai/catalog/standards/cen/d31ba87a-bc75-4dbe-bf5f-0d8bd2fb2c8a/en-726-3-1994
[STD20]	CEN, EN 1332-3, Identification card systems - User Interface - Part 3: Key pads https://standards.globalspec.com/std/14306099/en-1332-3
[EMV_40]	EMV Integrated Circuit Card Specifications for Payment Systems – Book 1: Application Independent ICC to Terminal Interface Requirements (EMVCo, Version 4.0, December 2000)
[EMV_41]	EMV Integrated Circuit Card Specifications for Payment Systems – Book 1: Application Independent ICC to Terminal Interface Requirements (EMVCo, Version 4.1, May 2004)
[EMV_44]	EMV Integrated Circuit Card Specifications for Payment Systems – Book 1: Application Independent ICC to Terminal Interface Requirements (EMVCo, Version 4.4, October 2022)
[MKT_10]	Multifunktionale KartenTerminals - MKT, MKT-Version 1.0, 15.04.1999 TeleTrusT, http://www.teletrust.de/publikationen/spezifikationen/mkt/
[PCSC2]	Interoperability Specification For ICCs and Personal Computer Systems – Part 2: Interface Requirements for Compatible IC Cards and Readers (Revision 2.01, June 2005)
[PCSC3]	Interoperability Specification For ICCs and Personal Computer Systems – Part 3: Requirements for PC-Connected Interface Devices (Revision 2.01, June 2005)
[IND1]	I2C Peripherals – Data Handbook IC12 (Philips Semiconductors, 1997 Apr 07)
[IND2]	Infineon-Datenbuch, beim Hersteller anzufragen
[IND3]	Infineon-Datenbuch, beim Hersteller anzufragen
[gemSpec_KT]	Einführung der Gesundheitskarte, Spezifikation eHealth-Kartenterminal, Version: 2.8.0 http://www.bmg.bund.de/SharedDocs/Downloads/DE/GV/GT/Gesundheitskarte/Technische_20Festlegungen_20fuer_20die_20Testverfahren/Dezentrale_20Komponenten/eHealth-Kartenterminal,templateId=raw,property=publicationFile.pdf/eHealth-Kartenterminal.pdf http://www.bitkom.org/files/documents/gematik_KT_eHealth_Kartenterminal_V2.8.0.pdf
[RFC 768]	User Datagram Protocol, http://www.ietf.org/rfc/rfc768.txt
[RFC 791]	INTERNET PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION http://www.ietf.org/rfc/rfc791.txt

[RFC2132]	DHCP Options and BOOTP Vendor Extensions, http://www.ietf.org/rfc/rfc2132.txt
[RFC 2136]	Dynamic Updates in the Domain Name System (DNS UPDATE), http://www.ietf.org/rfc/rfc2136.txt
[RFC 3268]	Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS) , http://www.ietf.org/rfc/rfc3268.txt
[RFC 3330]	Special-Use IPv4 Addresses, http://www.ietf.org/rfc/rfc3330.txt
[RFC3927]	Dynamic Configuration of IPv4 Link-Local Addresses, http://www.ietf.org/rfc/rfc3927.txt
[RFC791]	RFC 791: Internet Protocol; DARPA; J. Postel; 1981.
[RFC793]	RFC 793: Transmission Control Protocol; DARPA; 1981.
[RFC768]	RFC 768: User Datagram Protocol; J. Postel; 1980.
[RFC1122]	RFC 1122: Requirements for Internet Hosts - Communication Layers; IETF; R. Braden; 1989.
[RFC1918]	RFC 1918: Address Allocation for Private Internets; Cisco, Crysler, RIPE, Silicon Graphics; http://www.ietf.org/rfc/rfc1918.txt ; 1996.
[RFC2131]	RFC 2131: Dynamic Host Configuration Protocol; Bucknell University; R. Droms; http://www.ietf.org/rfc/rfc2131.txt .
[RFC2136]	RFC 2136: Dynamic Updates in the Domain Name System (DNS UPDATE); IEC, Bellcore, Cisco, Dell; http://www.ietf.org/rfc/rfc2136.txt ; 1997.
[RFC3927]	RFC 3927: Dynamic Configuration of IPv4 Link-Local Addresses; Apple Computer, Microsoft Corporation, Sun Microsystems; http://www.ietf.org/rfc/rfc3927.txt ; 2005.
[RFC3927]	RFC 3827: Dynamic Configuration of IPv4 Link-Local Addresses; Apple Computer, Microsoft Corporation, Sun Microsystems; http://www.ietf.org/rfc/rfc3927.txt ; 2005.
[RFC3330]	RFC 3330: Special-Use IPv4 Addresses; IANA; http://www.ietf.org/rfc/rfc3330.txt ; 2002.
[RFC2246]	RFC 2246: The TLS Protocol Version 1.0; Certicom; http://www.ietf.org/rfc/rfc2246.txt ; 1999.
[RFC5246]	RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2. ; http://tools.ietf.org/rfc/rfc5246.txt ; 2008;
[RFC3268]	RFC 3268: Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS); Skygate Technology; http://www.ietf.org/rfc/rfc3268.txt ; 2002.
[RFC3280]	RFC 3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile; RSA Laboratories, NIST, VeriSign, CitiGroup; http://www.ietf.org/rfc/rfc3280.txt ; 2002.
[RFC 8446]	The Transport Layer Security (TLS) Protocol Version 1.3, August 2018, https://datatracker.ietf.org/doc/html/rfc8446
[PCSC20]	Interoperability Specification for ICC and Personal Computer Systems Part 10: IFD with Secure Pin Entry Capabilities; PCSC Workgroup; 2005.

Anhang A (Normativ)

ISO / EMV Vergleichstabelle

Der folgende Vergleich zeigt die unterschiedlichen Parameter (z.B. Werte, Anforderungen, Verhaltensweisen) folgender beiden Normen auf:

- ISO/IEC 7816-3, siehe [STD6]
- EMV 4.1, siehe [EMV_41], wobei die elektrische Parameter den Werten für "Neue Terminals ab Juli 2009" entsprechen

Parameter, bei denen keine Unterschiede festgestellt wurden, sind nicht aufgeführt.

In der letzten Spalte wurde eine Zuordnung getroffen, welche Parameter bei einem SICCT-Terminal einzuhalten sind. Maßgabe war bei dieser Zuordnung, dass möglichst wenig Parameter im Terminal umschaltbar gestaltet werden müssen. Nur in den Fällen, wo die Parameter aus ISO und EMV unvereinbar sind, muss ein Terminal, das auch den optionalen EMV-Mode unterstützt, im "ISO-Mode" und im "EMV-Mode" unterschiedliche Parameter unterstützen.

Dieses Verfahren stellt sicher, dass ein SICCT-Terminal einen vollständigen EMV-Level-1-Test (zurzeit nur gemäß [EMV_40] verfügbar) im EMV-Mode bestehen kann. In diesem Fall wären dann im ISO-Mode nur noch die Differenzen abzutesten.

Soll alternativ kein EMV-Level-1-Test für ein SICCT-Terminal gemacht werden, wäre ein vollständiger Test im ISO-Mode notwendig. In diesem Fall benötigt das Terminal keinen EMV-Mode, die Umschaltung wäre entbehrlich.

Unterstützt ein SICCT-Terminal diese Mode-Umschaltung, muss die Default-Einstellung "ISO-Mode" sein.

Description	ISO 7816-3		EMV 4.1		Value to be used by SICCT	
	Ref.	Value	Ref.	Value	ISO Mode	EMV Mode (note 1)
A1: Electrical Characteristics						
1. VCC class selection	6.2.4	see 6.2.4, Fig. 3	5.5.6	note 4	like ISO, starting with 5V	
2. VCC class A	5.2.1, Tab. 1	4.5...5.5V max. 60mA	5.5.6	4.6...5.4V min. 55mA	4.6...5.4V min. 60mA	
3. VCC class B	5.2.1, Tab. 1	2.7...3.3V max. 50mA	5.5.6	2.76...3.24V min. 55mA	like EMV	
4. VCC class C	5.2.1, Tab. 1	1.62...1.98V max. 30mA	5.5.6	1.66...1.94V min. 35mA	like EMV (note 1)	
5. VCC class A spikes	5.2.1, Tab. 2	20nAs @ 400ns/100mA	5.5.6	30nAs @ 400ns/100mA	like EMV	
6. VCC class B spikes	5.2.1, Tab. 2	10nAs @ 400ns/50mA	5.5.6	17,5nAs @ 400ns/50mA	like EMV	
7. VCC class C spikes	5.2.1, Tab. 2	6nAs @ 400ns/30mA	5.5.6	11,1nAs @ 400ns/30mA	like EMV, note 1	
8. VCC current limit	note 3		note in 5.5.6	max. 200mA (note 2)	like EMV	
9. VCC signal perturbations low voltage	note 3		5.5.6	min. -0.25V	like EMV	
10. I/O ICC input high voltage VIH (VOH in terminal transmission mode)	5.2.5, Tab. 5	min. 0.7*Vcc @ -300µA max. Vcc @ 20µA (note 5)	5.5.2.1	min. 0.8*Vcc @ 20µA max. Vcc	like EMV	
11. I/O ICC input low voltage VIL (VOL in terminal transmission mode)	5.2.5, Tab. 5	min. 0V @ -1000µA max. 0.15*Vcc @ 20µA (note 5)	5.5.2.1	min. 0V max. 0.15*Vcc @ -500µA	like EMV	
12. I/O rise and fall times (terminal transmission mode)	5.2.5, Tab. 5	max. 1µs	5.5.2.1	max. 0.8µs	like EMV	
13. I/O signal perturbations low voltage (VOL in terminal transmission mode)	5.2.5, Tab. 5	min. -0.3V	5.5.2.1	min. -0.25V max. 0.15*Vcc	like EMV	
14. I/O signal perturbations high voltage (VOH in terminal transmission mode)	5.2.5, Tab. 5	max. Vcc+0.3V	5.5.2.1	min. 0.8*Vcc max. Vcc+0.25V	like EMV	
15. I/O ICC output high voltage VOH (VIH in terminal reception mode)	5.2.5, Tab. 5	min. 0.7*Vcc max. Vcc @ 20µA Terminal Pull-Up 20k	5.5.2.2	min. 0.6*Vcc max. Vcc	like EMV	

16. I/O ICC output low voltage VOL (VIL in terminal reception mode)	5.2.5, Tab. 5	min. 0V max. $0.15 \cdot V_{CC} + 1000\mu A$ (class C: $+500\mu A$) @	5.5.2.2	min. 0V max. $0.2 \cdot V_{CC}$	like EMV max. ICC current $500\mu A$ (class A: $1000\mu A$)
17. I/O rise and fall times (terminal reception mode)	5.2.5, Tab. 5	max. $1\mu s$	5.5.2.2	max. $1.2\mu s$	like EMV
18. I/O current limit	note 3		5.5.2	max. $\pm 15mA$	like EMV
19. CLK high voltage	5.2.3, Tab. 4	min. $0.7 \cdot V_{CC}$ @ $-20\mu A$ max. V_{CC} @ $100\mu A$ (note 5)	5.5.4	min. $0.8 \cdot V_{CC}$ @ $50\mu A$ max. V_{CC}	like EMV
20. CLK low voltage	5.2.3, Tab. 4	min. 0V @ $-100\mu A$ max. $0.5V$ (Class C: $0.2 \cdot V_{CC}$) @ $20\mu A$ (note 5)	5.5.4	min. 0V max. $0.15 \cdot V_{CC}$ @ $-50\mu A$	like EMV
21. CLK rise and fall times	5.2.3, Tab. 4	max. 9% of CLK period	5.5.4	max. 8% of CLK period	like EMV
22. CLK signal perturbations low voltage	5.2.3, Tab. 4	min. $-0.3V$	5.5.4	min. $-0.25V$ max. $0.15 \cdot V_{CC}$	like EMV
23. CLK signal perturbations high voltage	5.2.3, Tab. 4	max. $V_{CC} + 0.3V$	5.5.4	min. $0.8 \cdot V_{CC}$ max. $V_{CC} + 0.25V$	like EMV
24. CLK signal duty cycle	5.2.3	min. 40% of CLK period max. 60% of CLK period	5.5.4	min. 45% of CLK period max. 55% of CLK period	like EMV
25. CLK frequency stability	note 3		5.5.4	max. $\pm 1\%$	like EMV
26. RST high voltage	5.2.2, Tab. 3	min. $0.8 \cdot V_{CC}$ @ $-20\mu A$ max. V_{CC} @ $150\mu A$ (note 5)	5.5.4	min. $0.8 \cdot V_{CC}$ @ $50\mu A$ max. V_{CC}	like EMV
27. RST low voltage	5.2.2, Tab. 3	min. 0V @ $-200\mu A$ max. $0.12 \cdot V_{CC}$ @ $20\mu A$ note 5	5.5.4	min. 0V max. $0.15 \cdot V_{CC}$ @ $-50\mu A$	like EMV
28. RST rise and fall times	5.2.2, Tab. 3	max. $1\mu s$	5.5.4	max. $0.8\mu s$	like EMV
29. RST signal perturbations low voltage	5.2.2, Tab. 3	min. $-0.3V$	5.5.4	min. $-0.25V$ max. $0.15 \cdot V_{CC}$	like EMV
30. RST signal perturbations high voltage	5.2.2, Tab. 3	max. $V_{CC} + 0.3V$	5.5.4	min. $0.8 \cdot V_{CC}$ max. $V_{CC} + 0.25V$	like EMV
31. VPP	5.2.4	SPU function: rfu	5.5.3	C6 isolated ($>10M\Omega$) or $0 \dots 1.05 \cdot V_{CC}$ (class A terminal only)	C6 isolated
32. contact resistance	note 3		5.5.7	max. $500 m\Omega$	like EMV
33. short circuit resilience	note 3		5.5.8		like EMV
34. powering and depowering of terminal with ICC in place	note 3		5.5.9		like EMV
A2: Card Operating Procedure (Activation, Reset, PPS, Deactivation)					

1. voltage levels before activation	6.2.1	RST = State L	6.1.2	RST, CLK, I/O = VOL VCC ≤ 0.4V	like EMV
2. activation sequence	6.2.1	1. VCC powered 2. I/O in reception mode 3. CLK activated	6.1.2	1. VCC powered 2. I/O in reception mode 3. CLK activated	no difference anymore
3. cold reset: time t_b (see ISO Fig. 1)	6.2.2	$t_b \geq 400T_{clk}$	6.1.3.1	$40000T_{clk} \leq t_b \leq 45000T_{clk}$	like EMV
4. cold reset: time t_c (see ISO Fig. 1)	6.2.2	$400T_{clk} \leq t_c \leq 40000T_{clk}$	6.1.3.1	terminal reception window: $380T_{clk} \leq t_c \leq 42000T_{clk}$	like EMV
5. cold reset: time to deactivation without ATR	6.2.2	min. $40000T_{clk}$	6.1.3.1	min. $42001T_{clk}$ max. $42000T_{clk} + 50ms$	like EMV
6. warm reset: time t_e (see ISO Fig. 2)	6.2.3	$t_e \geq 400T_{clk}$	6.1.3.2	$40000T_{clk} \leq t_e \leq 45000T_{clk}$	like EMV
7. warm reset: time t_f (see ISO Fig. 2)	6.2.3	$400T_{clk} \leq t_f \leq 40000T_{clk}$	6.1.3.2	terminal reception window: $380T_{clk} \leq t_f \leq 42000T_{clk}$	like EMV
8. warm reset: time to deactivation without ATR	6.2.3	min. $40000T_{clk}$	6.1.3.2	min. $42001T_{clk}$ max. $42000T_{clk} + 50ms$	like EMV
9. clock stop mode	6.3.2		note 4		like ISO
10. PPS	9		note 4		like ISO
11. deactivation sequence	6.4	1. RST = State L 2. CLK = State L 3. I/O = state L 4. VCC deactivated	6.1.5	1. RST = State L 2. CLK and I/O = State L 3. VCC deactivated	like ISO
12. voltage levels after deactivation	6.4	RST, CLK = State L I/O = state L	6.1.5	RST, CLK, I/O = VOL VCC ≤ 0.4V	like EMV
13. max. deactivation time	note 3		6.1.5	max. 100ms	like EMV
14. abnormal deactivation	note 3		6.2		like EMV
A3: ATR Timing and Error Handling					
1. min. delay between ATR bytes	8.1	12etu	8.4	11.8etu	like EMV
2. max. delay between ATR bytes (initial wait time)	8.1	9600etu	8.4	10080etu	like EMV
3. max. time to deactivation if timeout (max. delay between ATR bytes)	note 3		8.4	14400etu (start of last byte to RST=L)	like EMV
4. max. time for receiving all ATR bytes	note 3		8.4	20160etu	like EMV
5. max. time to deactivation if timeout (max. time for receiving all ATR bytes)	note 3		8.4	24000etu (start of TS to RST=L)	like EMV

6. max. time to deactivation if ICC is rejected	note 3		8.4	24000etu (start of TS to RST=L)	like EMV
7. max. time to warm reset, if cold ATR is rejected	note 3		8.4	24000etu (start of TS to RST=L)	like EMV
8. max. time to deactivation, if warm ATR is rejected	note 3		8.4	24000etu (start of TS to RST=L)	like EMV
9. max. time to deactivation, if parity error at any ATR byte	note 3		8.4	24000etu (start of TS to RST=L)	like EMV
10. min. guard time after last ATR byte to first transmission	note 3		8.4	16 (T=0) or 22etu (T=1)	like EMV
11. T=0 parity error signaling during ATR	8.1	optional for the terminal	9.2.3	not allowed for the terminal	like EMV
A4: ATR Coding					
1. TS unequal '3F' or '3B'	note 3		8.3.1	reject ICC (deactivation)	like EMV
2. TA1 is absent	8.3	default TA1='11'	8.3.3.1	default TA1='01' or TA1='11'	default TA1='11' (f _{max} =5MHz)
3. TB1 is absent or TB1 unequal '00' (cold ATR)	8.3	ignore TB1 and accept ATR	8.3.3.2	reject ATR (note 6)	like ISO like EMV
4. TB1 is absent or TB1 with any value (warm ATR)	8.3	ignore TB1 and accept ATR	8.3.3.2	accept ATR as though TB1='00' (default)	like EMV
5. TD1 containing a protocol unequal T=0 or T=1	8.2.3	if protocol is supported: accept ATR	8.3.3.4	reject ATR (note 6)	like ISO like EMV
6. TA2 is present (specific mode), use of b8 (if specific parameters not supported by terminal)	8.3	b8=0: warm reset b8=1: deactivation	8.3.3.5	b8 should be ignored: reject ATR (note 6)	like EMV
7. TA2 is present (specific mode), use of b5	8.3	b5=0: F/D defined by TA1 b5=1: F/D implicitly defined	8.3.3.5	b5=0: accept ATR if F/D values of TA1 are supported b5=1: reject ATR (note 6)	like EMV
8. TA2 is present (specific mode), use of b4...b1	8.3	protocol to be used	8.3.3.5	if protocol is unequal first indicated protocol: reject ATR (note 6)	like ISO like EMV
9. TB2 is present	8.3	ignore TB2 and accept ATR	8.3.3.6	reject ATR (note 6)	like ISO like EMV
10. TC2 = '00' (T=0 only)	10.2	rfu	8.3.3.7	reject ATR (note 6)	like EMV

11. TD2 containing a protocol unequal T=1 or T=14 (only if T=0 in TD1)	8.2.3	if protocol is supported: accept ATR	8.3.3.8	reject ATR (note 6)	like ISO	like EMV
12. allowed values for TA3 (IFSC) (T=1 only)	11.4.2	TA3='00' and TA3='FF': rfu	8.3.3.9	reject ATR (note 6), if TA3='00...0F' or if TA3='FF'	like ISO	like EMV
13. allowed values for low nibble of TB3 (CWI) (T=1 only)	11.4.3	all values allowed (CWI='0'...'F')	8.3.3.10	reject ATR (note 6), if CWI='6'...'F' or if $2^{CWI} \leq N+1$ (TC1)	like ISO	like EMV
14. allowed values for high nibble of TB3 (BWI) (T=1 only)	11.4.3	BWI='A'...'F': rfu	8.3.3.10	reject ATR (note 6), if BWI='5'...'F'	like ISO	like EMV
15. TB3 is absent (T=1 only)	11.4.3	accept ATR as though TB3='4D' (default)	8.3.3.10	reject ATR (note 6)	like ISO	like EMV
16. allowed values for TC3 (EDC) (T=1 only)	11.4.4	TC3 unequal '00' or '01' rfu If CRC is supported, accept ATR with TC3='01'	8.3.3.9	reject ATR (note 6), if TC3 unequal '00'	like ISO	like EMV
17. T _{Ai} (clock stop and voltage class)	8.3	accept ATR, if values supported	note 4		like ISO	
18. T _{Bi} (use of SPU/C6)	8.3	accept ATR, if values supported	note 4		like ISO	
A5: T=0 Protocol, Timing and Error Handling						
1. min. delay between received bytes	8.3	12etu	9.2.2.1	11.8etu	like EMV	
2. max. delay WWT	10.2	WWT=960*D*W _I [etu]	9.2.2.1	reception until WWT+(D*480) [etu]	like EMV	
3. max. time to deactivation after WWT timeout	note 3		9.2.2.1	WWT+(D*9600) [etu] (start of last byte to RST=L)	like EMV	
4. min. guard time after transmission to reception	note 3		9.2.2.1	reception within 15etu	like EMV	
5. min. guard time after reception to transmission	10.2	16etu (D=64 only)	9.2.2.1	16etu	like EMV	
6. number of character repetitions after parity error signaling (transmission and reception)	note 3		9.2.3	max. 4 times (5 times totally)	like EMV	
7. max. time to deactivation after last repetition (parity error)	note 3		9.2.3	D*960 [etu] (invalid byte to RST=L)	like EMV	
8. max. time to deactivation after wrong procedure or status byte	note 3		9.2.3	9600etu (invalid byte to RST=L)	like EMV	

9. definition of CLA byte in command GET RESPONSE	12.2.1	identical to APDU CLA byte	9.3.1.3	CLA='00'	like ISO	like EMV
A6: T=1 Protocol, Timing and Error Handling						
1. min. delay between received bytes	8.3, 11.2	if TC1='FF': 11etu else 12etu	9.2.4.2.2	10.8etu		like EMV
2. max. delay between received bytes (CWT)	11.4.3	$CWT = 11 + 2^{CWI} [etu]$	9.2.4.2.2	reception until $CWT + 4 [etu]$		like EMV
3. terminal action after CWT timeout	note 3		9.2.5.1	deactivation or error handling		error handling
4. max. time to deactivation or error handling after CWT timeout	note 3		9.2.5.1	$CWT + 4800 [etu]$ (start of last byte received to $RST=L$ or start of first byte for error handling)		$CWT + 4800 [etu]$ (start of last byte received to start of first byte for error handling)
5. max. time to reception after transmission (BWT)	11.4.3	$BWT = 11 + 2^{BWI} * 960 * 372 * D/F [etu]$	9.2.4.2.2	reception until $BWT + (D * 960) [etu]$		like EMV
6. terminal action after BWT timeout	11.6.3.2	error handling	9.2.5.1	deactivation or error handling		error handling
7. max. time to deactivation or error handling after BWT timeout	note 3		9.2.5.1	$BWT + (D * 4800) [etu]$ (start of last byte transmitted to $RST=L$ or start of first byte for error handling)		$BWT + (D * 4800) [etu]$ (start of last byte transmitted to start of first byte for error handling)
8. max. time to reception after waiting time extension (WTX)	11.6.2.3	$WTX = n * BWT [etu]$	9.2.5.1	reception until $(n * BWT) + (n * D * 960) [etu]$		like EMV
9. terminal action after WTX timeout	note 3		9.2.5.1	deactivation or error handling		error handling
10. max. time to deactivation or error handling after WTX timeout	note 3		9.2.5.1	$(n * BWT) + (n * D * 4800) [etu]$ (start of last byte transmitted to $RST=L$ or start of first byte for error handling)		$(n * BWT) + (n * D * 4800) [etu]$ (start of last byte transmitted to start of first byte for error handling)
11. min. guard time after transmission to reception (BGT)	11.2	22etu	9.2.2.1	reception within 21etu		like EMV
12. min. guard time after reception to transmission (BGT)	11.2	22etu	9.2.2.1	22etu		like EMV
13. NAD	11.3.2.1	NAD used for node addressing	9.2.4.1.1	NAD='00', otherwise error handling		like EMV
14. LEN, I-block with LEN='00'	11.3.3	allowed	9.2.4.1.1	not supported		like ISO
15. IFSD	11.4.2	default '20' S(IFS request) with '01' ... 'FE' allowed	9.2.4.1.1	S(IFS request) with 'FE' at start of protocol is mandatory		like EMV

16. terminal support for S(RESYNCH request)	A.3.5	mandatory	9.2.5.1	optional (not supported in level 1 test cases)	like ISO	like EMV
17. terminal action after sending 3 consecutive blocks without obtaining a valid response (at the start of the protocol)	A.3.5	warm reset or deactivation	9.2.5.1	Deactivation within BWT+(D*14400) [etu] (start of last byte transmitted to RST=L)	like EMV	
18. terminal action after sending 3 consecutive blocks without obtaining a valid response (during the protocol)	A.3.5	sending S(RESYNCH request)	9.2.5.1	Deactivation within BWT+(D*14400) [etu] (start of last byte transmitted to RST=L)	like ISO	like EMV
19. terminal action after sending 3 S(RESYNCH request) without obtaining a valid response	A.3.5	warm reset or deactivation	note 4		like ISO	
20. terminal support of S(ABORT request)	A.3.4	allowed	9.2.5.1	not allowed	like EMV	
21. terminal action after receiving a S(ABORT request) block	A.3.4	sending S(ABORT response) and termination of chaining operation	9.2.5.1	Deactivation within D*9600 [etu] (start of last byte received to RST=L)	like ISO	like EMV
22. terminal support of S(Error on VPP state) block	11.3.2.2	not supported	9.2.4.1.1	not supported, otherwise error handling	like EMV	

ANMERKUNGEN

- Note 1 if supported by terminal
- Note 2 recommended only
- Note 3 not specified at ISO standard
- Note 4 not specified at EMV standard
- Note 5 specified values for currents are questionable
- Note 6 Rejecting a cold ATR means: performing a warm reset.
Rejecting a warm ATR means: performing a deactivation.

Anhang B (Informativ)

eCard Strategie des Bundes / ECard Framework

1 Architektur

Hauptziel des eCard-Frameworks ist es, Anwendungsentwicklern eine möglichst einfache und homogene Schnittstelle über aktuell verfügbare und zukünftige elektronische Services (ePassport, eHealth, Elster etc.) zur Verfügung zu stellen. Gleichzeitig sollen aber möglichst geringe Einschränkungen für bereits bestehende Dienste entstehen.

Die Architektur des eCard-Frameworks unterteilt sich in drei wesentliche Ebenen, die nachfolgend beschrieben sind.

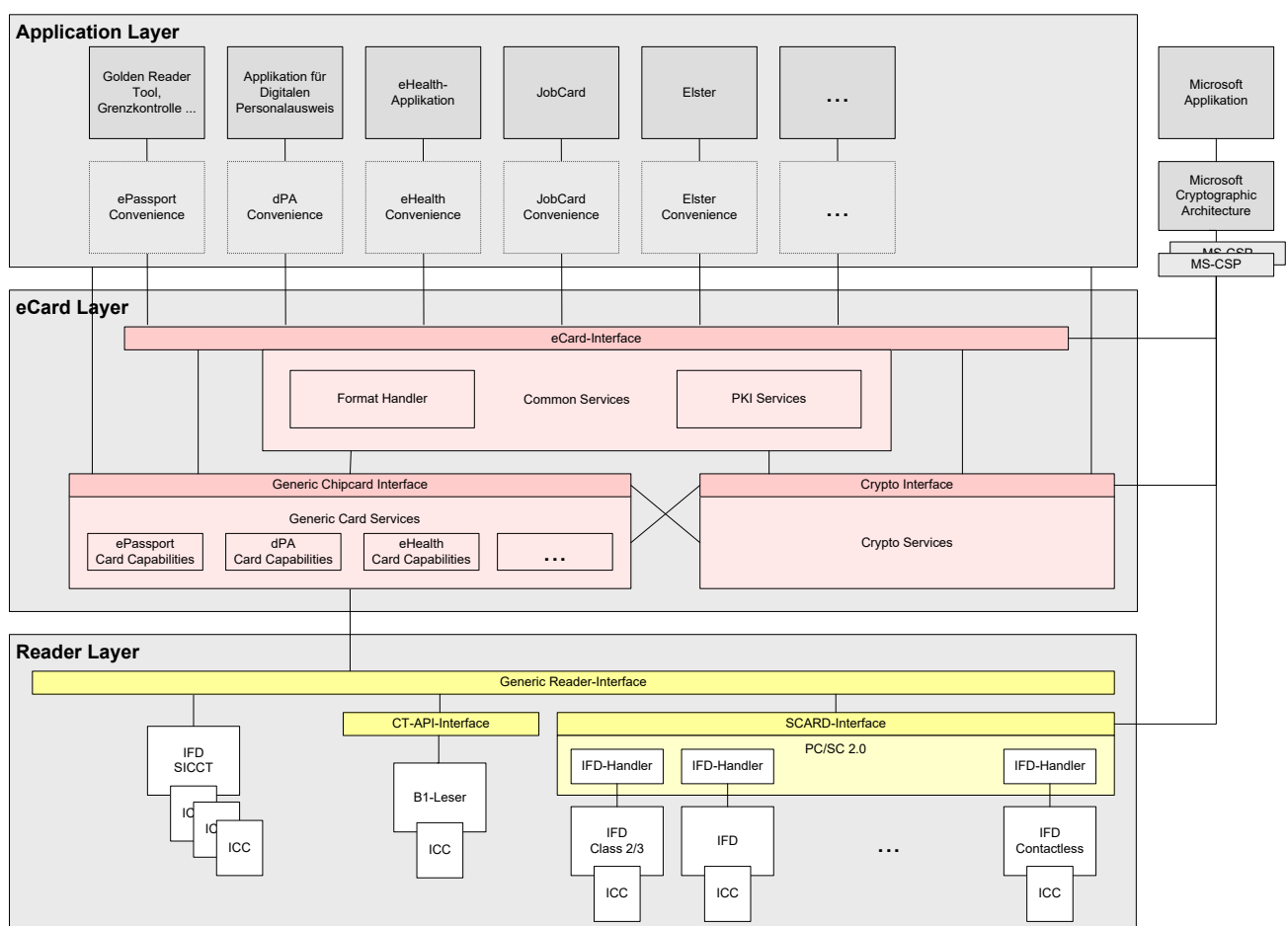


Abbildung 10: eCard-Framework

2 Application Layer

Innerhalb des Application Layer finden sich Anwendungen der unterschiedlichsten Zielsetzungen. Ziel des eCard-Frameworks ist es, innerhalb dieser Anwendungen dennoch eine möglichst homogene Schnittstelle zu den Kartendiensten anzubieten, so dass eine Anwendung möglichst wenig oder kein Spezialwissen über die anzusprechende Chipkarte besitzen muss.

Ein weiteres Ziel ist, keine Interferenzen zu bereits bestehenden etablierten Standards, wie z.B. dem Microsoft Crypto-API-Konzept, zu generieren. Das konkrete Konzept sieht sogar eine Verbindung dieser unterschiedlichen Welten vor, so dass es z.B. problemlos möglich ist, einen Crypto Service Provider auf die herkömmliche Art oder unter Benutzung des eCard-Framework zu realisieren. Selbst ein Parallelbetrieb beider Konzepte ist durch die gewählte Kartenkommunikation möglich.

3 eCard Layer

Etabliert man ein Framework mit einer möglichst generischen Schnittstelle zur Applikations-Ebene und einer möglichst generischen Schnittstelle zur Chipkarten-Ebene, so muss dennoch an einer Stelle das Spezialwissen über die verwendete Chipkarten-Art und deren Personalisierung vorhanden sein. Diese applikationsspezifischen Besonderheiten sind im eCard Layer realisiert. Die existierenden Signaturkarten besitzen eine andere Personalisierung als die elektronische Gesundheitskarte und der Heilberufsausweis oder der elektronische Reisepass. Darüber hinaus wird auch der digitale Personalausweis eine andere Personalisierung aufweisen. Außerdem besitzen die Signaturkarten der verschiedenen Herausgeber derzeit meist unterschiedliche Personalisierungen.

Im eCard Layer sollen zumindest folgende Kartentypen unterstützt werden:

- eGK
- HBA / SMC
- ePass
- dPA (European Citizen Card)
- (beliebige) Signaturkarten

Um die generischen Anfragen der Applikationsebene auf die spezielle Personalisierung einer Karte zu wandeln, werden für jeden Karten- oder Anwendungstyp spezielle Services implementiert. Ziel der Implementation ist es dabei, zum einen ein einheitliches Interface für gemeinsam nutzbare Funktionalitäten wie Signatur, Verschlüsselung etc. zur Verfügung zu stellen, auf der anderen Seite aber ebenfalls eine Erweiterbarkeit zu ermöglichen, um spezielle Funktionalitäten der einzelnen Applikationen (e-Passport, eGK, dPA) zu unterstützen. Unterstützt werden diese Services durch zwei weitere Module, die kryptographische Algorithmen (Crypto Services) und sonstige Sicherheitsdienste, wie OCSP-Anfragen, Zeitstempel, etc., übernehmen (Common Services). Die Crypto Services sollten folgende Funktionen umfassen:

- symmetrische Verschlüsselungsalgorithmen
- asymmetrische Verschlüsselungsalgorithmen
- Hash-Algorithmen
- MAC-Algorithmen
- Grundlegendes x.509-Zertifikats-Handling
- Digitale Signaturen
- Schlüsselaustausch
- Zufallszahlengenerierung

4 Reader Layer

Die generelle Karten- und Kartenlesersteuerung soll über eine einheitliche Schnittstelle bereitgestellt werden, durch die von der konkreten Ausprägung des Lesegerätes abstrahiert wird. Hierbei sollen unterschiedlichste Lesegeräte (kontaktbehaftete Leser (Klasse 1, Klasse 2/3), SICCT-Leser, kontaktlose Leser etc.) unterstützt werden, so dass im eCard-Framework beliebige ISO 7816-kompatible Chipkarten ohne Spezialwissen über Übertragungsprotokolle oder Chipkartenleser genutzt werden können. Auch die kontaktlosen ISO-14443 Protokolle sowie reine Speicherkarten werden komplett durch die API gekapselt. Eine Applikation kann nach wenigen Initialisierungsbefehlen beispielsweise sofort APDUs über beliebige Kartenleser an ebenfalls beliebige ISO 7816-kompatible Chipkarten senden. Spezielle Eigenschaften wie Netzwerkfähigkeit, Multislot-Eigenschaften und die Ansteuerung von Displays, Eingabefeldern und biometrischen Sensoren, wie sie sich beispielsweise im SICCT-Umfeld wiederfinden, müssen ebenfalls unterstützt werden. Aus diesem Grund wird zwischen dem hier beschriebenen Reader-Layer und dem darüber liegenden eCard-Layer eine weitere generische Schnittstelle etabliert, um den Zugriff des eCard-Frameworks auf Kartenleser aller Typen zu vereinheitlichen, ohne jedoch Einschnitte bei der unterstützten Funktionalität hinnehmen zu müssen.

5 Umsetzung

Das eCard-Framework sollte für verschiedene Entwicklungs- und Betriebssystemplattformen zur Verfügung stehen. Darüber hinaus ist eine Netzwerkfähigkeit des eCard-Frameworks wünschenswert, so dass unterschiedliche Komponenten des Frameworks auf verschiedenen Rechnersystemen verteilt ablaufen können. Bereits beim Design des eCard-Frameworks wurde auf die Evaluierbarkeit und Zertifizierungsfähigkeit der zugehörigen Implementierungen gemäß Common Criteria geachtet. Hierbei sind insbesondere die Anforderungen an Produkte für qualifizierte elektronische Signaturen aus dem Signaturgesetz zu beachten. Aufgrund der Vielzahl der zu unterstützenden Systeme wird bei der Erstellung des eCard-Frameworks großer Wert auf Modularität gelegt. Das bezogen auf die jeweilige Anwendung nötige spezifische Wissen wird in separaten Modulen gekapselt.

Bundesverband IT-Sicherheit e.V. (TeleTrust)

Der Bundesverband IT-Sicherheit e.V. (TeleTrust) ist ein Kompetenznetzwerk, das in- und ausländische Mitglieder aus Industrie, Verwaltung, Beratung und Wissenschaft sowie thematisch verwandte Partnerorganisationen umfasst. Durch die breit gefächerte Mitgliedschaft und die Partnerorganisationen verkörpert TeleTrust den größten Kompetenzverbund für IT-Sicherheit in Deutschland und Europa. TeleTrust bietet Foren für Fachleute, organisiert Veranstaltungen bzw. Veranstaltungsbeteiligungen und äußert sich zu aktuellen Fragen der IT-Sicherheit. TeleTrust ist Träger der "TeleTrust European Bridge CA" (EBCA; PKI-Vertrauensverbund), der Personenzertifikate "TeleTrust Information Security Professional" (T.I.S.P.) und "TeleTrust Professional for Secure Software Engineering" (T.P.S.S.E.) sowie der Vertrauenszeichen "IT Security made in Germany" und "IT Security made in EU". TeleTrust ist Mitglied des European Telecommunications Standards Institute (ETSI). Hauptsitz des Verbandes ist Berlin.



Kontakt:

Bundesverband IT-Sicherheit e.V. (TeleTrust)
Dr. Holger Mühlbauer
Chausseestraße 17
10115 Berlin
Telefon: +49 30 4005 4310
E-Mail: info@teletrust.de
<https://www.teletrust.de>



