SICCT
Secure Interoperable ChipCard Terminal

Version: 1.10
Date: 19.12.2006
Editor: Uwe Harasko

VERSIONSHISTORIE

Datum	Version
29.07.2002	0.01
11.08.2005	0.02
12.08.2005	0.03
17.08.2005	0.04
22.08.2005	0.05
26.08.2005	0.06
29.08.2005 31.08.2005	0.07
10.10.2005	0.10
28.10.2005 31.10.2005	0.20
03.11.2005	0.21
04.11.2005	0.30
09.12.2005	0.40
17.01.2006	0.50
30.01.2006	0.50
31.01.2006	0.51
01.02.2006	0.51
02.02.2006	0.51
07.02.2006	0.52
08.02.2006	0.52
13.02.2006	0.52
20.02.2006	0.55
22.02.2006	0.60
27.02.2006	0.90
28.02.2006	1.00
20.04.2006	1.01
08.05.2006	1.02
02.06.2006	1.03
19.12.2006	1.10

HINWEISE ZUM DOKUMENTENSTAND UND HAFTUNGSAUSSCHLUSS BEFINDEN SICH IN KAPITEL 9 !

AUTOREN

Jürgen Atrott	TÜV Informationstechnik GmbH
Thomas Bründl	Omnikey GmbH
Volker Czmok	Celectronic GmbH
Klaus Erichsen	Sagem Orga GmbH
Uwe Harasko	Cherry GmbH
Jörg Kühnl	Cherry GmbH
Klaus Leistner	Celectronic GmbH
Torsten Maykranz	SCM Microsystems GmbH
Frank Neumann	Gemplus mids GmbH
Frank Osthoff	Sagem Orga GmbH
Uwe Schnabel	Omnikey GmbH
Thore Simonides	Sagem Orga GmbH

1 Scope

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. geschlossen 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 erfüllte, 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 Standardplattform 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 Nomen 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 soll 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 Working Group SICCT, einem seit Juli 2005 bestehendenden Industriekonsortiums unter dem Verband des Deutschen TeleTrust e.V. 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 sollen erfüllt werden, so dass diese Geräte für sensitive Anwendungsfälle wie Digitale Signatur- oder Anwendungen im Gesundheitswesen gleichermaßen einsetzbar sind. Für bereits identifizierte Anwendungsbereiche sollen Anhangkapitel, eine Konkretisierung der Terminalausprägung vom Basiskonzept ableiten und fixieren. Die WG SICCT sieht vor, spezifische Ableitungen in Kooperation mit Anwendergruppen zu erarbeiten, und diese als separate Dokumente oder als Anhänge dieser Spezifikation sukzessive veröffentlicht zu sehen. Die erste Ausprägung eines SICCT-Terminals wird ein Vorschlag zur Standardisierung einer Kartenterminalplattform entsprechend den Anforderungen des Deutschen Gesundheitswesens nach Vorgaben der nationalen Vorgabeinstanz gematik zur Einführung einer elektronischen Gesundheitskarte sein [gematik_KT].

Unter Berücksichtigung dieser Spezifikation 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 Geräteressourcen 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 soweit zu vereinfachen, dass eine verbesserte Interoperabilität unter diversen Lieferanten, Modellen und Ausbaustufen erzielt wird, wie es bei anderen standardisierten IT-Komponenten selbstverständlich geworden ist.

2 Einleitung

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

Die Anforderungen ergeben sich aus einer beabsichtigen (partiellen) Konformität und Harmonisierung der Standards

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

Hinsichtlich der funktionalen Betrachtung wird von einem applikationsunabhängigen und multifunktonalen 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.

Besondere Berücksichtigung findet dabei die eCard-Strategie zur Unterstützung der flächendeckenden Einführung von Chipkarten im Bereich der Bundesverwaltung welche das Bundeskabinett am 9. März 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 digitale Personalausweis (dPA)
- Der elektronische Reisepass (ePass)
- Die elektronische Steuererklärung (Elster)
- Das Jobcard-Verfahren

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

3 Systemarchitektur

Das "Secure Interoperable Chip Card Terminal" (SICCT) ist ob 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.

3.1 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.

Im Folgenden werden die Anforderungen aufgelistet.

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 bezieht die Chipkarten-Interfaces (Kontaktiereinheiten) sowie Anschlüsse und Kommunikationsports (z. B. RS-232, USB, Ethernet 802.3) zu anderen IT-Systemen mit ein.
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 bezieht sich im wesentlichen auf die Unterstützung verschiedener Chipkartentypen. Dieses umfasst die Verpflichtung, Prozessor-chipkarten nach den üblichen spezifischen Protokollen betreiben zu können. Die Unterstützung von Speicherchipkarten und sowie von RF-ID-Token ist optional.

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 stellt eine sichere Umgebung für Chipkarten bereit und signalisiert dem Kartenanwender einen erkennbar sicheren und betraubaren Betrieb.
GA_8	Anwenderauthentisierung	Das Kartenterminal bietet technische Funktionen zur Anwenderauthentifizierung (Keypad, biometrische Sensorik, RF/ID Leser). Authentisierungsdaten werden vom Kartenterminal nicht an das Host-System weitergegeben, intern nicht gespeichert und fallbezogen angewendet.
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 vorzunehmen.

3.2 Architekturmodell

Die Systemarchitektur sieht generell vor, 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 kontrolliert und gewährt das Terminal über eine definierte Kommandoschnittstelle. Der Zustand und die Verfügbarkeit von Ressourcen wird durch das Kartenterminal intern überwacht, kontrolliert und kann in Form von Zustandsvariablen an die externe Welt gemeldet werden. Einer

bestimmten Anzahl diskreter Kartenterminal- und Kartenzustände werden hierfür einheitliche Werte bzw. Statuscodes zugeordnet.

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

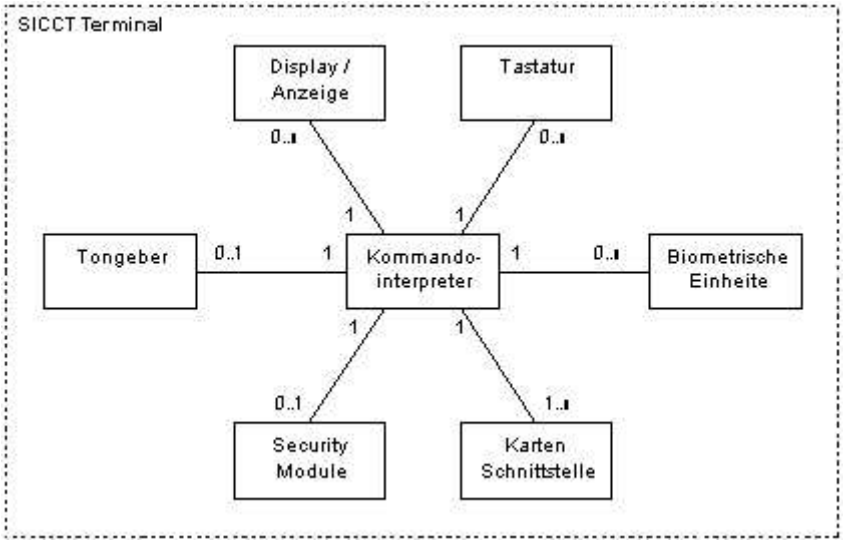


Abbildung 1: Modell SICCT Terminal

3.3 Kommunikationsmodell

Das SICCT-Kommunikationsmodell stellt aus Sicht des Netzwerks ein Client-Server-Kommunikationsmodell dar, wobei das Kartenterminal in der Rolle des Servers Dienste anbietet. 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 geschieht ausschließlich über die verfügbaren Basisdienste. 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 internen 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 gehaltenen 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, sollen keine Vorgaben oder Restriktionen bestehen, 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.

3.4 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 folgende Zustände

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

Folgende Vorgaben soll für alle SICCT- Terminals verpflichtend gelten.

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 sollen 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 beinhalten einen oder mehrere Betriebsmodi welche jeweils einem Kommandointerpreter oder Kommandosatz an Basisfunktionen zugeordnet sind.	Kommandosatz
APP_5	▪ SICCT-Kartenterminals bieten zur Ansteuerung des Kartenterminals sowie der Chipkartenslots eine Kommandoschnittstelle an	Kommandoschnittstelle
APP_6	▪ Der SICCT-Kommandosatz basiert auf dem CT-BCS Befehlsatz der MKT und der PC/SC Spezifikationen [MKT]. Die Struktur der Kommandos folgt derjenigen für ISO 7816-4 APDUs.	Kommandostruktur
APP_7	▪ SICCT-Kartenterminals unterhalten eine Schnittstelle zur zentralen Administration des Terminals.	Administrationsschnittstelle
APP_8	▪ Spezifische SICCT-Terminals für gleiche Anwendungsdomänen sollen eine "Drop in"-Ersetzbarkeit bei Produkten unterschiedlicher Herstellern aufweisen.	Interoperabilität

Es ist offensichtlich, das eine konkrete Ausprägung eines auf SICCT-basierten Kartenterminals für eine oder mehrere Anwendungsdomänen nicht universell vorgegeben werden kann, 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:

Optionale Vorgaben an 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 optional 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 ▪ RF/ID 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 der Kommunikationspfades
OPT_6	▪ Unterstützung eines als Komfortsignatur benannten Mechanismus	Komfortsignatur

3.5 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 vom eingegeben wurden, müssen der externen Welt verborgen bleiben und dürfen nur gerichtet an eine Chipkarte gesendet werden. Zu schützende Authentifikationsdaten erkennt das Kartenterminal aus einem korrespondierenden Kommandokontext, 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.

3.6 Benutzerrollen und CT Session

Ein Zugriff auf das Kartenterminal erfolgt stets auf Basis einer Benutzerrolle im Rahmen einer Kartenterminalsitzung (CT Session). Das SICCT Kartenterminalkonzept unterscheidet folgende Akteure entsprechend ihrer Zugriffsmöglichkeiten auf Funktionen und Betriebsparameter des Kartenterminals in drei Benutzerrollen.

Role	Access	Function
Kartenterminal-administrator (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 Kartenterminal-kommandos,
Kartenterminal-steuerung (CT Control)	Unprivilegierter und beschränkter Zugriff ohne Authentifizier-ung durch das Kartenterminal (anonymous control)	Programmatische Steuerung des Kartenterminalbetriebs: ▪ Keine direkte Modifikation von Betriebsparametern, ▪ Abfrage bzw. Anzeige bestimmter Betriebsparameter, ▪ Eingeschränkte Nutzung bestimmter Kartenterminal-kommandos,

Karten- bzw. Kartenterminal-anwender (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 (programmatischer) Zugriff,

Ein steuernder (programmatischer) Zugriff auf das Kartenterminal erfolgt durch die Benutzerrolle
- CT Admin oder
- CT Control.

CT Session	Autorisation	Authentification	Session Handling
CT ADMIN Session	CT Admin Rolle	Explizite Logininformation ▪ Benutzername(Username) ▪ Benutzeridentifikation (Useridentifikation)	▪ Explizites Öffnen und Schließen einer CT Session.
CT CONTROL Session	CT Control Role	Implizite Logininformation ▪ Benutzername(Username) 'Anonymous Control' ▪ Keine Benutzeridentifikation	▪ Explizites Öffnen und Schließen einer CT Session.
-	CC / CT User Role	-	▪ Bedienerzugriff im Rahmen einer bestehenden CT Session. ▪ CC / CT User kann keine CT Session aufbauen.

4 Beschreibung der Schnittstelle zur Karte

Maßgeblichkeit der zugrunde liegenden Standards

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

- ISO7816: [STD2] bis [STD9]
- EMV: [EMV_41]

Die ISO-Norm, hier speziell ISO7816-3, 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 (ähnlich der ISO-Norm) 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-Norm (direkt oder indirekt über die Anforderungen an die Karte), so ist diese maßgeblich.
3. Widersprechen sich die Anforderungen der beiden Normen, so greift die aktuelle Terminaleinstellung "ISO" oder "EMV".

Der Anhang A als normativer Teil der SICCT-Spezifikation ist eine detaillierte Gegenüberstellung der Anforderungen aus den ISO- und EMV-Normen (letztere 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 Norm 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 der 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.

Der Auslieferungszustand ist die Einstellung "ISO".

Als kontaktlose Karten werden nur "Proximity Cards" berücksichtigt, die sich nach der Norm ISO 14443 [STD10] bis [STD15] richten.

4.1 Mechanische Anforderungen

4.1.1 Geeignete kontaktbehaftete Kartentypen

Das Terminal ist vorgesehen zur Aufnahme von einer oder von mehreren kontaktbehafteten Karten der Größe ID-1.

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

Für alle diese Karten gelten die Normen [STD1], [STD2], [STD3] und [STD4].

4.1.2 Anforderungen an die Kartenkontaktierungen für Größe ID-1

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

Für die Karten ID-1 wird eine hohe Zuverlässigkeit und Funktionssicherheit auch bei hoher Steckzyklenzahl für Kontaktierung und Karte verlangt. Hierfür ist zu gewährleisten:

- 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. Auswurftaste
- 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 ist die Möglichkeit einer Notentnahme der Karte bei Fehlfunktionen des Terminals oder bei Stromausfall sicherzustellen.

4.1.3 Anforderungen an die Kartenkontaktierungen für Größe ID-000

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

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

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 Interruptgesteuerte 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 (Proximity Cards nach [STD10]) 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 Spezifikationen

- ISO/IEC 7816-3 [STD6] in Verbindung mit Amendment 1 [STD7]
- EMV Version 4.1 [EMV_41]

Die Vorgaben für das Verhalten eines SICCT-Terminals sind in Abschnitt 4.0 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 fordert die EMV-Norm in Kapitel 5.5.3, dass C6 nicht angeschlossen werden soll. Dies gilt auch für ein SICCT-Terminal. Sollte das Terminal optional den Kontakt C6 ansteuern, so ist die Aktivierungssequenz aus Kapitel 5.2 [STD6] einzuhalten.

Anforderungen an Kontakte C4 und C8: Diese Kontakte brauchen laut EMV-Norm nicht vorhanden zu sein. Dies gilt auch für ein SICCT-Terminal. Sollte das Terminal optional USB-Karten nach ISO7816-12 und/oder synchrone Karten vom Typ 2 nach ISO7816-10 unterstützen, muss C4 und evtl. C8 entsprechend ansteuerbar sein. In diesem Fall sollen 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 sind alle vorhandenen Karten unabhängig voneinander ansteuerbar. 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 Standards ISO 7816 und EMV sind in Abschnitt 4.0 und in der Tabelle Anhang A detailliert aufgeführt.

Ergänzungen und Erläuterungen:

Im Rahmen der ATR-Prozedur wird ergänzend zu [EMV_41] die Versorgungsspannung für die folgende Chipkarten-Session vereinbart. Dazu ist das in Kapitel 4.2.2 von [STD7] definierte Verfahren anzuwenden. In Ergänzung zu [STD7] ist folgende Reihenfolge der Aktivierungen einzuhalten: zunächst 5V, dann 3V, optional schließlich 1,8V.

Als abschließenden Teil der Einschaltprozedur führt das Terminal ergänzend zu [EMV_41] das PPS-Verfahren (Protocol Parameter Selection) durch. Das Terminal erfüllt folgende Mindestanforderungen:

- 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 Kapitel 6.5.2 von [STD6].

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 Kapitel 5.3.4 [STD6] wird ergänzend zu [EMV_41] unterstützt.

4.2.3 Unterstützung von synchronen Karten

Zusätzlich zu asynchronen Prozessorkarten können auch Speicherkarten mit synchroner Datenübertragung unterstützt werden [STD9]. Die Verfahren und Protokolle sind der MKT-Spezifikation Teil 2, Abschnitt 7 zu entnehmen [MKT1].

Die Kartenaktivierung ist bei Unterstützung synchroner Karten wie folgt zu erweitern: Sendet die Karte beim Kalt-Reset keine asynchronen Antwortzeichen, wird die Karte deaktiviert und das ATR-Verfahren nach [STD9] zur Erkennung einer synchronen Karte angewendet.

Die optionale Unterstützung synchroner Karten kann 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 unterstützt, so richten sich die Terminal-Anforderungen nach ISO 14443 Typ A und Typ B (Proximity Cards), siehe [STD11] bis [STD14].

Darüber hinausgehend 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. Weitere Protokolle können optional unterstützt werden.

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

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 [STD9], Teil 6 von [MKT1] und [IND3]
- S=11 FCBP, siehe Typ 2 in [STD9]

4.3.2 Protokolle für kontaktlose Karten

Werden kontaktlose Karten unterstützt, gilt das Protokoll T=CL gemäß [STD15]. Weitere Protokolle sind Option.

Darüber hinausgehend 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 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 kann prinzipiell über diverse Bussysteme (z.B. RS-232, USB, Ethernet 802.3, ...) geschehen und ist transparent und unabhängig zu den Kommandostrukturen und Inhalten gestaltet. Das Kapitel 6 beschreibt die Abläufe sowie das zugrundegelegte netzwerkbasierte Protokoll.

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

Der Umfang des SICCT-Kommandosatzes (SICCT-CS) definiert das Verhalten des Kommandointerpreters. Jede erfolgreiche Abarbeitung eines Kommandos erzeugt einen definierten Gerätezustand oder Betriebsmodus. 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 folgt der bytesequenziellen Kommandostruktur für Chipkarten als sog. APDU (Application Protocol Data Unit, Byte-Folge) nach ISO-7816-4 [STD8], bzw. dem 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 (s.6.3.4, Ereignisbenachrichtigung).

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

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

<table>
<tr><td colspan="11"><C-APDU></td></tr>
<tr><td colspan="11">Length: 4 ... ((0..3) + Lc + (0..3))</td></tr>
<tr><td colspan="4">Header
mandatory</td><td colspan="7">Body
[optional]</td></tr>
<tr><td colspan="4">Length: 4</td><td colspan="7">Length: 0 ... ((0 ... 3) + Lc + (0 ... 3))</td></tr>
<tr><td rowspan="6">1</td><td rowspan="6">2</td><td rowspan="6">3</td><td rowspan="6">4</td><td colspan="7">Coding Rules for Nc and Ne
in accordance to ISO 7816-4</td></tr>
<tr><td colspan="3">Max. Value indicated by Lc:
Nc < = 65535</td><td colspan="4">Max. Value indicated by Le
Ne < = 65536</td></tr>
<tr><td>Lc absent</td><td>empty</td><td>Nc = 0</td><td>Le absent</td><td>empty</td><td colspan="2">Ne = 0</td></tr>
<tr><td>Lc short[1]</td><td>1 Byte</td><td>'01' <= Lc <= 'FF'
1 <= Nc <= 255</td><td>Le short</td><td>1 Byte</td><td colspan="2">'01' <= Le <= 'FF'
1 <= Ne <= 255

Le = '00'
Ne = 256</td></tr>
<tr><td rowspan="2">Lc extended[2]</td><td rowspan="2">3 Byte</td><td rowspan="2">'000001 <= Lc <= '00FFFF'
1 <= Nc <= 65535</td><td rowspan="2">Le extendend</td><td>3 Byte</td><td colspan="2">Condition: Lc absent

'000001' <= Le <= '00FFFF'
1 <= Ne <= 65535

Le = '000000'
Ne = 65536</td></tr>
<tr><td>2 Byte</td><td colspan="2">Condition: Lc extended

'0001' <= Le <= 'FFFF'
1 <= Ne <= 65535

Le = '0000'
Ne = 65536</td></tr>
<tr><td>CLA</td><td>INS</td><td>P1</td><td>P2</td><td>[Lc]</td><td colspan="3">[Data Field]
<Nc Bytes of Data></td><td colspan="3">[Le]
< requesting Ne Bytes of response Data ></td></tr>
</table>

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 Body sind optional und können ein Datenfeld und / oder 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 klassifiziert ein Kommando durch einen eindeutigen Wert. Unterschiedliche Kommandosätze sind einer Klasse zugeordnet.

[1] According ISO 7816-4 [STD8] either both Le and Lc are short or extended.

[2] According ISO 7816-4 [STD8] either both Le and Lc are short or extended.

5.2.2 INSTRUCTION - Byte

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

5.2.3 Parameter P1

Der Parameter P1 kodiert einen Ausführungskontext für einen Befehl am Kartenterminal. 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 folgen den Kodierungsregeln nach ISO 7816-4 und unterstützen optional das Extended Length Format.

Üblicherweise besitzen Lc und Le eine Länge von einem Byte. 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 filed) 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, wieviele Datenbytes in der Antwort (R-APDU) maximal zurückgegeben werden sollen.

- Sonderfall 1 - Die Anzahl zurückgegebener Daten unterschreitet oder ist gleich dem (erwarteten) Wert von Le. Das bedeutet, dass alle verfügbaren Daten, zurückgeliefert wurden. Der korrespondierende Statuscode (SW1SW2, siehe 5.4.1) für SICCT-Kommandos ist dann '9000'.

- Sonderfall 2 - Wird der Wert für Le auf '00' bzw. '000000' gesetzt, bedeutet dieses alle verfügbaren Daten, maximal 256 bzw. 655536 Bytes, zurückzuliefern. Der korrespondierende Statuscode (SW1SW2, siehe 5.4.1) für SICCT-Kommandos ist dann '9000'.

- Sonderfall 3 - Existieren Le und Lc müssen diese gleich kodiert entweder in short oder extended Form vorliegen.

- Sonderfall 4 - Le wird in nur zwei-Byte <HB | LB> kodiert, wenn Lc in extended (Drei-Byte-Darstellung) vorliegt.

5.3.2 Command Data Field

Das Command Data Field kann beliebig strukturierte Befehlsdaten beinhalten. Der Inhalt und die Datenstruktur sind befehlsspezifisch. 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 wird von SICCT generell in Form eines R-APDUs zurückgegeben. 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 wird stets auf ein C-APDU zurückgesendet. Der Trailer beinhaltet ein Zwei-Byte-Statuswort (SW1-SW2), dessen Kodierung in Anlehnung an ISO 7816-4 erfolgt.
Das erste Byte SW1 kodiert stets den Ausgang einer Operation.

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 - 1st 'Checking Error', 2nd 'Warning' and 'Normal Processing' , '3rd Execution Error'				

Das Byte SW2 spezifiziert den Ausgang einer Operation genauer. Der Wert ist befehlsspezifisch.

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 (Objetcs) given within command.	Zuviele oder zuwenig Daten (Objekte) im Kommanod 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 Response Body

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

5.5 Command Set Overview and General Return Codes

Ein SICCT Kartenterminal verfügt über einen zentralen Kommandointerpreter und wird über einen Kommandosatz gesteuert. 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 ist einem Betriebsmodus zugeordnet. Über die Selektion eines Betriebsmodus kann der Kommandosatz eingestellt bzw. gewechselt werden. Es ist ein Betriebsmodus zur Zeit aktiviert.

5.5.1 Adressierung und Transport von Kommandos

Am Transportendpunkt des Kartenterminals befindet sich der Kommandointerpreter, der genereller Empfänger aller Kartenterminalkommandos ist.

Kartenterminal- wie Chipkartenkommandos werden auf APDU-Level neben einer eindeutigen Sequenznummer, die der Absender des Kommandos vorgibt, zur Ausführung an den Kommandointerpreter übergeben. Das Kartenterminal antwortet nach Ausführung des Kommandos (Response) unter Angabe der Sequenznummer.

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 Messagetypeabhängigen Datenteils.

5.5.2 Ausführung und Wirkweise von Kommandos

Der SICCT Kommandosatz ermöglicht primär die Aktivierung, Kommunikation und Deaktivierung mit Chipkarten. Zusätzlich sind Kommandos zum Management des Kartenterminals sowie zur Realisierung einer Benutzerführung am Kartenterminal vorhanden.
Jedes Kommando adressiert Ressourcen des Kartenterminals und realisiert primär eine Hauptaktion (wie z.B. Aktivierung, Deaktivierung, Displayausgabe, Keypad-Eingabe). Über optionale Kommandoqualifizierer erlauben einige Kommandos die Kombination mit einer oder mehreren Zusatzaktionen wie Statusanzeige über die Mensch-Maschine-Schnittstelle, Benutzerdialoge oder mechanischer Auswurf einer Chipkarte.

Vor dem Hintergrund gesehen, kann man sagen, dass die Ausführung der hauptsächlichen Kommandos in einzelnen Phasen abläuft:

Ausführungsphasen der SICCT Kommandos				
Phase		**Stage**	**option**	**Description**
preprocessing phase	Vorbereitungsphase	1	✓	z.B. ▪ Anzeige eines Benutzerdialogs, ▪ Eingabeaufforderung ▪ Chipkartenanforderung
processing phase	Ausführungsphase	2		z.B. ▪ Ausführen der Hauptaktion, z.B. Aktiveren einer Chipkarte nach ISO 7816
postprocessing phase	Nachbereitungsphase	3	✓	z.B. ▪ Benutzerdialoganzeige ▪ Chipkatenentnahme ▪ Chipkartenauswurf

Die Mächtigkeit und Wirkweise einzelner Kommandos richtet sich nach der Art des Kommandos, sowie den technischen Möglichkeiten und verfügbaren Ressourcen der jeweiligen Terminalplattform (LED, Display, Keypad, interner Timer, RFID-Antenne etc.).

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 TERMINATE COMMANDs unter Angabe der Kommandosequenznummer abgebrochen werden, sofern ein bestimmter Kommandotyp den Kommandoabbruch unterstützt.

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

5.5.4 Nebenläufige Kommandobearbeitung

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

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

5.5.5 Betriebsmodi

Ein SICCT Terminal kann einen oder mehre Betriebsmodi bieten, von denen immer einer als 'Aktiver Betriebsmodus' für das gesamte Gerät selektiert ist.

Ein erster optionaler Betriebsmodus 'BCS' basiert auf dem Basisbefehlsatz der MKT CT-BCS Spezifikation [MKT_10]. Dieser Modus bewegt sich ausschließlich im Rahmen Möglichkeiten der MKT -Spezifikation und erlaubt die Verarbeitung kontaktbehafteter synchroner und asynchroner Chipkarten.

Ein weiterer Betriebsmodus 'SICCT' soll eine erweiterte Funktionalität oberhalb der das BCS-Modes abbilden. Die korrespondierenden Kartenterminalkommandos werden in Form neu definierter SICCT-Kommandos festgelegt. Der Modus 'SICCT' steuert und kontrolliert den Zustand kontaktbehafteter und (optional) kontaktloser Kartenkontaktiereinheiten (asynchrone und optional synchrone Chipkarten) und bietet weitere Funktionen hinsichtlich einer komfortableren Terminalansteuerung.

Zwischen den Modi kann je nach Fähigkeit und Kenntnis einer steuernden Entität (HOST) gewählt und gewechselt werden. Per Gerätekonfiguration wird ein Betriebsmodus als

'Standard-Betriebsmodus' aktiviert. Diesen Modus wählt das Gerät automatisch nach einem Reset des Kartenterminals.

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 kompatibel' steuert und kontrolliert über den BCS-Kommandosatz ausschließlich den Zustand kontaktbehafteter Chipkarten-Kontaktiereinheiten für synchrone und asynchrone Chipkarten nach ISO 7816.

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

Zusätzlich muss dieser Modus das Kommando unterstützen
- SICCT SELECT CT MODE (s. 5.6).

Dieser Betriebsmodus unterstützt generell keine nebenläufige Kommandobearbeitung und keinen Kommandoabbruch.

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

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 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.

Table : Overview of the SICCT supported Basic Command Set

5.5.7 Betriebsmodus 'SICCT' und Command Set

Der Mode 'SICCT kompatibel" steuert den Zustand kontaktbehafteter und kontaktloser Kartenkontaktiereinheiten und ermöglicht den Betrieb synchroner wie asynchroner Chipkarten nach ISO 7816 und 14443. 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 basiert auf einer expliziten Anwahl durch die Anwendungsumgebung. Diese geschieht entweder durch eine statische Konfiguration am Terminal (gewählter 'Standard-Betriebsmodus') oder durch eine Selektion über das SICCT-Kommando SELECT CT MODE. Damit verfügt die Anwendungsumgebung über die Kenntnis, dass dieser spezielle Modus selektiert wurde, und kann von allen Funktionserweiterungen Gebrauch machen.

Die Struktur der SICCT-Terminalkommandos entspricht derjenigen für BCS-Kommandos. 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.

CardTerminal Basic Command	**CLA Code (hex)**	**INS Code (hex)**	**P1**	**P2**	**Lc**	**Cmd Data**	**Le**	**Brief Description**	
SICCT RESET CT / ICC	80	11	FU	CQ	opt	opt	opt	man	Perform Cold / Warm Reset and optional PPS. Reset of the ▪ cardterminal device ▪ RF antenna (option) ▪ Chipcards
SICCT REQUEST ICC	80	12	FU	CQ	opt	opt	opt	man	Request for chipcard presentation and monitoring for time period.
SICCT GET STATUS	80	13	FU	CQ	opt	opt	opt	Man	Request parameter for the ▪ Cardterminal ▪ functional units ▪ ChipCard
SICCT EJECT ICC	80	15	FU	CQ	opt	opt	opt	man	Disable chipcard logically and / or electrically. Optionallly: mechanical operation.
SICCT INPUT	80	16	FU	CQ	opt	opt	opt	man	Query input data by the user interface functions of the cardterminal ▪ Keypad ▪ (Biometrical) Sensors

CardTerminal Basic Command	CLA Code (hex)	INS Code (hex)	P1	P2	Lc	Cmd Data	Le	Brief Description	
SICCT OUTPUT	80	17	FU	CQ	opt	opt	opt	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	opt	opt	opt	Man	Process Card Holder Verification. Perform a password / PIN entry operation, build a chipcard command and perform verification by an addressed chipcard.
SICCT MODIFY VERIFICATION DATA	80	19	FU	CQ	opt	opt	opt	man	Perform modify operation of Card Holder Verification Data ▪ Password / PIN
SICCT SELECT CT MODE	80	20	FU	CQ	opt	opt	opt	man	Select Operation / Command Set Mode ▪ BCS Mode ▪ SICCT Mode
SICCT COMFORT_ AUTHENTICATION	80	21	FU	CQ	opt	opt	-	opt	Support command for authentication. Learns and stores an authentication dataset with ▪ authentization 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 ▪ authentization data and ▪ Serialnumber (ICCSN).
SICCT SET STATUS	80	23	FU	CQ	opt	opt	opt	man	Set parameter (Data Object) for the ▪ cardterminal ▪ functional units. ▪ chipcards.
SICCT DOWNLOAD INIT	80	24	FU	CQ				man	Start of FW download
SICCT DOWNLOAD DATA	80	25	FU	CQ	man	man	opt	man	Data transportation during FW download
SICCT DOWNLOAD FINISH	80	26	FU	CQ				man	Manifestation and completion of a FW download
SICCT TERMINATE COMMAND	80	27	FU	CQ	man	man		man	Abort or termination of SICCT command.
SICCT INIT CT SESSION	80	28	FU	CQ	man	man	man	man	Init and open a Cardterminal .
SICCT CLOSE CT SSESSION	80	29	FU	CQ	man	man	man	man	Close a Cardterminal Session.
FU = Functional Unit									
CQ = Command Qualifier									
opt = optional									
man = mandatory									

Tabelle : Overview of the SICCT Command Set

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

Return Code		Return of ...	Meaning

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

Tabelle: CardTerminal Return Codes SW1-SW2 for successful operation

5.5.8 Unterstützte Chipkarten

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

Chipcard Type	Normative Reference(s)	Chipcard Protocol		BCS Mode	SICCT Mode	Remark
Asynchronous Smartcard	ISO7816-3	T=0	T0CP	✓	✓	Microprocessor Chipcard with block transmission protocol T=1
		T=1	T1BP	✓	✓	Microprocessor Chipcard with character transmission protocol T=0
Synchronous Chipcard	ISO7816-10 Type 1	S=8	SDAP	✓	opt	Memory Chipcard with I²C-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	ISO14443-4	T=CL	T=CL		opt	Proximity Chipcard with block transmission protocol

Tabelle: Unterstützte Chipkarten und Protokolle

5.5.8.1 Kommandos für synchrone Chipkarten

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

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

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		for chipcards with Verification Data (security function)	Vergleich von Verifikationsdaten mit den in der Karten gespeicherten Verifikationsdaten.
CHANGE REFERENCE DATA	ISO 7816-8		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 aufgezeigten Wertebereichs angesprochen werden. Je nach technischer Ausprägung und Betriebsmode des SICCTs stehen unterschiedliche Funktionseinheiten zur Verfügung.

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.

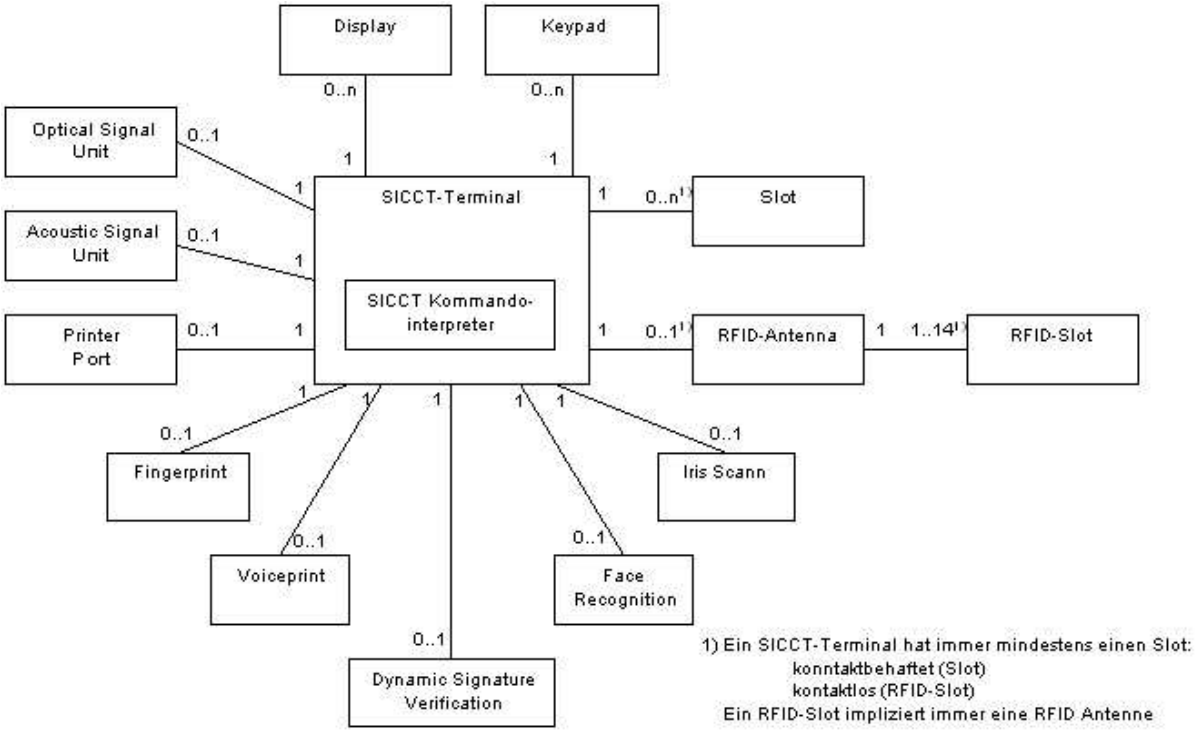


Abbildung 2: SICCT Functional Units

5.5.9.2 Ressourcentabelle

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

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

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

Tabelle 1: 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. s. 5.5.10.9) erscheinen kann, wenn eine Referenced Coding Darstellung von P1 erfolgt."
Der Wertebereich des Indexwerts ist abhängig von der maximalen Anzahl der Functional Units, und darf ein Byte ('00' bis 'FF') umfassen.

Friendly Names stellen ASCII-kodierte Zeichenketten mit einer maximal Länge von 32 Zeichen dar.

5.5.9.3 Direct Coding

Folgende Tabelle zeigt die verfügbaren Werte in der Direct Coding-Darstellung für den Parameter P1. Einige Werte erscheinen auch im Parameter P2, 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	1st contact bound ChipCard Interface Unit
:	:	**:**			:	:
ICC14	ICC-Interface 14	**'0E'**			opt	14th ChipCard Interface Unit
Address RFID / Contactless ChipCard Interface						
RFID	RFID Antenna Unit	**'10'**	-	✓	opt	Address RFID Antenna Unit with all recognized RFID Tokens.
RFIDn	RFID Token n	**'1x'**	-	✓	opt	Address single RFID Token.
RFID1	RFID Token 1	**'11'**			opt	1st RFID Token at Antenna Unit
:	:	**:**			:	:
RFID14	RFID Token 14	**'1E'**			opt	14th RFID Token at Antenna Unit
Address Human Interface Device Unit						
CT-Display (0)	Standard Display	**'40'**	✓	✓	opt	1st 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	1st 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	Printerport	**'60'**	✓	✓	opt	Cardterminal controlled Printerport
CT RTC	Real Time Clock	**'61'**		✓	opt	Internal Real Time Clock
Address Human Interface Device Unit of Type 'Biometric Sensor'						
Biometrical Unit		'7x'	✓	✓	opt	2nd 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 Verification Unit	**'72'**			opt	Cardterminal controlled Dynamic Signature Verification 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 2: 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 ist wie folgt möglich: 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 den 1. RFID Slot:

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

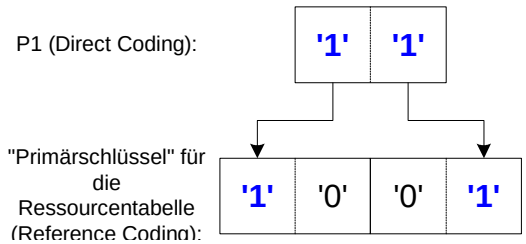


Abbildung 3: Beispielabbildung für Direct Coding auf Ressourcentabelle

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

Functional Units (Referenced Coding Representation)					
Functional Unit	**Meaning**	**Referenced Coding** (2 Byte)	**Mode** **SICCT**		**Description**
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	1st contact bound ChipCard Interface Unit
:	:	**:**		opt	:
ICC255	ICC-Interface 255	**'00FF'**		opt	14th ChipCard Interface Unit
Address RFID / Contactless ChipCard Interface					
RFID	RFID Antenna Unit	**'1000'**	✓	opt	Address RFID Antenna Unit with all recognized RFID Tokens.
RFIDn	RFID Token n	**'100x'**	✓	opt	Address single RFID Token.
RFID1	RFID Token 1	**'1001'**		opt	1st RFID Token at Antenna Unit
:	:	**:**		opt	:
RFID14	RFID Token 14	**'100E'**		opt	14th RFID Token at Antenna Unit
Address Human Interface Device Unit					

Functional Units (Referenced Coding Representation)					
Functional Unit	**Meaning**	**Referenced Coding** (2 Byte)	**Mode**		**Description**
			SICCT		
CT-Display (0)	Standard Display	**'4000'**	✓	opt	1st 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	1st 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	Printerport	**'6000'**	✓	opt	Cardterminal controlled Printerport
Address Human Interface Device Unit of Type 'Biometric Sensor'					
Biometrical Unit		'700x'	✓	opt	2nd 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' - 'F0FE'	✓	opt	Vendor defined Functional Unit

Tabelle 3: Functional Units - Referenced Coding Representation

Folgende Tabelle zeigt einige Beispiele für Referenced Coded Values.

Referenced Coded Value	FU Type	FU-Index-Value	Friendly Name	Remark
0000	00	00	"Terminal"	Cardterminal
0001	00	01	"ICC SLOT 1"	1st 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

Referenced Coded Value	FU Type	FU-Index-Value	Friendly Name	Remark
:	:	:	:	:
00FF	00	FF	"ICC SLOT 255"	Contact bound ChipCard Slot 255
:	:	:	:	:
1000	10	00	"RFID Antenna"	RFID Antenna
1001	10	01	"RFID SLOT 1"	1st RFID Slot
:	:	:	:	:
100E	10	0E	"RFID SLOT 14"	14th RFID Slot
4000	40	00	"Standard Display"	1st Display
4001	40	01	"Display 1"	Additional Display 1
:	:	:	:	:
5000	50	00	"Standard Keypad"	1st Keypad
5001	50	01	"Keypad 1"	Additional Keypad 1
:	:	:	:	:
6000	60	00	"Printer Port"	One (optional) Printer Port
6001	60	01	“Real Time clock”	One (optional) RTC
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	"1st SICCT Generic FU"	
:	:	:	:	:
A000	A0	00	"1st Vendor defined generic FU"	

Tabelle 4: Typkennzeichnung von Functional Units in der Ressourcentabelle

5.5.10 Data Objects

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

Das Design der Daten Objekte folgt dem Prinzip der MKT-Spezifikation, mit dem Ziel, die im Basic Command Set enthaltenen Datenobjekte unverändert erhalten zu können. Für

ergänzte Funktionalitäten werden weitere DO eingeführt, die im Betriebsmodus SICCT aber nicht im Modus BCS nutzbar sind.

5.5.10.1 Data Objects supported in BCS-Mode

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 interval ▪ time until 1st input by Keypad.

Tabelle 5 : 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			

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 OUT	PPS - Request Byte Sequence 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 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 Used within SICCT commands ▪ SICCT REQUEST ICC ▪ SICCT RESET CT / ICC
DLPARAM DO	TLV DO	'73'		✓	man	OUT	Download Parameter Data Object ▪ Max. Blocksize ▪ Timeout Value Used within SICCT commands ▪ SICCT DOWNLOAD INIT
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

Data Object	Type	Tag	Mode			IN / OUT	Description
			BCS	SICCT			
WT DO	TLV DO	'80'	✓	✓	man	IN	Waiting Time Data Object codes time periods for user interactions ▪ card insertion interval ▪ card removal interval time until 1st input by Keypad. 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
RFIDS DO	TLV DO	'83'		✓	opt	OUT	RFID Token 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 all SICCT commands
CS DO	TLV DO	'85'		✓	opt	IN OUT	Character Set Data Object indicates display type and application label specific charactere set. Used within SICCT commands ▪ SICCT GET STATUS
SMTBD DO	TLV DO	'A0'		✓	opt	IN	SICCT Message-To-Be-Displayed Data Object combines an Applicatiuon 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-

Data Object	Type	Tag	Mode			IN / OUT	Description
			BCS	SICCT			
						OUT	DO. Used within SICCT commands ▪ SICCT GET STATUS ▪ SICCT SET STATUS
FU CON DO	TLV DO	'A2'		✓	opt	IN	Functional Unit Context Data Object Used within SICCT commands ▪ 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
RFIDAS DO	TLV DO	'64'		✓	opt	OUT	RFID Antenna 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 TERMINATE COMMAND ▪ SICCT GET STATUS
CTSESS DO	TLV DO	'69'		✓	man	IN OUT	Cardterminal Session Data Object Used within SICCT commands ▪ SICCT INIT CT SESSION ▪ SICCT CLOSE CT SESSION

Tabelle 6 : Overview SICCT Data Objects in SICCT Mode

5.5.10.3 TLV-Data Objects

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

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.

<table>
<tr><th>Field</th><th>Bytes</th><th colspan="6">Coding</th></tr>
<tr><td rowspan="12">Tag</td><td colspan="7">one byte coding</td></tr>
<tr><td rowspan="6">1</td><td rowspan="4">b8 b7</td><td>00</td><td colspan="4">Universal</td></tr>
<tr><td>01</td><td colspan="4">Application Context</td></tr>
<tr><td>10</td><td colspan="4">Context specific</td></tr>
<tr><td>11</td><td colspan="4">Private</td></tr>
<tr><td rowspan="2">b6</td><td>0</td><td colspan="4">primitive data object</td></tr>
<tr><td>1</td><td colspan="4">constructed data object</td></tr>
<tr><td>b5...b1</td><td>00000
...
11110</td><td>'00' ... '1E'</td><td>0 ... 30</td><td colspan="2">Tag Number</td></tr>
<tr><td colspan="7">two bytes coding</td></tr>
<tr><td rowspan="4">1</td><td>b8 b7</td><td colspan="5">see above</td></tr>
<tr><td rowspan="2">b6</td><td>0</td><td colspan="4">primitive data object</td></tr>
<tr><td>1</td><td colspan="4">constructed data object</td></tr>
<tr><td>b5...b1</td><td>11111</td><td colspan="4">Indicates two byte tag :
tag number cotained in the next (2nd) byte</td></tr>
<tr><td rowspan="2">2</td><td>b8</td><td>0</td><td rowspan="2">'1F' ... '7F'</td><td rowspan="2">31 ... 127</td><td rowspan="2">Tag Number</td></tr>
<tr><td>b7...b1</td><td>0011111
...
1111111</td></tr>
<tr><td rowspan="15">LEN</td><td colspan="7">one byte coding</td></tr>
<tr><td rowspan="2">1</td><td>b8</td><td>0</td><td rowspan="2">'00'
...
'7F'</td><td colspan="2">Range <LEN></td></tr>
<tr><td>b7...b1</td><td>0000000
...
1111111</td><td colspan="2">0 ... 127</td></tr>
<tr><td colspan="7">two byte coding</td></tr>
<tr><td rowspan="2">1</td><td>b8</td><td>1</td><td rowspan="2">'81'</td><td colspan="2" rowspan="2">129</td></tr>
<tr><td>b7...b1</td><td>0000001</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">b8...b1</td><td rowspan="2">00000000
...
11111111</td><td colspan="3">Range <LEN></td></tr>
<tr><td>'00'
...
255</td><td colspan="2">0 ... 255
Note: It shall be allowed to start with zero.</td></tr>
<tr><td colspan="7">three byte coding</td></tr>
<tr><td rowspan="2">1</td><td>b8</td><td>1</td><td rowspan="2">'82'</td><td colspan="2" rowspan="2">130</td></tr>
<tr><td>b7...b1</td><td>0000010</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">b8...b1</td><td rowspan="2">00000000
...
11111111</td><td>High Byte</td><td colspan="2" rowspan="2">Range <LEN>
< HighByte | LowByte >
Note: It shall be allowed to start with zero.</td></tr>
<tr><td>'00'
...
'FF'</td></tr>
<tr><td>3</td><td>b8...b1</td><td>00000000
...
11111111</td><td>Low Byte
'00'
...
'FF'</td><td>'0000'
...
'FFFF'</td><td>0 ... 65535</td></tr>
<tr><td rowspan="2">Value</td><td colspan="7"><LEN> == 0 : absent - no value data - empty data object</td></tr>
<tr><td>1
:
<LEN></td><td colspan="6">1 .. 65535 bytes of data</td></tr>
</table>

Tabelle 7 : TLV-Coding according ASN.1-BER

5.5.10.4 Answer-To-Reset Data Object

Das Answer-To-Reset Datenobjekt liefert die unveränderte Reset-Antwort einer Chipcard zurück, wie diese von der Chipkarte an das Kartenterminal gesendet und vom Kartenterminal empfangen wurde.
Das DO wird in Anlehnung an ISO 7816-6 ASN.1 kodiert.

Answer-To-Reset Data Object						
TAG	'5F41'		Two byte tag according ISO 7816-6: Answer-To-Reset			
			Tag coding according 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 7816-3	
		7816-10 ICC	Synchronous CC ATR	2 ... 4 Bytes	according 7816-10	
	PICC	PICC Reset information - max. 255 Bytes				
		PICC-A	Answer-To-Select Information (ATS)	3 ... 255 Bytes	PICC-A ATS according 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 Resonse		

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

Application Familiy Identifier (AFI)			
family	sub-family	Meaning according ISO 14443-3	Remark
MSB	LSB	Type of requested (responding) PICC	
'0'	'0'	All families and sub-families	no application preselection
X	'0'	All sub-families of family X	application preselection
X	Y	All sub-families Y of family X	
'0'	Y	Propoietary 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	MBLI	CID	CRC_B		ATTRIB Response according

Response	Maximum Buffer Length Index Type B	Card Indentifier Range: 0 ... 14	Cyclic Redundany Check Error Detection Code Type B 2 bytes	ISO 14443-3 given to an ATTRIB command without higher layer Information (command) field.

5.5.10.5 Historical Byte Data Object

Das Historical Byte Data Object ist eine Teilmenge des Answer-To-Reset Datenobjekts.
Das DO wird in Anlehnung an ISO 7816-6 ASN.1 kodiert.

Historical Byte Data Object				
TAG	'5F52'	Two byte tag according ISO 7816-6: Historical Bytes		
		Tag coding according 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 ISO 14443-4

5.5.10.6 CardTerminal Manufacturer Data Object

Das CardTerminal Manufacturer Data Object beinhaltet invariante Angaben zum Kartenterminal (pre-issuing data) , die der Hersteller zum Zweck der Erkennung zum Zeitpunkt der Herstellung eingebracht hat.
Diese Angaben dienen der Erkennung folgender Informationen

- CardTerminal Manufacturer
- CardTerminal Type
- CardTerminal Software Version
- (optional) Discretionary Data, z. B. Serial Number.

CardTerminalManufacturer Data Object				
TAG	'46'	One byte tag according MCT-Specifications: CardTerminalManufacturer Data Object		
		Tag coding according 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 <= 30			
	'00' ... '1E'	0 <= LEN <= 30		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 <=-15 Bytes
	Cardterminal Manufacturer	Cardterminal Type	Cardterminal Software Version	Discretionary Data
	RID provided	Vendor 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 the RID German National Registration Authority

Data		Len		Description												
				2 byte Country Code according ISO 3166					3 byte Manufacturer-Acronym.							
				Germany												
				'44'		'45'		'20'		'20'		'20'				
CTT	Cardterminal Type	5	man	5 Byte ASCII String- padded with Space ('20')												
				Vendor specific Coding - recommended : "SICCT"												
				'53'		'49'		'43'		43'		'54'				
CTSV	Cardterminal Software Version	5	man	5 Byte ASCII String- padded with Space ('20')												
				Major Number		Major Number		Minor Number		Minor Number		Release Number				
				'30'		'31'		'30'		'30'		'20'				
DD	Discretionary Data	0 ... 15	opt	optional, but recommended, ASCII String												
				Vendor specific Coding - recommended : Unique Identifier												
				Identification String or Number e.g. Serial Number												
				'01'	'02'	'03'	'04'	'05'	'06'	'07'	'08'	'09'	'0A'	'0B'	'0C'	'0D'	'0E'	'0F'

5.5.10.7 ICC Status Data Object

Das ICC Status Data Object stellt ein Objekt dar, um den Betriebsstatus von kontaktbehafteten Chipkarten (nach ISO 7816-3 / 7816-10) vom Kartenterminal mittels eines einzigen Datenobjekts abfragen zu können.

ICC Status Data Object				
TAG	'80'	One byte tag according MCT-Specifications: : ICC Status Data Object		
		Tag coding according 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>
	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

Data		Len bytes	Description			
			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 ist 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 stellt ein Objekt dar, 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							
TAG	'81'	One byte tag according MCT-Specifications: Functional Unit Data Object					
		Tag coding according 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'	'01'	'00' ...'FF'	256 <= LEN <= 512	Three byte coding		
		'02'	'00'				
VALUE	Coding, when operating in BCS Mode						
	List of existent and addressable Functional Units						
	BCS-Mode - Direct Coding - At maximum 255 Functional Units per one SICCT Terminal						
	The list is given by a set of one byte values according the Functional Unit Index value - see 5.5.9.4						
	ICC Slots			[1st Display]	[1st Keypad]	[1st Printerport]	[1st Fingerprint]	[...]
	'01'	[...]	['0E']	[40]	[50]	[60]	[70]	[...]
	Coding, when operating in SICCT Mode						
	Referenced Coding - At maximum 255 Functional Units per one SICCT Terminal						
	The list is given by a set of two byte values according the Functional Unit Index value - see 5.5.9.4						
	ICC Slots	RFID Antenna	RFID Slots	Displays	Keypads	...	Vendor specific FUs
	'0001' [... '00FF']	['1000']	['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) stellt ein TLV-Objekt zur Kodierung von Functional Units in Referenced Coding Darstellung dar. Der Wert des Objekts gibt einen Ein-Byte-Typqualifizierer der Functional Unit an. Nachfolgend erscheint der korrespondierende Indexwert, den das SICCT Terminal in der Ressourcentabelle zusammen mit einem Friendly Name verwaltet.

Functional Unit Index Data Object			
TAG	'84'	One byte tag according SICCT-Specifications: Functional Unit Index Data Object	
		Tag coding according 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 (1 Byte)		
	'00'... 'A0'	See 5.5.9.5	
	Functional Unit Index Number (1Byte)		
	'00' ... 'FF'	See 5.5.9.5	

5.5.10.10 Functional Unit Context Data Object

Das FU Context Data Object (FU CON DO) dient dazu, einen Ausführungskontext aus verschiedenen Functional Units für die Ausführung eines SICCT-Kommandos definieren zu können.

Das FU Context Data Object stellt ein zusammengesetztes TLV-Datenobjekt dar, welches minimal zwei und maximal drei Functional Unit Index Data Objects (FUI DO) enthalten kann. Die Anzahl und Kombination 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 / RFID-Slot
- einem Ausgabegerät: ein Display
- einem Eingabegerät: ein Keypad oder eine biometrische Sensoreinheit.

Functional Unit Context Data Object			
TAG	'A2'	One byte tag according SICCT-Specifications: Functional Unit Context Data Object	
		Tag coding according 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) kann nur in Referenced Coding Darstellung und anstelle eines FUI Datenobjekts verwendet werden.

5.5.10.11 Functional Unit Name Data Object

Das Functional Unit Name Data Object (FU Name DO) stellt ein TLV-Objekt zur Kodierung des 'Friendly Names' einer FU dar. Der 'Friendly Name' darf maximal 32 (darstellbare) Zeichen umfassen, und wird entsprechend dem angegeben Index (FUI-DO) zugeordnet und in der Ressourcentabelle vom SICCT-Terminal verwaltet.

Functional Unit Name Data Object			
TAG	'A1'	One byte tag according SICCT-Specifications: Functional Unit Name Data Object	
		Tag coding according 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 : 6 <= LEN <= 38		
	'06' ... '26'	6 <= LEN <= 38	One byte coding
VALUE	Functional Unit Index Data Object		
	'84'	L = '02'	Functional Unit Index 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

5.5.10.12 RFID Token Status Data Object

Das RFID Token Status Data Object stellt eine Erweiterung dar, um den Betriebsstatus von kontaktlosen Token, wie kontaktlose Chipkarten, vom Kartenterminal mittels eines einzigen Datenobjekts abfragen zu können. Wie das ICC Status Datenobjekt liefert das Kommando den Status aller an der Funktionseinheit RF Antenne sichtbaren RFID Token zurück.

RFID Token Status Data Object					
TAG	'83'	One byte tag according SICCT-Specifications: RFID Status Data Object			
		Tag coding according 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	RFID Status for max. 14 RFID slots / token				
	Coding of RFID Status Byte see table below				
	RFID_SB1		RFID_SB2	...	RFID_SB<n> '01' <= n <= '0E'
	RFID Token Status 1		RFID Token Status 2	...	RFID Token Status <n>

RFID Status Byte Coding			
RFID Token Status Byte			
	b8…b5		
	b4..b1		

5.5.10.13 Reduced / Card Terminal Status Data Object

Das Cardterminal Status Data Object stellt eine Erweiterung dar, um Parameter des Kartenterminals mittels eines einzigen Datenobjekts abfragen bzw. setzen zu können. Es handelt sich um ein 'constructed' ASN.1-Datenobjekt, welches weitere ASN.1-Datenobjekte aufnehmen kann.

Reduced / Card Terminal Status Data Object		
TAG	'63'	One byte tag according SICCT-Specifications:

		Tag coding according 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 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
	'82'	'01'	'00' ...'FF'	256 <= LEN <= 512	Two byte coding
		'02'	'00'		
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: RFID Antenna Status Data Object		See 5.5.10.14
	[✓]	[✓]	optional - if present: RFID Token Status Data Object		See 5.5.10.12

5.5.10.14 RFID Antenna Status Data Object

Das RFID Antenna Status Data Object stellt eine Möglichkeit dar, um Parameter der RFID-Antenne mittels eines einzigen Datenobjekts abfragen oder setzen zu können. Es handelt sich um ein 'constructed' ASN.1-Datenobjekt, welches weitere ASN.1-Datenobjekte aufnehmen kann.

RFID Antenna Status Data Object					
TAG	'64'	One byte tag according SICCT-Specifications:			
		Tag coding according ASN.1 BER see 5.5.10.3			
		BER-Coding : Application context, constructed, Tag-Number = 4 ('04')			
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	RFID Antenna Options				
	Tag	LEN max.	Remark:.		
	'02'	'01'	Maximum Number of RFID Token supported by this antenna		
			'00'	RFID Antenna in state 'OFF' and not ready to communicate with any RFID Token	
			'01' ... '0E'	1 <= Number RFID TOKEN <= 14	

5.5.10.15 ICC Interface Status Data Object

Das ICC Interface Status Data Object stellt eine Erweiterung dar, um Kommunikationsparameter einer Chipkartenfunktionseinheit vom Kartenterminal mittels eines Datenobjekts abfragen bzw. setzen zu können. Es handelt sich um ein 'constructed' ASN.1-Datenobjekt, welches weitere ASN.1-Datenobjekte aufnehmen kann.

ICC Interface Status Data Object					
TAG	'65'	One byte tag according SICCT-Specifications:			
		Tag coding according 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	

<table>
<tr><td rowspan="30">VALUE</td><td colspan="10">Interface Device Protocol Options</td></tr>
<tr><td>Tag</td><td>LEN max.</td><td colspan="8">Remark: Object construction according 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.</td></tr>
<tr><td rowspan="11">'80'</td><td rowspan="11">'04'</td><td colspan="8">Current Protocol Type</td></tr>
<tr><td rowspan="9">ICC</td><td colspan="7">Contact bound Chipcard Protcol Setting</td></tr>
<tr><td>b1</td><td>T=0</td><td colspan="2">Asynchronous T=0 Protocol</td><td colspan="2" rowspan="2">According ISO 7816-3</td></tr>
<tr><td>b2</td><td>T=1</td><td colspan="2">Asynchronous T=1 Protocol</td></tr>
<tr><td>b3</td><td>S=8</td><td>SDAP</td><td rowspan="2">I²C Memory Chipcard</td><td colspan="2">One Byte Address Mode</td></tr>
<tr><td>b4</td><td>S=8</td><td>SDAP</td><td colspan="2">Two Byte Address Mode</td></tr>
<tr><td>b5</td><td>S=9</td><td>3WBP</td><td>Memory Chipcard</td><td colspan="2" rowspan="3">According ISO 7810</td></tr>
<tr><td>b6</td><td>S=10</td><td>2WBP</td><td>Memory Chipcard</td></tr>
<tr><td>b7</td><td>S=11</td><td>FCBP</td><td>Memory Chipcard</td></tr>
<tr><td>b8</td><td>'0'</td><td colspan="3">Contact bound</td></tr>
<tr><td rowspan="5">PICC</td><td colspan="7">Contactless Chipcard Protcol Setting</td></tr>
<tr><td colspan="2">b1..b3</td><td colspan="4">RFU</td></tr>
<tr><td>b4</td><td>T=CL</td><td colspan="2">Proximity Smart Card</td><td colspan="2">According ISO 14443-4</td></tr>
<tr><td colspan="2">b5 .. b7</td><td colspan="4">RFU</td></tr>
<tr><td>b8</td><td>'1'</td><td colspan="4">Contactless</td></tr>
<tr><td rowspan="3">'81'</td><td rowspan="3">'04'</td><td rowspan="2">ICC</td><td colspan="8">Current Clock</td></tr>
<tr><td colspan="8">Current ICC CLK frequency in KHz encoded as little endian integer value.
Example: 3.58 MHz is encoded as the integer value 3580.</td></tr>
<tr><td>PICC</td><td colspan="8">Current RF frequency</td></tr>
<tr><td>'82'</td><td>'04'</td><td>ICC</td><td colspan="8">Current F (Clock Conversion Factor)</td></tr>
<tr><td rowspan="2">'83'</td><td rowspan="2">'04'</td><td>ICC</td><td colspan="8">Current D (Bit Rate Conversion Factor)</td></tr>
<tr><td>PICC</td><td colspan="8">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 little endian integer, which is bit rate factor for the direction PICC to PCD.
▪ DR is encoded by the least significant 2 bytes as a little endian integer, which is bit rate factor for the direction IFD to PICC</td></tr>
<tr><td>'84'</td><td>'04'</td><td>ICC</td><td colspan="8">Current N (Guard Time Factor)
N encoded as a little endian integer.</td></tr>
<tr><td>'85'</td><td>'04'</td><td>ICC</td><td colspan="3">Current W (Work Waiting Time)
only T=0 Protocol</td><td colspan="5">Current W</td></tr>
<tr><td rowspan="2">'86'</td><td rowspan="2">'04'</td><td>ICC</td><td colspan="3">Current IFSC (Information Field Size Card)
only T=1 Protocol</td><td colspan="5">Current IFSC</td></tr>
<tr><td>PICC</td><td colspan="8">Current FSC (maximum frame size which the PICC is able to receive)
FSC encoded as a Little Endian integer, if it applies.</td></tr>
<tr><td rowspan="2">'87'</td><td rowspan="2">'04'</td><td>ICC</td><td colspan="3">Current IFSD (Information Filed Size Device)
only T=1 Protocol</td><td colspan="5">Current IFSD</td></tr>
<tr><td>PICC</td><td colspan="8">Current FSD (maximum frame size which the IFD is able to receive.)
FSD encoded as a Little Endian integer, if it applies.</td></tr>
<tr><td rowspan="2">'88'</td><td rowspan="2">'04'</td><td>ICC</td><td colspan="3">Current BWT (Block Waiting Time)
only T=1 Protocol</td><td colspan="5">Current BWT</td></tr>
<tr><td>PICC</td><td colspan="8">Current FWT (Frame Waiting Time)
FWT encoded as a Little Endian integer, if it applies.</td></tr>
<tr><td>'89'</td><td>'04'</td><td>ICC</td><td colspan="3">Current CWT (Character Waiting Time)
only T=1 Protocol</td><td colspan="5">Current CWT</td></tr>
<tr><td>'8A'</td><td>'04'</td><td>ICC</td><td colspan="3">Current EDC (Error Detection Code)
only T=1 Protocol</td><td>'00'</td><td colspan="4">LRC</td></tr>
<tr><td colspan="3"></td><td>'01'</td><td colspan="4">CRC</td></tr>
</table>

5.5.10.16 ICC Interface Capabilities Data Object

Mittels des ICC Interface Capabilities Data Objects (INTFC DO) werden Eigenschaften einer physikalischen Chipkartenschnittstelle dargestellt. Es handelt sich um ein 'constructed' ASN.1-Datenobjekt, welches weitere ASN.1-Datenobjekte aufnehmen kann.

ICC Interface Capabilities Data Object						
TAG	'66'	One byte tag according SICCT-Specifications:				
		Tag coding according 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 / avalaible Mechanical Attributes			
			b1	1	swallowing mechanism	
			b2	1	ejection mechanism	
			b3	1	capture mechansim	
			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-Asycnchr.	
			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 ISO 14443-1	
			b4	1	supported according ISO 14443-2	
			b5	1	supported according ISO 14443-3	
			b6	1	supported according 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' .. '1100'		32 .. 4352 Bytes	

5.5.10.17 Display Capabilities Data Object

Mittels des Display Capabilities Data Objects (DSPLC DO) werden die Eigenschaften einer physikalischen Anzeigeeinheit dargestellt. Es handelt sich um ein 'constructed' ASN.1-Datenobjekt, welches weitere ASN.1-Datenobjekte aufnehmen kann.

Display Capabilities Data Object					
TAG	'67'		One byte tag according SICCT-Specifications:		
			Tag coding according 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 visbile lines		
			b8 ..b1	Number of visbile 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 per line (scrolling):		
			b81..b1	Number of virtual lines per line (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
			b61..b8		Number of LEDs: '0' <= u <= '3'
			'00'		none
	One or sequence of Character Set Data Objects				
	'85'	'01'	Character Set Data Object		
			b81..b1	Supported Character Set	See 5.5.10.20
	:	:	:	:	:

5.5.10.18 PPS - Request / Response Data Object

Das PPS Request / Response Data Object kodiert eine Protocol Parameter -Select-Message (PPS, nach ISO 7816-3 [STD5]) der Applikationsschicht, die das Kartenterminal unverändert während der Abarbeitung eine SICCT RESET ICC Kommandos an eine adressierte Chipkarte unmittelbar nach dem Reset der Chipkarte senden wird, sofern diese Bereitschaft zu einer Negotiation der Kommunikationsparameter im Answer-To-Reset signalisiert hatte.

Die Antwort der Chipkarte (PPS-Response) wird vom Kartenterminal in demselben Format als PPS-Response Data Object zurückgegeben, wenn die PPS-Kommunikation mit der Chipkarte stattgefunden hat. Die PPS-Response signalisiert der Applikationsschicht die aktuellen Kommunikationsparameter nach ISO7816-3 [STD5].

PPS - Request / Response Data Object							
TAG	'53'	One byte tag according ISO 7816-6: Discretionary Data Objects					
		Tag coding according 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 TDDU: PPS Request according 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 baudrate 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			

Discrete Protocol-Parameter Values for: PPS Request / Response TDDU according ISO 7816-3			
PPPS	b8 .. b1	'FF'	PPS request or response
PPPS0	b8	1	RFU

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

Discrete Protocol-Parameter Values for: PPS Request / Response TDDU according ISO 7816-3						
		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	
		1010	'A'	768	7,5	The support of frequencies higher than 5 MHz is optional and vendor dependent.
		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		

Discrete Protocol-Parameter Values for: PPS Request / Response TDDU according ISO 7816-3						
		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.			

5.5.10.19 Application Label Data Object

Das Application Label Data Object (APPL DO) kodiert eine ASCII-kodierte Display-Message, die das Kartenterminal dem Anwender zur Benutzerführung darstellen soll. Aufgrund der Kompatibilität zum BCS-Modus, wird eine implizite ASCII-Kodierung der Display-Messages vorausgesetzt.
Vor jeder neuen Ausgabe einer Display-Message löscht das Terminal das Display.
Ein leeres Datenobjekt bzw. eine leere Message bewirkt das Löschen des Displays am Kartenterminal.
Das APPL DO kann in einigen Befehlskontexten alternativ zum SICCT Message-To-be-Displayed DO, welches neben einer Display-Message explizit auch den Zeichsatz angibt, verwendet werden. Innerhalb eines SICCT Befehls darf entweder die eine oder andere Objektvariante (APPL oder SMTBD Objects) aber nicht beide Formen zugleich zum Einsatz kommen.

Application Label Data Object			
TAG	'50'	One byte tag according ISO 7816-6: Application Label	
		Tag coding according 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 a two line display with 16 characters each.	
VALUE	Application Label (Display Message)		
	Sequence of ASCII characters.		

Display-Texte sind in ASCII zu codieren (ISO 646 bzw. nationale Referenz-Version). Der Zeichensatz, der mindestens unterstützt werden muß, sofern das SICCT über ein Display verfügt, umfäßt die folgenden Zeichen.

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	miscellaneous and special characters	' '	'20'	Leerzeichen	Space
		'+'	'2B'	Pluszeichen	plus sign
		','	'2C'	Komma	comma
		'-'	'2D'	Minuszeichen	minus sign

Coding of Display Messages					
Minimal supported ASCII Character Set for Display Messages					
		'.'	'2E'	Punkt	dot
		':'	'3A'	Doppelpunkt	colon
		'='	'3D'	Gleichheitszeichen	equal sign
		'?'	'3F'	Fragezeichen	question mark

5.5.10.20 Character Set Data Object

Das Character Set Data Object kodiert einen Zeichensatz, den das Kartenterminal für eine Display-Message verwenden soll.

Character Set Data Object			
TAG	'85'	One byte tag according SICCT-Specifications: Character Set	
		Tag coding according 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 amerikanische 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	Latin-5			

Character Set Index Value	Normative Reference	Description
	8859-9	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 Lappisch (Sami).
'0B'	ISO / IEC 8859-11	Thai
		ISO 8859-11 versucht möglichst viele Zeichen der Thai-Schrift abzudecken.
'0C'	not supported	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 abzudecken wie auch ISO 8859-4 (*Latin-4* eher baltisch) und ISO 8859-10 (*Latin-6* 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 abzudecken.
'0F'	ISO / IEC 8859-15	Latin-9
		ISO 8859-15 versucht möglichst viele Sonderzeichen vorwiegend 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 die südosteuropäischen Albanisch, Kroatisch, Ungarisch, Italienisch, Polnisch, Rumänisch und Slowenisch, aber auch Finnisch, Französisch, Deutsch und irisches Gälisch (neue Rechtschreibung). Im Vergleich zu seinen Geschwistern legt ISO 8859-16 viel mehr Wert auf Buchstaben mit Diakriten und verzichtet dafür auf andere (Satz-)Zeichen.
'20'	UTF-8	UTF-8 according RFC 3629 is a transformation format of ISO 10646 (Unicode)
		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 abbilden 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 ausschließlich 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.

5.5.10.21 SICCT Message-To-Be-Displayed Data Object

Das SICCT Message-To-Be-Displayed Data Object (SMTBD DO) verbindet einen Zeichensatz mit einer entsprechend strukturierten Display Message, die das Kartenterminal über ein Display anzeigen soll.
Vor jeder neuen Ausgabe einer Display-Message löscht das Terminal das Display.
Ein leeres Datenobjekt bzw. eine leere Message bewirkt das Löschen des Displays am Kartenterminal.
Das SMTBD DO DO kann in einigen Befehlskontexten alternativ zum Application Label DO, welches ASCII-kodierte Display-Messages angibt, verwendet werden.

SICCT Message-To-Be-Displayed Data Object				
TAG	'A0'	One byte tag according SICCT-Specifications: SICCT Message-To-Be-Displayed		
		Tag coding according 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			
	one or two byte 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	One Character Set DO and corresponding Application Label DO.			
	Character Set Data Object		see 5.5.10.20	

SICCT Message-To-Be-Displayed Data Object			
	'04'	The actual supported length depends on the capabilities of the selected FU of type 'display' . At minimum 32 ('20') characters according a two line display with 16 characters each.	
		'00' <= L <= '7F'	ASN.1 coded OCTETSTRING (Universal 4)
			0 <= L <= 127
			Byte Sequence formatted according the selected character set giving the message to be displayed.

5.5.10.22 Waiting Time Data Object

Das Waiting Time Data Object kodiert das maximale Zeitintervall (in Sekunden) für notwendige Benutzerinteraktionen am Kartenterminal.

Aktion	Anwendung in Kommando	Referenz
Action	Usage within Command	Reference
max. Zeitdauer bis zum Stecken einer Chipkarte	SICCT REQUEST ICC	See 5.12
max. Zeitdauer bis zur Entnahme einer Chipkarte	SICCT EJECT ICC	See 5.13
max. Zeitdauer zur Anzeige einer Textmeldung	SICCT OUTPUT	See 5.17
max. Zeitdauer bis zur ersten Tastenbetätigung bei einer KeyPad-Eingabe bzw. PIN-/ Passwort-Eingabe	SICCT INPUT	See 5.16
	SICCT PERFORM VERIFICATION	See 5.18
max. Zeitdauer bis zur Betätigung der Bestätigungstaste nach einer erfolgten KeyPad-Eingabe bzw. PIN-/ Passwort-Eingabe	SICCT MODIFY VERIFICATION DATA	See 5.3.1

Waiting Time Data Object			
TAG	'80'		One byte tag according MCT-Specifications: Waiting Time Data Object
			Tag coding according 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
			Note: The special value '00' overrides the timing monitoring within the terminal in a way that the command be executed and the will terminal return immediately. *This matches the same behaviour as in case no WTDO has been send with a SICCT command.*

5.5.10.23 Command–To-Perform Data Object

Das Command-To-Perform Data Object kodiert ein Template in dem das Kartenterminal ein Chipkarten-APDU oder Teile eines Chipkarten-APDus einbetten soll.

Der Inhalt und Aufbau des Value-Feldes variiert in Abhängigkeit der Anzahl der erforderlichen PIN-Nummern und orientiert sich am Anwendungsfall des Datenobjekts im Kontext von

- SICCT PERFORM VERIFICATION
- SICCT MODIFY VERIFICATION DATA
- Zu beachten ist, dass zweite (2^{nd}) Insertion Position für die Ausführung des Kommandos SICCT MODIFY VERIFICATION DATA verpflichtend ist. Dieses Byte kann beim SICCT PERFORM VERICATION entfallen oder enthalten sein, würde dann aber vom Terminal ignoriert werden.

Command-To-Perform Data Object							
TAG	'52'	One byte tag according MCT-Specifications: Command To Perform					
		Tag coding according 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				
		'FF'	In case of biometric identification				
	1st Insertion Position Byte	'06' ... 'FF'	Mandatory Field - Insertion Position for the (1st) PIN				
			SICCT PERFORM VERIFICATION	Mandatory Field - Insertion position for the PIN			
			SICCT MODIFY VERIFICATION DATA	Mandatory Field - Insertion position for the 1st 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 1st insertion position is 'equal or greater than 06' (after a five byte APDU: CLA, Len, P1, P2, Lc).				
	[2nd Insertion Position Byte]	'06' ... 'FF'	Conditional Field - Insertion Position for the 2nd PIN				
			SICCT PERFORM VERIFICATION	Note: This field must not be transmitted. for this command.			
			SICCT MODIFY VERIFICATION DATA	Conditional Field - Insertion position for the 2nd 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 2nd insertion position is greater than the 1st insertion position (after a five byte APDU: CLA, Len, 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			

Command-To-Perform Data Object						
				Change Cardholder Verification according GSM 11.11		
				Change Cardholder Verification according prEN 726-3		
			'26'	**DISABLE CHV**		
				Disable Cardholder Verification according ISO 7816-8		
				Disable Cardholder Verification according GSM 11.11		
				Disable Cardholder Verification according prEN 726-3		
			'28'	**ENABLE CHV**		
				Enable Cardholder Verification according ISO 7816-8		
				Enable Cardholder Verification according GSM 11.11		
				Enable Cardholder Verification according prEN 726-3		
			'2A'	**PERFORM SECURITY OPERATION**		
				P1	P2	according ISO 7816-8, 5.9. table 13
				'82'	'80'	PSO (Encipher) operation according ISO 7816-8 in order to support VERSA-concept (VERSA : "distributed signature workplaces").
				'84'		
				'86'		
			'2C'	**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.
		0001 ... 1100		‘1’ … ‘12’	Number of PIN digits or characters
					Minimal 1 digit / character
					Maximal 12 digits / characters
	b4..b3	0	0	‘0’	default
		0	1	’1’ ... ‘3’	RFU
		1	0		
		1	1		
	b2 .. b1	0	0	BCD coded PIN	
		0	1	T.50 coding according International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) - Information technology - 7-bit coded character set for information interchange, see table below	
				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	0	Format 2 PIN Block according to ISO 9564-1, see table below	
		1	1	RFU	
	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	

T.50 coding - technology - 7-bit coded character set for information interchange				
b8…b1	0011 0000	'30'	0	according 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 4 up to 10 nibbles							

Die Bedingungen, wie Informationen vom Kartenterminal einzufügen sind, hängt vom entsprechenden Kartenkommando (APDU) und von dem PIN-Format ab. Eine PIN nach ISO 9564-1 hat minimal 4 und maximal 12 Zeichen (digits). Für andere PIN-Darstellungen sollen Eingaben im Bereich 1 bis max. 12 Zeichen zulässig sein.

Für die Ergänzung von PIN-Eingaben am Terminal in den Kommandos
- SICCT PERFORM VERIFICATION
- SICCT MODIFY VERIFICATION DATA

sind folgende Formen des APDU möglich
- 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.

5.5.10.24 Sequence Number Data Object

Kommandos und Antworten werden nach dem Command-Response-Modell ausgetauscht. Der Aufrufer vergibt für jedes Kommando eine Kommandosequenznummer, die der Aufrufer lediglich im Übertragungsprokoll angibt, über die zu einem späteren Zeitpunkt aber auch der Bearbeitungsstatus abgefragt oder modifiziert werden kann.

Das Sequence Number Data Object dient der Übermittlung der Sequenznummer und
- im Kontext von SICCT GET STATUS zur Abfrage des Bearbeitungsstatus eines zuvor an das SICCT Terminal gesendeten Kommandos,
- im Kontext von SICCT TERMINATE COMMAND zum Beenden eines zuvor an das SICCT Terminal gesendeten Kommandos.

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

Innerhalb einer Antwort vom Kartenterminal enthält das Datenobjekt zusätzlich ein weiteres INTEGER-Datenobjekt, welches den Bearbeitungsstatus angibt.

Sequence Number Data Object		
TAG	'68'	One byte tag according SICCT-Specifications:
		Tag coding according ASN.1 BER see 5.5.10.3
		BER-Coding : Application context, constructed, Tag-Number = 8 ('08')

<table>
<tr><th colspan="7">Sequence Number Data Object</th></tr>
<tr><td rowspan="11">LEN</td><td colspan="6">LEN coding see 5.5.10.3</td></tr>
<tr><td colspan="6">one byte coding - LEN in the range of : 6 <= LEN <= 8</td></tr>
<tr><td colspan="3" rowspan="3">'04' <= LEN <= '08'</td><td colspan="2" rowspan="3">LEN = 4 .. 8</td><td>One byte coding</td></tr>
<tr><td>DO in Request: 4 Bytes</td></tr>
<tr><td>DO in Response: 8 Bytes</td></tr>
<tr><td colspan="4">Command Sequence Number</td><td>A distinct two byte value</td></tr>
<tr><td rowspan="2">'02'</td><td rowspan="2">'L = '02'</td><td><high byte></td><td><low byte></td><td>0000' <= Sequence Number <= 'FCFF'</td><td rowspan="11">ASN.1 Universal 2 Coding</td></tr>
<tr><td>'00' ... 'FF'</td><td>'00' ... 'FF'</td><td>Note:
'FD00' <= Sequence Number <= 'FFFF'
signal Events sent by the SICCT to the host entity.</td></tr>
<tr><td colspan="4" rowspan="2">[Command Sequence Status Word]</td><td>Presence - Conditional</td></tr>
<tr><td>For requests
not. present.</td></tr>
<tr><td rowspan="5">['02']</td><td rowspan="5">['00' <= L <= '02']</td><td>[<high byte>]</td><td>[<low byte>]</td><td>For responses:
A distinct two byte value set by the CT for the response APDU.</td></tr>
<tr><td>'90'</td><td>'00'</td><td>Sequence Number found:
Command queued and not in processing state</td></tr>
<tr><td>'62'</td><td>'0x'</td><td>Sequence Number found:
command in processing state '0x'</td></tr>
<tr><td>'64'</td><td>'0x'</td><td>Sequence Number found:
processing state found at stage '0x' - state cannot be changed.</td></tr>
<tr><td>'6F'</td><td>'00'</td><td>Sequence Number not found:
no information given on processing state</td></tr>
</table>

Use Cases for Sequence Number Data Object		
SICCT Command	Description	Reference
SICCT GET STATUS	Query command execution state for given Sequence No DO	see 5.14
SICCT TERMINATE COMMAND	Terminate command execution for given Sequence No DO	see 5.8

5.5.10.25 Download Parameter Data Object

Das Download Parameter Data Object enthält einzelne Parameter zum Handling des Downloads zum Austausch der Gerätefirmware:

- Timeout
- Maximale Download Data Object Größe

<table>
<tr><th colspan="6">Download Parameter Data Object</th></tr>
<tr><td rowspan="3">TAG</td><td colspan="3" rowspan="3">'73'</td><td colspan="2">One byte tag according ISO 7816-6: Discretionary Data Objects</td></tr>
<tr><td colspan="2">Tag coding according ASN.1 BER see 5.5.10.3</td></tr>
<tr><td colspan="2">BER-Coding : Application context, constructed, Tag-Number = 19 ('13')</td></tr>
<tr><td rowspan="5">LEN</td><td colspan="5">LEN coding see 5.5.10.3</td></tr>
<tr><td colspan="5">one byte coding - LEN in the range of : 0 <= LEN <= 65535</td></tr>
<tr><td colspan="3">'00' ... '7F'</td><td>0 <= LEN <= 127</td><td>One byte coding</td></tr>
<tr><td colspan="2">'81'</td><td>'80' ...'FF'</td><td>128 <= LEN <= 255</td><td>Two byte coding</td></tr>
<tr><td>'82'</td><td>'01'</td><td>'00' ...'FF'</td><td>256 <= LEN <= 65535</td><td>Two byte coding</td></tr>
</table>

Download Parameter Data Object					
		'02'	'00'		
VALUE	Download Parameter				
	Sequence of TLV-Objects				
	'80'	'01'	'xx'	Waiting Time Data Object	Timeout Waiting time in seconds how long the device needs to process last download data object.
	'02'	'01'	'xx'	ASN.1 BER Coding INTEGER (UNIVERSAL2)	Maximum Length of Download Data Object as number of bytes.
		'02'	'xxyy'		
	T	L	V	Other vendor specific Data Objects may follow	

5.5.10.26 Download Data Object

Das Download Data Object enthält jeweils ein Download-Datenpaket.

Download Data Object				
TAG	'73'		One byte tag according ISO 7816-6: Discretionary Data Objects	
			Tag coding according 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'	'01' '00' ...'FF'	256 <= LEN <= 65535	Two byte coding
		'02' '00'		
VALUE	Download Data Package			
	Vendor specific Data Objects or Fomat			

5.5.10.27 Download Termination Data Object

Das Download Termination Data Object enthält einzelne Parameter zum Abschluß des Downloads.

Download Termination Data Object					
TAG	'73'			One byte tag according ISO 7816-6: Discretionary Data Objects	
				Tag coding according 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'	'01'	'00' ...'FF'	256 <= LEN <= 65535	Two byte coding
		'02'	'00'		
VALUE	Download Parameter				
	Sequence of TLV-Objects				
	'80'	'01'	'xx'	Waiting Time Data Object	Timeout Waiting time in seconds how long the device needs to process last download data object or finish the download process.
	T	L	V	Other vendor specific Data Objects may follow	

5.5.10.28 CT SESSION Data Object

Das Cardterminal Session Data Object (CTSESS DO) enthält folgende Benutzerparameter zum rollenbasierten Zugriff und zur Identifikation einer Cardterminal Session

- Benutzername (Username)

- Benutzeridentifkation (Password)
- Session-Identifier (Session ID)

<table>
<tr><th colspan="8">CT Session Data Object</th></tr>
<tr><td rowspan="3">TAG</td><td colspan="2" rowspan="3">'69'</td><td colspan="5">One byte tag according SICCT-Specifications:</td></tr>
<tr><td colspan="5">Tag coding according ASN.1 BER see 5.5.10.3</td></tr>
<tr><td colspan="5">BER-Coding : Application context, constructed, Tag-Number = 9 ('09')</td></tr>
<tr><td rowspan="3">LEN</td><td colspan="7">LEN coding see 5.5.10.3</td></tr>
<tr><td colspan="7">one byte coding - LEN in the range of : 0 <= LEN <= 42</td></tr>
<tr><td colspan="2">'06' ... '2A'</td><td colspan="3">6 ... 42 Bytes</td><td colspan="2">One byte coding</td></tr>
<tr><td rowspan="10">VALUE</td><td colspan="7">CT Session User Parameter</td></tr>
<tr><td rowspan="3">Username Data Object</td><td rowspan="3">Username</td><td rowspan="3">1st TLV DO</td><td>Tag</td><td>'13'</td><td colspan="2">ASN.1 BER : Printable String</td></tr>
<tr><td>Length</td><td colspan="3">0 ... 12 Byte</td></tr>
<tr><td>Value</td><td colspan="3">0 ... 12 alphanumerical characters (Byte)</td></tr>
<tr><td rowspan="3">Password Data Object</td><td rowspan="3">Password</td><td rowspan="3">2nd TLV DO</td><td>Tag</td><td>'13'</td><td colspan="2">ASN.1 BER : Printable String</td></tr>
<tr><td>Length</td><td colspan="3">0 ... 12 Byte</td></tr>
<tr><td>Value</td><td colspan="3">0 ... 12 alphanumerical characters (Byte)</td></tr>
<tr><td rowspan="3">Session ID Data Object</td><td rowspan="3">Session ID</td><td rowspan="3">3rd TLV DO</td><td>Tag</td><td>'13'</td><td colspan="2">ASN.1 BER : Printable String</td></tr>
<tr><td>Length</td><td colspan="3">0 ...12 Byte</td></tr>
<tr><td>Value</td><td colspan="3">0 ... 12 alphanumerical characters (Byte)</td></tr>
</table>

Für den anonymen Zugriff können leere Datenobjekte mit der Länge Null verwendet werden.

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

- Ein Display-Text kann explizit per Kommandoparameter in Form auf den aktuellen Befehlskontext bezogener Standard-Texte oder explizit über Daten Objekte zur Anzeige gebracht werden.
- Vor der Ausgabe eines neuen Display-Textes wird das adressierte Display zuvor gelöscht.
- Textmeldungen können in Form von Application Label oder SICCT Message-To-be-Displayed Data Objects innerhalb von SICCT-Kommandos übergeben werden.
- 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.
- Das explizite Löschen eines Displays erfolgt durch die Übergabe eines leeren Message-Datenobjekts.
- Nach der Ausführung eines SICCT Kartenterminalkommandos bleiben Textmeldungen bestehen, bis eine neue Ausgabe diese überschreibt.
- Während der Ausführung kann das Terminal kontextbezogene Textnachrichten (z.B. Abbruch-Meldung nach Betätigung der Abbruch-Teste im Rahmen einer Benutzer-interaktion) über das Display ausgeben.
- Lokale Benutzerinteraktionen (z.B. Tastenecho bei Keypad-Eingabe, Abbruch-Meldung) werden direkt am Terminaldisplay dargestellt.
- Die Darstellung impliziter Textnachrichten kann bei bestimmten SICCT-Kommandos über eine Option unterdrückt werden. Diese Kommandos verhalten sich dann insofern 'display-neutral', dass auch keine vom Terminal generierten Nachrichten angezeigt werden.
- Die impliziten Display-Nachrichten beim Anfordern und Deaktivieren einer Chipkarte durch die SICCT-Kommandos REQUEST ICC und EJECT ICC entfallen, sofern eine Karte bereits eingeführt bzw. entnommen wurde.
- Die Darstellung einer Kartenterminal-Idle-Message erfolgt 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.
- Der Inhalt der Idle-Message kann herstellerspezifisch ausfallen und über die Geräte-administration anpassbar gehalten sein.

5.7 Command SICCT SELECT CT MODE

Mittels des SICCT SELECT CT MODE Kommandos kann zur Laufzeit der 'Aktive Betriebsmodus', entweder BCS Mode oder SICCT Mode, für das Kartenterminal ausgewählt werden. Die Aktivierung des gewählten Betriebsmodus erfolgt erst mittels eines nachfolgenden Resets des Kartenterminals.

5.7.1 Funktion

Fehlerfreier Betrieb

Das Kommando SICCT SELECT CT MODE richtet sich an den Kommandointerpreter. Der gewählte Betriebsmode wird als Parameter P2 übertragen. Das Kartenterminal übernimmt den gewählten Betriebsmode und sendet ein Statuswort SW1SW2 zurück. Der Betriebsmodus wird erst zum Zeitpunkt der Abarbeitung des nächst folgenden SICCT RESET Kommandos aktiviert.

Konfigurationsdaten

Eine gewählte Einstellung gilt für das gesamte Kartenterminal bis zum nächsten Kaltstart und verändert nicht die Terminalkonfiguration. Die Umschaltung bzw. die Speicherung des Betriebsmodus erfolgt nicht persistent. Nach einem Kaltstart (Neustart) startet das Kartenterminal in der Betriebsart wie diese durch die Konfigurationsdaten des Terminals ('Standard-Betriebsmodus') vorgegeben wird.

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

5.7.2 Anwendungsbedingungen

Das Kommando SICCT SELECT CT MODE wird in den Betriebsmodi BCS und SICCT verarbeitet, um zwischen den Modi ('Aktiver Betriebsmodus') wechseln zu können.

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

Das Kommando verfügt nicht über eine Reset- oder Neustartfunktion und kann zu beliebigem Zeitpunkt an das Terminal gesendet werden. Folgt dem Befehl kein SICCT RESET Kommando, bleibt der zuvor aktive Betriebsmodus unverändert.

Ausnahmebehandlung

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

Abbruchbedingung

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

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

5.7.3 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 2 (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(s) referenced by Functional Unit Index Data Object (FUI DO) or Functional Unit Context Data Object (FU CON DO). One FUI DO or one FU CON DO contained within Command Data Field.
		Direct Coding (mandatory)	
		'00'	Address Cardterminal
		other values RFU	

P2	Command Qualifier		
	bit8..bit1	'00'	Select BCS Mode
		'01'	Select SICCT Mode
		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	
		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.
	FU CON DO		

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

5.7.4 Data Objects

Das SICCT SELECT CT MODE Kommando arbeitet optional mit dem Functional Unit Index Data Object, sofern Referenced Coding für den Parameter P1 gewählt wird.

Data Object	COMMAND RESPONSE	Desription	Remarks
FUI DO	CMD	In case of Referenced Coding: Functional Unit Index Data Object referencing the cardterminal: '84020000'	see 5.5.10
FU CON DO	CMD		see 5.5.10

5.7.5 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

5.7.6 Status-Codes SW1-SW2

Der Ausgang des Kommandos wird über das SW1-SW2-Statuswort an den Aufrufer signalisiert. Der Wert '90' in SW1 signalisiert den Erfolgsfall. Alle anderen SW1-Werte signalisieren eine Warnung oder einen Fehlerzustand. SW2 qualifiziert den Zustand weiter.

SW1-SW2	Addressed Functional Unit P1	Specification	Meaning
'6200'	CT	Warning - Specified Mode already set.	The specified CT mode had already been set.
'6700'		Wrong (command) length	Wrong Lc: Inconsistent command body or no Command Data field supported.
'6A00'		Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ Invalid FUI DO referenced by P1 (command data) ▪ P2 specifies not supported value
'6C00'		Wrong Length Le	Wrong Le: Le exists, but is not allowed to.
'9000'		Mode Selection was successful	The specified CT mode has been selected. In order to activate the mode the cardterminal has to be reset.

5.8 Command SICCT CONTROL COMMAND

SICCT CONTROL COMMAND dient der Steuerung der Kommandobearbeitung im Terminal. Es richtet sich daher an das Modul Kommandoverwaltung im Terminal.
(Note: SICCT V1.0, V1.03: The command was named "SICCT TERMINATE COMMAND".)

SICCT CONTROL COMMAND hat zwei Varianten:

- Es erteilt Auskunft über ein zuvor gesendetes Kommando ("GET STATUS")
- Es kann ein an das Terminal bereits gesendetes Kommando zur Laufzeit abbrechen oder beenden. Ein Kommando, das sich noch in einem Wartestatus vor der Ausführung befindet (Queue), wird aus dem Speicher gelöscht ("TERMINATE").

5.8.1 Funktion

Fehlerfreier Betrieb
Das Kommando SICCT CONTROL COMMAND richtet sich an den Kommandoverwalter und übergibt diesem die Kommandosequenznummer des zu adressierenden Kommandos. Im Fall, dass der Kommandverwalter die Kommandosequenznummer auffindet, teilt er dem Aufrufer den Status der Operation mit. Für die Variante "TERMINATE" unterbindet oder beendet er die Ausführung des Kommandos zusätzlich.

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

5.8.2 Anwendungsbedingungen

Das Kommando SICCT CONTROL COMMAND wird im Betriebsmodus BCS nicht unterstützt.

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			
BCS	SICCT				Stage		
					1	2	3
no	✓	✓	✓	no	-	-	-

5.8.3 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 2 (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(s) referenced by Functional Unit Index Data Object (FUI DO) or Functional Unit Context Data Object (FU CON DO). One FUI DO or one FU CON 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	

Data	Command Data		
	In case of Direct Coding of 'P1' (mandatory) and 'P2' <> '0E'		
	SEQNO DO	see **5.5.10.24**	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.
	FU CON DO		
	In case of Referenced Coding of 'P1' and 'P2' <> '0E'		
	SEQNO DO	see **5.5.10.24**	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

5.8.4 Data Objects

Das SICCT CONTROL COMMAND Kommando arbeitet optional mit dem Functional Unit Index Data Object, sofern Referenced Coding für den Parameter P1 gewählt wird. Desweiteren benötigt das Kommando die Angabe der eindeutigen Kommandosequenznummer (außer P2='0E').

Data Object	COMMAND RESPONSE	Desription	Remarks
FUI DO	CMD	In case of Referenced Coding:	see **5.5.10**
FU CON DO	CMD	Functional Unit Index Data Object referencing the cardterminal: '84020000'	
SEQNO DO	CMD	Sequence Number Data Object Sequence number of comamnd to terminate.	see **5.5.10.24**

5.8.5 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

5.8.6 Status-Codes SW1-SW2

Der Ausgang des Kommandos wird über das SW1-SW2-Statuswort an den Aufrufer signalisiert. Der Wert '90' in SW1 signalisiert den Erfolgsfall. Alle anderen SW1-Werte signalisieren eine Warnung oder einen Fehlerzustand. SW2 qualifiziert den Zustand weiter.

SW1-SW2	Addressed Functional Unit P1	Specification	Meaning
'6200'	CT	Warning - Specified command not found	The specified command could not be found.
'64xx'	CT	Warning : Command not terminated.	The ct could not terminate the specified command.
'6401'			Not terminated; command currently at stage 1
'6402'			Not terminated; command currently at stage 2
'6403'			Not terminated; command currently at stage 3
´6700´	CT	Wrong (command) length	Wrong Lc: Inconsistent command body or no Command Data field supported.
´6A00´	CT	Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ Invalid FUI DO referenced by P1 (command data) ▪ P2 specifies not supported value
´6C00´	CT	Wrong Length Le	Wrong Le: Le exists, but is not allowed to.
'90xx'	CT	Operation was successful	The specified command with the given sequence number (not P2='0E') is in actual stage 'xx' – or has been terminated at stage 'xx'
´9000´			Command status: prior to execution (queue) - or command discarded before execution – or all commands discarded before execution
'9001'			Command status: preprocessing - or command terminated at preprocessing phase
'9002'			Command status: processing - or command terminated at processing phase
'9003'			Command status: postprocessing - or command terminated at postprocessing phase
'900F'			Command abortion forced
'90FF'			Command status unknown - or command terminated at unknown stage

5.9 Command SICCT INIT CT SESSION

Sofern das SICCT-Terminal in der Betriebsart 'SICCT' arbeitet, erfolgt der Zugriff auf das Kartenterminal stets rollenbasiert im Rahmen einer Cardterminal Session (CT Session). Bevor Kommandos im Mode ‚SICCT' vom SICCT Kartenterminal verarbeitet werden können, ist eine CT Session zu öffnen. Eine CT Session besteht bis ein korrespondierendes Kommando SICCT CLOSE CT SESSION die geöffnete Session wieder schließt.

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

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

Je nachdem ob die Login-Informationen eine Benutzerkennung und Authentifizierungsdaten (Password-Phase) enthält, differenziert das SICCT Terminal den Zugriff durch

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

Das Terminal prüft die Login-Informationen gegen zuvor administrierte Zugangsdaten. Es obliegt der Terminalkonfiguration, ob ein anonymer Zugriff, d.h. ohne Angabe von Benutzer und Passwortphrase erfolgen darf.
Stereotyp zum Verhalten des Kommandos SICCT RESET CT werden zum Zeitpunkt der CT-Session-Eröffnung alle Functional Units (FU) zurückgesetzt bzw. Kartenslots sind sind deaktiviert. Ein optional vorhandenes Display zeigt nach der Ausführung die Kartenterminal Idle-Message.

5.9.1 Funktion

Fehlerfreier Betrieb

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

- 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 prüft den Zugang und ordnet anhand zuvor hinterlegter Zugangsdaten eine Benutzerrolle (SICCT-Administrator oder SICCT-Benutzer) und eine Session-ID zu, und sendet dem Aufrufer ebenfalls ein CT Session Data Object zurück,

- 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 identifcation data (password) ▪ check and grant authorisation ▪ open ct session ▪
postprocessing phase	Nachbereitungsphase	3		▪ send return value

5.9.2 Anwendungsbedingungen

Das Kommando SICCT INIT CT SESSION richtet sich an den SICCT-Kommandointerpreter über eine bestehendene Verbindung. Es wird angenommen, dass die zur Prüfung der Login Information benötigten Konfigurationsdaten im Kartenterminal vorliegen.

Nach der Kommandoausführung sind alle Functional Units im Reset-Zustand bzw. deaktiviert. Ein optional vorhandenes Display zeigt nach der Ausführung die Kartenterminal Idle-Message.

Das Kommando SICCT INIT CT SESSION wird im Betriebsmodus BCS nicht unterstützt.

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

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

5.9.3 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(s) referenced by Functional Unit Index Data Object (FUI DO) or Functional Unit Context Data Object (FU CON DO). One FUI DO or one FU CON 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)						
	CT Session DO	Cardterminal 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.			
	FU CON DO						
	CT Session DO	Cardterminal 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:

5.9.4 Data Objects

Das SICCT INIT CT SESSION Kommando arbeitet optional mit dem Functional Unit Index Data Object, sofern Referenced Coding für den Parameter P1 gewählt wird.

Data Object	COMMAND RESPONSE	Desription	Remarks
FUI DO	CMD	In case of Referenced Coding: Functional Unit Index Data Object referencing the cardterminal: '84020000'	see 5.5.10
FU CON DO	CMD		see 5.5.10
CTSESS DO	CMD	Datenobjekt CT Session Data Object	see 5.5.10
	RSP		

5.9.5 Response Structure

SICCT INIT CT SESSION	Kodierung R-APDU			
	[Body:]		**Trailer**	
	[Requested Data / Information]		Status Byte 1	Status Byte 2
	CT Session DO	Returns Username and Session-ID.	SW1	SW2

5.9.6 Status-Codes SW1-SW2

Der Ausgang des Kommandos wird über das SW1-SW2-Statuswort an den Aufrufer signalisiert. Der Wert '90' in SW1 signalisiert den Erfolgsfall. Alle anderen SW1-Werte signalisieren eine Warnung oder einen Fehlerzustand. SW2 qualifiziert den Zustand weiter.

SW1-SW2	Addressed Functional Unit P1	Specification	Meaning
'6200'	CT	Warning - CT Session already set.	CT session already set..
'6400'		Error - Opening CT Session was not successful	Error
			Invalid Username
			Invalid Password.
'6700'		Wrong (command) length	Wrong Lc: Inconsistent command body or no Command Data field supported.
'6900'		Command not allowed.	Open CT Session found.
'6A00'		Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ Invalid FUI DO referenced by P1 (command data) ▪ P2 specifies not supported value
'6C00'		Wrong Length Le	Wrong Le:
'9000'		Opening CT Session was successful	The specified CT session has been opened.

5.10 Command SICCT CLOSE CT SESSION

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

5.10.1 Funktion

Fehlerfreier Betrieb
Der Aufrufer übermittelt mit dem SICCT INIT CT SESSION Kommando ein Datenobjekt CT Session Data Object. Der SICCT-Kommandointerpreter prüft die Gültigkeit der Session-ID und schließt die Cardterminal Session.

Nach der Kommandoausführung sind alle Functional Units im Reset-Zustand bzw. deaktiviert. Ein optional vorhandenes Display zeigt nach der Ausführung die Kartenterminal Idle-Message.

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

5.10.2 Anwendungsbedingungen

Das Kommando SICCT CLOSE CT SESSION richtet sich an den SICCT-Kommando-interpreter über eine bestehendene Verbindung.

Das Kommando SICCT CLOSE CT SESSION wird im Betriebsmodus BCS nicht unterstützt.

Abbruchbedingung
Das Kommando SICCT CLOSE CT SESSION kann selbst nicht beeinflußt oder abge-brochen werden.

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

5.10.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	[Lc]	[Data]	[Le]
SICCT CLOSE	'80'	'29'	Functional Unit	'00'	Length Command Data	Command Data	absent

CT SESSION	▪ CLA = Class ▪ INS = Instruction ▪ P1, P2 = Parameter 1 and 2 ▪ Lc = Length of command data field ▪ Le = Length of expected data	Case 2 (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(s) referenced by Functional Unit Index Data Object (FUI DO) or Functional Unit Context Data Object (FU CON DO). One FUI DO or one FU CON 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)					
	CT Session DO	Cardterminal Session Data Object				
		'69'	L	Value		
			'13'	00	absent	Username
			'13'	00	absent	Password
			'13'	Len	<1 ... 12 Byte>	Session ID
	In case of Referenced Coding of 'P1'					
	FUI DO	'84020000'			Functional Unit Index Data Object referencing the cardterminal.	
	FU CON DO					
	CT Session 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

5.10.4 Data Objects

Das SICCT CLOSE CT SESSION Kommando arbeitet optional mit dem Functional Unit Index Data Object, sofern Referenced Coding für den Parameter P1 gewählt wird.

Data Object	COMMAND RESPONSE	Desription	Remarks
FUI DO	CMD	In case of Referenced Coding: Functional Unit Index Data Object referencing the cardterminal: '84020000'	see 5.5.10
CT SESSION DO	CMD	Datenobjekt CT Session Data Object	see 5.5.10

5.10.5 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

5.10.6 Status-Codes SW1-SW2

Der Ausgang des Kommandos wird über das SW1-SW2-Statuswort an den Aufrufer signalisiert. Der Wert '90' in SW1 signalisiert den Erfolgsfall. Alle anderen SW1-Werte signalisieren eine Warnung oder einen Fehlerzustand. SW2 qualifiziert den Zustand weiter.

SW1-SW2	Addressed Functional Unit P1	Specification	Meaning
'6200'	CT	Warning - No pending / open CT Session found.	CT session already closed.
'6400		Error - Closing CT Session was not successful	Error
'6403'			Invalid Session-ID
'6700'		Wrong (command) length	Wrong Lc: Inconsistent command body or no Command Data field supported.
'6900'		Command not allowed.	No open CT Session
'6A00'		Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ Invalid FUI DO referenced by P1 (command data) ▪ P2 specifies not supported value
'6C00'		Wrong Length Le	Wrong Le:
'9000'		Closing CT Session was successful	The specified CT session has been closed.

5.11 Command SICCT RESET CT / ICC

Das Kommando SICCT RESET dient dem selektiven Rücksetzen (Reset) einer vorhandenen Chipkarten bzw. einer Chipkartenfunktionseinheit oder alternativ des gesamten Kartenterminals mit allen verfügbaren Funktionseinheiten. Das Rücksetzen erfolgt elektrisch und / oder logisch, um das Kartenterminal und / oder Chipkarten (-Funktionseinheiten) in einen definierten initialen Zustand zu setzen.

5.11.1 Funktion

Je nach adressierter Functional Unit, Kartenterminal oder Chipkartenkontaktiereinheit (Slot), stellt die Ausführung des RESET Kommandos einen entprechend unterschiedlichen Grundzustand her. Der hauptsächliche Unterschied der Grundzustände ist

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

Das Kommando adressiert das Kartenterminal

Ein Reset des Kartenterminals bedeutet allgemein ein Rücksetzen aller Kartenterminalressourcen sowie eine Neuinitialisierung des internen Kommunikationsmoduls, der internen Zustands- und Steuerlogik, der Protokolllogik, der Interaktions- und Anzeigeelemente wie z.B. KeyPad, LCDisplay, Statusanzeigen (LED), Sensoren.
Nach Ausführung des Kartenterminal-Resets sind alle Chipkartenfunktionseinheiten bzw. im Wirkbereich befindlichen Chipkarten elektrisch / logisch deaktiviert.

LCDisplay

Sofern ein Display vorhanden ist, erscheint nach Ausführung des Reset CT Kommandos die Idle Message des Kartenterminals im Display.

Warm / Cold Reset

Das Kartenterminal differenziert die Anforderung nicht, und führt immer einen Cold Reset aus, der alle Functional Units deaktiviert bzw. in deren Grundzustand überführt.

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

Adressiert das RESET-Kommando mit PPSR-DO das Terminal, bleiben vorgegebene PPS-Parameter unberücksichtigt, das Kommando wird abgewiesen und mit der Rückgabe eines Fehlerstatus beendet. Es wird kein (leeres) PPSR-Datenbojekt zurückgegeben.

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 Wirkbereich beinhalten, bedeutet die Parametrierung der Chipkartenkontaktiereinheit sowie nachfolgend den elektrisch / logischen Reset der im Zugriff befindlichen Chipkarte(n). Nach erfolgreicher Ausführung ist eine Chipcard im Wirkbereich aktiviert und betriebsbereit für den Empfang von Kommandos (APDUs) im Reset-Zustand.

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

Warm / Cold Reset

Das CT-Kommando bewirkt (über den Parameter P2) wahlweise einen Warm - oder Cold - Reset an einer ICC - Funktionseinheit. Für kontaktbehaftete Karten nach ISO 7816-3 (ICC) oder nach ISO 14443 A / B für kontaktlosen Karten (PICC).

Während ein Kaltstart immer und bei deaktivierter Chipcard erfolgen kann, wird ein Warmstart nur ausgeführt, sofern die Chipcard bereits aktiviert und betriebsbereit war.

Rückgabe von Reset-Information
Es wird generell erwartet, dass ein Chipkarten-Reset bei kontaktbehafteten Karten mit einer Answer-To-Reset-Information und bei kontaktlosen Karten mit einem Answer-To-Select (ATS) beantwortet wird.

Automatisches und explizites PPS-Verfahren
Das SICCT Kartenterminal wertet generell die ATR / ATS - Informationen einer zurückgesetzten Chipkarte aus, um festzustellen, ob eine Protokoll Parameter Selection (PPS) erfolgen kann.

Eine PPS-Negotiation kann vom Terminal generell nicht durchgefürt 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, ISO 14443 B, Prozessorkarte nach ISO 7816-3 mit specific mode, unbekannter Kartentyp).

Im Gutfall führt das Terminal automatisch ein implizites Protocol Parameter Selection (PPS)-Verfahren nach ISO 7816-3 durch, 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 wird 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.

Übergabe eines Parameter-Request- / Rückgabe eine PPS Response Datenobjekts
Sofern ein SICCT RESET Kommando mit PPSR-DO ausgeführt wird, verwendet der PPS-Mechanismus des Kartenterminals diese Parameter für die Aushandlung der Protokollparameter (Protocol Negotiation), unmittelbar 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 entsprechend der adressierten Kartentechnologie vorzunehmen:

- ISO 7816-3 für Prozessorchipkarten
- ISO 14443 A für kontaktlose Karten / Token

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

Rückgabe des Status
Das CT-Kommando liefert den Kommandostatus (SW1SW2) sowie auf Anforderung des Aufrufers die angeforderten Information (Datenobjekte) der zurückgesetzten Chipkarte(n-kontaktiereinheit) oder des Kartenterminals. Sofern ein Fehlerstatus angezeigt wird, werden keine Datenobjekte zurückgeliefert.

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)

5.11.2 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 eine Grundzustand des gesamten Geräts herzustellen.
- vor Beendigung der Kartenterminalsession verwendet werden, um den Grundzustand des gesamten Geräts herzustellen.

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 ist elementar 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			
BCS	SICCT				Stage		
					1	2	3
✓	✓	✓	✓	no	-	-	-

5.11.3 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 (cmd data, no rsp data:) Lc=1-255 Bytes no Le		
					Case 3 (no cmd data, rsp data): no Lc, Le=1-256 Bytes		
					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(s) referenced by Functional Unit Index Data Object (FUI DO) or Functional Unit Context Data Object (FU CON DO). 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 or RFID Antenna	
		'x'	x <> '0', '1' : other values RFU	
	bit4 .. bit1	'0'	bit 8 .. bit 5 = '0'	Cardterminal (CT)
			bit 8 .. bit 5 = '1'	RFID Antenna Unit
		'1' : 'E'	bit 8 .. bit 5 = '0'	1st ICC-Interface ... 14th ICC-Interface
			bit 8 .. bit 5 = '1'	1st RFID-Token ... 14th RFID-Token

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 the RFID Antenna	
		0	Cold Reset of RFID Antenna
		1	RFU
		In case P1 addresses an ICC , RFID-Slot	
		0	Cold Reset of ICC, PICC
		1	Warm Reset of ICC, PICC
	bit7.. bit6	Protocol Parameter Selection Control	
		In case P1 addresses an ICC , RFID-Slot	
		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 , RFID-Slot	
		0	No requested data

		1	Request ICC Interface Status Data Object
	bit4 .. bit1	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 the RFID Antenna	
		'1'	Return complete RFID Antenna Status Data Object
		In case P1 addresses an ICC , RFID-Slot	
		'1'	Request complete ATR Data Object(s)
		'2'	Request Historical Byte Data Object(s) (of ATR information)
		'3'	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 bytescontained 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 Lc = '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
	FU CON DO	
	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:

5.11.4 Data Objects

Das SICCT RESET CT ICC Kommando arbeitet mit den folgenden Daten Objekten.

Data Object	COMMAND RESPONSE	optional	Desription	Remarks
FUI DO	CMD	✓	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10.9
FU CON DO	CMD	✓		
ATR DO	RESP	✓	Answer To Reset Information of a Chipcard	see 5.5.10.4

Data Object	COMMAND RESPONSE	optional	Desription		Remarks
HB DO	RESP	✓	Historical Byte Data Object of an ATR of a Chipcard		see 5.5.10.5
CTS DO	RESP	✓	Cardterminal Status Information	full Information	see 5.5.10.13
				reduced Information	
INTFS DO	RESP	✓	ICC Interface Status Data Object		see 5.5.10.15
PPSR DO	CMD	✓	PPS - Request Data Object		see 5.5.10.18
	RESP	✓	PPS - Response Data Object		

5.11.5 Response Structure

	Kodierung R-APDU			
	[Body:]		**Trailer**	
	[Requested Data / Information]		Status Byte 1	Status Byte 2
SICCT RESET CT / ICC	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		

5.11.6 Status-Codes SW1-SW2

Nach erfolgreicher Ausführung des RESET-Kommandos enthält das erste Byte des Statusworts SW1 den Wert '90'.
Das Statusbyte SW2 qualifiziert weitere Umstände in Abhängigkeit zu Kommandoparametern.
Dieser Wert Sw2 läßt eine Typerkennung der Chipkarte nach dem Reset zu:

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

SW1-SW2	Addressed Functional Unit			Specification	Meaning
	P1				
	Of Type				
'6400'	CT	-	-	Reset not successful	Error occured during cardterminal reset. State of terminal and functional units possibly unchanged.

	-	ICC<n>	RFID<n>	Reset not successful	Error occured during chipcard reset. State of addresed chipcard set to deactived state.
'64A1'	-	ICC<n>	RFID<n>	No Card present	Addressed ICC Interface / slot did not contain or contact a chipcard / RFID token.
'6700'	CT	ICC<n>	RFID<n>	Wrong (command) length	Wrong Lc: Inconsistent command body or no Command Data field supported.
'6A00'	CT	ICC<n>	RFID<n>	Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ P2 specifies not supported value
'6C00'	CT	ICC<n>	RFID<n>	Wrong Length Le	
'6F00'	CT	-	-	Communication with CT not possible	
	-	ICC<n>	RFID<n>	Communication with ICC not possible	
'9000'	CT	-	-	Reset successful	Cardterminal and all functional units successfully set to reset state. All chipcards deactivated.
	-	ICC<n>	RFID<n>	Reset successful, synchronous ICC	Synchronous chipcard detected, activated and successfully set to demanded reset state.
'9001'	-	ICC<n>	RFID<n>	Reset successful, asynchronous ICC	Asynchronous chipcard detected, activated and successfully set to demanded reset state.

5.12 Command SICCT REQUEST ICC

Das Kommando SICCT REQUEST ICC dient der Anforderung und Aktivierung einer Chipkarte an einer adressierten Chipkartenkartenfunktionseinheit. Das Kommando bewirkt stets einen Cold -Reset der Chipkarte und liefert mit der Antwort Informationen über den Typ der Chipkarte sowie den generellen Status zurück.

5.12.1 Funktion

Für den Fall, dass sich keine aktivierte Chipkarte im Wirkbereich 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 erfolgt automatisch ein COLD-Reset sowie eine Aktivierung der angeforderten Chipkarte. Das mit dem Einführungsprozess verbundene Rücksetzen der Chipkarte bzw. der Chip-kartenfunktionseinheit erfolgt elektrisch und / oder logisch, um die Chipkarten - (funktionseinheit) in einen definierten Initialzustand zu versetzten. Optional kann die Reset-Information der Chipkarte zurückgegeben werden.

Für den Fall, dass sich eine bereits zuvor aktivierte Chipkarte im Wirkbereich der Chipkartenfunktionseinheit befindet, führt das Kommando keinen COLD - Reset aus und liefert keine Typinformation. Der elektrisch / logische Zustand der Chipkarte bleibt unverändert. Ein gesonderter Fehlerkode informiert den Aufrufer über diese Situation. Die ggf. angeforderte und im Terminal vorliegende Reset-Information kann angefordert werden.

Folgende Optionen sind generell wählbar:

- Anzeige eines Bedienerdialogs : Bei Kartenterminals mit Display kann eine Standard- oder frei wählbare Eingabeanforderung als Bedienerdialog angezeigt werden.

- Setzen eines optischen oder akustischen Signals: Bei Kartenterminals mit akustischem Tongeber oder optischer Anzeige (z.B. LED) kann ein Signal als Eingabeanforderung ausgelöst werden.

- die Zeit für die Einführung und / oder Aktivierung der Chipkarte kann getrennt vorgegeben und überwacht werden;

Das Kommando liefert nach Beendigung, d.h. nach erfolgtem Reset der Chipkarte, folgende Informationen zurück

- Typ der aktivierten Chipkarte
  (Synchrone / asynchrone Karte bzw. Speicher- oder Prozessorkarte),

- Kennung der Chipkarte: Answer-To-Reset-String (Cold ATR) bzw. den Answer-To-Select-String (ATS) der aktivierten Chipkarte,

- aktives Protokoll der aktivierten Chipkarte

- logischer Betriebsmodus der aktivierten Chipkarte
  (specific / negotiable).

- elektrischer Status der aktivierten Chipkarte
  (*defekt, falsche Lage, nicht kommunikationsfähig, ..*).

Der Reset von Chipkartenkontaktiereinheiten, welche eine Chipkarte in ihrem Wirkbereich beinhalten, bedeutet die Parametrierung der Chipkartenkontaktiereinheit sowie nachfolgend den elektrisch / logischen Reset der im Zugriff befindlichen Chipkarte(n). Das CT-Kommando bewirkt stets einen Cold - Reset an einer ICC - Funktionseinheit.

Es wird generell erwartet, dass ein Chipkarten-Reset bei kontaktbehafteten Karten mit einer Answer-To-Reset-Information und bei kontaktlosen Karten mit einem Answer-To-Select (ATS) beantwortet wird. Das SICCT Kartenterminal wertet generell die ATR / ATS - Informationen einer zurückgesetzten Chipkarte aus, und führt generell ein implizites Protocol Parameter Selection (PPS)-Verfahren nach ISO 7816-3 durch.

Das CT-Kommando liefert den Kommandostatus sowie auf Anforderung des Aufrufers die korrespondierende Reset-Information der zurückgesetzten Chipkarte(nkontaktiereinheit) oder des Kartenterminals.

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

5.12.2 Anwendungsbedingungen

Das Kommando eröffnet generell die Kommunikation und markiert den Beginn einer Session zu einer angeforderten Chipkarte.

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

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

Abbruchbedingung
Das Kommando SICCT REQUEST ICC hat nur dann eine direkt erkennbare Auswirkung auf den Benutzer, sofern Statuswechsel an der Mensch-Maschine-Schnittstelle entstehen. Die Anforderung, dem System eine Karte zuzuführen, kann durch eine entsprechende Interaktion des Anwenders des Kartenterminals beeinflusst werden. Diese Interaktionen sind: Zuführen der Karte, Betätigen der Abbruchtaste am Keypad des Kartenterminals.
Sobald eine angeforderte Karte zugeführt wurde, wird diese zurückgesetzt. Dieser Vorgang kann nicht durch den Anwender beeinflußt werden.

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

SICCT REQUEST ICC							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable			
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 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 (cmd data, no rsp data:) Lc=1-255 Bytes no Le		
					Case 3 (no cmd data, rsp data): no Lc, Le=1-256 bytes		
					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). 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 (RFID antenna)	
		'x'	x <> '0', '1' : other values RFU	
	bit4 .. bit1	'1' : 'E'	bit 8 .. bit 5 = '0'	Contact bound Chipcard Interface
			bit 8 .. bit 5 = '1'	Contactless Chipcard Interface (RFID Antenna Unit)
			bit 8 .. bit 5 = '0'	1st ICC-Interface ... 14th ICC-Interface
			bit 8 .. bit 5 = '1'	1st RFID-Token ... 14th RFID-Token

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 Application Label DO
		'F'	no message to be displayed
	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)

Lc	Length of Command Data Nc		
	Empty	non-existent: no Command Data provided; interpret Lc = '00'	
	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 Lc = '00': no Command Data provided
	APPLICATION LABEL DO	Application Label Data Object

	SICCT Message To Be Displayed DO	SICCT Message-To-Be-Displayed Data Object
	WAIT TIME DO	Waiting Time Data Object
	In case of Referenced Coding of 'P1'	
	FUI DO	Functional Unit Index Data Object
	FU CON DO	
	APPLICATION LABEL DO	Application Label Data Object
	SICCT Message To Be Displayed DO	SICCT Message-To-Be-Displayed Data Object
	WAIT TIME 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

5.12.4 Data Objects

Das SICCT REQUEST Kommando arbeitet mit den folgenden Daten Objekten.

Data Object	COMMAND RESPONSE	Desription	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10
FU CON DO	CMD	Functional Unit Context Data Object	
APPLICATION LABEL DO	CMD	Text / display message.	
SICCT Message To Be Displayed DO	CMD	Constructed TLV-DO containing one Character Set and one Application Label DO.	
ATR DO	RESP	Answer To Reset Information (Cold ATR)	
ATS DO	RESP	Answer To Select Information	
WAIT TIME DO	CMD	Max. Waiting Time in seconds	

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		

5.12.6 Status-Codes SW1-SW2

Nach erfolgreicher Ausführung des REQUEST ICC Kommandos enthält das erste Byte des Statusworts SW1 den Wert '90'.
Das Statusbyte SW2 qualifiziert weitere Umstände in Abhängigkeit zu Kommandoparametern.
Dieser Wert Sw2 läßt eine Typerkennung der Chipkarte nach dem Reset zu:

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

SW1-SW2	Addressed Functional Unit		Specification	Meaning
	P1			
	Of Type			
'6200'	ICC<n>	RFID<n>	Warning: No card presented in time	No card presented within specified time.
'6201'			Warning: Reset successfu	ICC already present and activated
'6400'			Reset not successful	Error occured during chipcard reset. State of addresed chipcard set to deactived state.
'6401'			Process aborted by pressing of CANCEL key	Process aborted by pressing of CANCEL key.
'6700'			Wrong (command) length	Wrong Lc: Inconsistent command body or no Command Data field supported.
'6930'			Command with timer not supported.	Terminal does not support the timer option.
'6940'			Command with Display not supported.	Command with Display not supported.
'6941'			Functional Unit (FU) busy / not available.	The addressed FU (display, slot, ..) is busy at the moment or not available.
'6942'			Selected Character Set not supported.	The addressed display does not support the selected character set.
'6A00'			Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ P2 specifies not supported value
'6C00'			Wrong (information) length parameter	Wrong Le.
'6F00'			Communication with ICC not possible	
'9000'			Reset successful, synchronous ICC	Synchronous chipcard detected, activated and successfully set to demanded reset state.
'9001'			Reset successful, asynchronous ICC	Asynchronous chipcard detected, activated and successfully set to demanded reset state.

5.13 Command SICCT EJECT ICC

5.13.1 Funktion

Das Kommando SICCT EJECT ICC dient zur elektrisch / logischen Deaktivierung einer adressierten Chipkartenkartenfunktionseinheit bzw. einer Chipkarte und veranlaßt die Ausführung optionaler Zusatzfunktionen.

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

5.13.2 Anwendungsbedingungen

Das Kommando beendet generell die Kommunikation und markiert das Ende einer Session zu einer Chipkarte, indem die Chipkarte elektrisch / logisch deaktiviert wird.

Folgende Optionen sind generell wählbar:

- Setzen eines akustischen Signals
- Setzen eines optischen Signals
  (Bei 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 dient primär zur Deaktivierung und Entnahme einer Chipkarte. Die Ausgabe einer Display-Nachricht entfällt, wenn der adressierte Slot bereits keine gesteckte Karte beinhaltet.

Das Kommando liefert mit der Antwort Informationen über den Zustand der Kontaktiereinheit bzw. der Chipkarte zurück.

Abbruchbedingungen
Die Anforderung vom System, eine Karte zu deaktivieren, kann nicht durch eine entsprechende Interaktion des Anwenders des Kartenterminals beeinflusst werden.
Diese Interaktionen zur Entnahme kann per Betätigung der Abbruchtaste am Keypad des Kartenterminals abgebrochen werden.

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

5.13.3 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 2 (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). 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 (RFID antenna)	
		'x'	x <> '0', '1' : other values RFU	
	bit4 .. bit1	'1' : 'E'	bit 8 .. bit 5 = '0'	Contact bound Chipcard Interface
			bit 8 .. bit 5 = '1'	Contactless Chipcard Interface (RFID Antenna Unit)
			bit 8 .. bit 5 = '0'	1st ICC-Interface ... 14th ICC-Interface
			bit 8 .. bit 5 = '1'	1st RFID-Token ... 14th RFID-Token

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 Application Label DO
		'F'	no message to be displayed
	Option Setting		
	bit1	'0'	(RFU)
	bit2	'0'	Delivery: Mechanical Throwout
		'1'	Delivery: No Mechanical Throwout - Keep chipcard
	bit3	'0'	Optical Signal: None
		'1'	Optical Signal: Show Optical Signal
	bit4	'0'	Acoustic Signal: None
		'1'	Acoustic Signal: Give Acoustic Signal

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 Lc = '00': no Command Data provided
	APPLICATION LABEL DO	Application Label Data Object
	SICCT Message To Be Displayed DO	SICCT Message-To-Be-Displayed Data Object
	WAIT TIME DO	Waiting Time Data Object
	In case of Referenced Coding of 'P1'	
	FUI DO	Functional Unit Index Data Object
	APPLICATION LABEL DO	Application Label Data Object
	SICCT Message To Be Displayed DO	SICCT Message-To-Be-Displayed Data Object
	WAIT TIME 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 EJECT Kommando arbeitet mit den folgenden Daten Objekten.

Data Object	COMMAND RESPONSE	Desription	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10
FU CON DO	CMD	Functional Unit Context Data Object	
APPLICATION LABEL DO	CMD	Text / display message.	
SICCT Message To Be Displayed DO	CMD	Constructed TLV-DO containing one Character Set and one Application Label DO.	
WAIT TIME DO	CMD	Max. Waiting Time in seconds	

5.13.5 Response Structure

	Kodierung R-APDU			
	[Body:]		**Trailer**	
	[Requested Data / Information]		Status Byte 1	Status Byte 2
SICCT EJECT ICC	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 SW1-SW2

Nach erfolgreicher Ausführung des EJECT ICC Kommandos enthält das erste Byte des Statusworts SW1 den Wert '90'.
Das Statusbyte SW2 qualifiziert weitere Umstände in Abhängigkeit zu Kommandoparametern.
Dieser Wert Sw2 läßt eine Statuserkennung der Chipkartenfunktionseinheit nach dem Deaktivieren zu:

- '01' Command Successful, card removed

SW1-SW2	Addressed Functional Unit P1 Of Type		Specification	Meaning
'6200'	ICC<n>	RFID<n>	Warning: Card not removed within specified time	Chipcard deactivated but not removed.
'6700'			Wrong (command) length parameter	Wrong Lc: Inconsistent command body or no Command Data field supported.
'6930'			Command with timer not supported.	Terminal does not support the timer option.
'6940'			Command with Display not supported.	Command with Display not supported.
'6941'			Functional Unit (FU) busy / not available.	The addressed FU (display, slot, ..) is busy at the moment or not available.
'6942'			Selected Character Set not supported.	The addressed display does not support the selected character set.
'6A00'			Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ P2 specifies not supported value
'6C00'			Wrong (information) length parameter	Wrong Le.
'9000'			Command successful	Chipcard deactivated but not removed.
'9001'			Command successful	Chipcard deactivated and removed.

5.14 Command SICCT GET STATUS

5.14.1 Funktion

Das SICCT GET STATUS - Kommando erlaubt die Adressierung eines GET- Interfaces der Kartenterminals bzw. der Funktionseinheiten, Statusinformation abfragen zu können. Je nach Funktionseinheit und Typus des Datenobjekts können 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,
Functional Unit Data Object	Query ▪ Presence, ▪ Quantity ▪ State ▪ Availability of Functional Units.
Functional Unit Name DataObject	Query ▪ Friendly Name of an Functional Unit
ICC Status Data Object	Query • Presence of an ICC • Electrical State of an ICC
RFID Token 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
Sequence Number Data Object	Query ▪ processing state of a specified SICCT command

Das Kommando liefert mit der Antwort den Ausgang der Operation sowie optional angeforderte Datenobjekte zurück.

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

5.14.2 Anwendungsbedingungen

Das Auslesen von Datenobjekten über die Kommandoschnittstelle geschieht in Abhängigkeit der Bedeutung des jeweiligen Datenobjekts und der aktiven Zugriffsrechte. Zur Zeit 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 Anwender des Kartenterminals weder beeinflußt noch abgebrochen werden.

Die Wirkung des Kommandos ist elementar 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			
BCS	SICCT				Stage		
					1	2	3
✓	✓	✓	✓	no	no	no	no

5.14.3 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 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 (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(s) referenced by Functional Unit Index Data Object (FUI DO) or Functional Unit Context Data Object (FU CON DO). 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 or RFID antenna
		'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'	bit 8 .. bit 5 = '0'	Cardterminal (CT)
			bit 8 .. bit 5 = '1'	RFID Antenna Unit
			bit 8 .. bit 5 = '4'	Standard Display
			bit 8 .. bit 5 = '5'	Standard Keypad
			bit 8 .. bit 5 = '6'	Standard Printer Port
			bit 8 .. bit 5 = '7'	Standard Biometric Sensor Unit: Fingerprint
		'1'' : 'E'	bit 8 .. bit 5 = '0'	1st ICC-Interface .. 14th ICC-Interface
			bit 8 .. bit 5 = '1'	1st RFID-Token .. 14th RFID-Token
			bit 8 .. bit 5 = '4'	2nd .. 14th Display
			bit 8 .. bit 5 = '5'	2nd .. 14th Keypad
			bit 8 .. bit 5 = '6'	RFU
			bit 8 .. bit 5 = '7'	Specific Biometrc 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. 'A2' '00' for FU Name DO.
	bit8 .. bit1	Direct Coding	
		Data Objects served by all Fundtiona Units	
		'A1'	Functional Unit Name Data Object
		In case P1 addresses the Cardterminal (CT):	
		'46'	CardTerminalManufacturer Data Object
		'80'	ICC Status Data Object (all ICC Interfaces)
		'81'	Functional Unit Data Object
		'68'	Sequence Number Data Object
		In case P1 addresses an ICC , RFID-Slot	
		'80'	ICC Status Data Object (ICC Interface as addressed by 'P1')
		'66'	Interface Capabilities Data Object
		In case P1 addresses an RFID-Slot	
		'83'	RFID Token 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 RFIF Antenna Unit	
		'64'	RFID Antenna Status 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 Lc = '00': no Command Data provided
	One Requested Data Object	In case 'P2' set to 'FF':
		available data objects : see 5.14.4

	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 5.14.4

Le	Length of Requested Data Ne Return up to <Le> 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
		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

5.14.4 Data Objects

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

Data Object	COMMAND RESPONSE	Desription	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10.9
	RESP		
FU NAME DO	CMD	Functional Unit Name Data Object	see 5.5.10.11
	RESP		
CTM DO	CMD	Card Terminal Manufacturer Data Object	see 5.5.10.6
	RESP		
FU DO	CMD	Functional Unit Data Object	see 5.5.10.8
	RESP		
ICCS DO	CMD	ICC Status Data	see 5.5.10.7
	RESP		
RFIDS DO	CMD	RFID Token Status	see 5.5.10.12
	RESP		
RFIDAS DO	CMD	RFID Antenna Status Information	see 5.5.10.12
	RESP		
DSPLC DO	CMD	Display Capabilities Data Object	see 5.5.10.17
	RESP		
CS DO	CMD	Character Set Data Object	see 5.5.10.20

Data Object	COMMAND RESPONSE	Desription	Remarks
	RESP		
INTFC DO	CMD	ICC Interface Capabilities Data Object	see 5.5.10.16
	RESP		
SEQNO DO	CMD	Sequence Number and Command processing state	see 5.5.10.24
	RESP		

5.14.5 Response Structure

	Kodierung R-APDU			
	[Body:]		**Trailer**	
	[Requested Data / Information]		Status Byte 1	Status Byte 2
SICCT GET STATUS	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		

5.14.6 Status-Codes SW1-SW2

Nach erfolgreicher Ausführung des SICCT GET STATUS - Kommandos enthält das erste Byte des Statusworts SW1 den Wert '90'.
Das Statusbyte SW2 qualifiziert ggf. weitere Umstände in Abhängigkeit zu Kommandoparametern.

SW1-SW2	Addressed Functional Unit P1		Specification	Meaning
'6400'	CT	ICC<n>	Execution Error	
'64A1'			No card present	Addressed ICC Interface / slot did not contain or contact a chipcard / RFID token.
'6700'	CT		Wrong (command) length parameter	Wrong Lc: Inconsistent command body or no Command Data field supported.
'6900'	CT		Command not allowed	▪ Cardterminal Session: Admin Access Rights required. ▪ No open CT Session
'6A00'	CT		Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ P2 specifies not supported value
'6A80'			Invalid Data Object	Invalid Data Object
			Incorrect parameters in the command data field	
'6A88'			Missing Data Object	Missing Data Object

SW1-SW2	Addressed Functional Unit		Specification	Meaning
	P1			
			Referenced data or reference data not found	
'6C00'	CT		Wrong (information) length parameter	Wrong Le.
'9000'	CT		Command successful	Data Object query successful.

5.15 Command SICCT SET STATUS

5.15.1 Funktion

Das SICCT SET STATUS - Kommando erlaubt die Adressierung eines SET-Interfaces der Kartenterminals bzw. der Funktionseinheiten, Statusinformation oder Geräteparameter setzen zu können.

Über das SET - Interface des Kartenterminals bzw. einer Funktionseinheit können über Datenobjekte generelle Zustands-, Chipkarten oder Geräteparameter gesetzt werden. Das Setzen kann eine generelle Betriebsmodus. oder Zustandsänderung bewirken.

Dieses Kommando setzt das Kommunikationsprotokoll sowie Betriebsparameter an einer Kartenkontaktiereinheit oder des Kommunikationsmoduls.

Folgende Terminalkonfigurationsparameter können innerhalb einer ADMIN-CT-Session gesetzt werden:

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

Das Kommando liefert mit der Antwort den Ausgang der Operation sowie optional angeforderte Datenobjekte zurück.

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 correponding state ▪ set correponding state
postprocessing phase	Nachbereitungsphase	3		▪ send return value

5.15.2 Anwendungsbedingungen

Das Setzen von Datenobjekten über die Kommandoschnittstelle kann nur in Abhängigkeit des jeweiligen Datenobjekts und der aktiven Zugriffsrechte geschehen. Zur Zeit 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 Kartenterminals weder beeinflusst noch abgebrochen werden.

Die Wirkung des Kommandos ist elementar 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			
BCS	SICCT				Stage		
					1	2	3
no	✓	✓	no	no	no	no	no

5.15.3 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 2 (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	
	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). 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 or RFID antenna	
		'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'	bit 8 .. bit 5 = '0'	Cardterminal (CT)
			bit 8 .. bit 5 = '1'	RFID Antenna Unit
			bit 8 .. bit 5 = '4'	Standard Display
			bit 8 .. bit 5 = '5'	Standard Keypad
			bit 8 .. bit 5 = '6'	Standard Printer Port
			bit 8 .. bit 5 = '7'	Standard Biometric Sensor Unit: Fingerprint
		'1'' : 'E'	bit 8 .. bit 5 = '0'	1st ICC-Interface .. 14th ICC-Interface
			bit 8 .. bit 5 = '1'	1st RFID-Token .. 14th RFID-Token
			bit 8 .. bit 5 = '4'	2nd .. 14th additional Display
			bit 8 .. bit 5 = '5'	2nd .. 14th additional Keypad
			bit 8 .. bit 5 = '6'	RFU
			bit 8 .. bit 5 = '7'	Specific Biometrc Sensor Unit Type

P2	Command Qualifier: indicates Data Object to be set		
	bit8 .. bit1	Referenced Coding	
		'FF'	Escape : Signal Referenced Coding of 'P2': Requested Data Object contained within Command Data Field.
	bit8 .. bit1	Direct Coding	
		In case P1 addresses the Cardterminal (CT):	
		In case P1 addresses an ICC , RFID-Slot	
		In case P1 addresses a Display FU	
		'85'	Character Set 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 '000000' <= Lc <= '00FFFF'	0 <= Nc <= 65535

Data	Command Data
	In case of Direct Coding of 'P1' (mandatory)

	Empty	non-existent : in case Lc = '00': no Command Data provided
	One Data Object with values to be set	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 Data Object with values to be set	In Case 'P2' set to 'FF'
		available data objects see 5.15.4

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

5.15.4 Data Objects

Das SICCT SET STATUS Kommando arbeitet mit den folgenden Daten Objekten. Zur Zeit sind in der Definition des SICCT-Standards noch nicht alle Datenobjekte und Zugriffsmöglichkeiten definiert.

Data Object	COMMAND RESPONSE	Desription	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.9
FU Name DO	CMD	Functional Unit Name Data Object	see 5.5.10.11

5.15.5 Response Structure

<table>
<tr><td rowspan="8">SICCT SET STATUS</td><td colspan="3">Kodierung R-APDU</td></tr>
<tr><td colspan="2">[Body:]</td><td>Trailer</td></tr>
<tr><td colspan="2">[Requested Data / Information]</td><td>Status Byte 1</td><td>Status Byte 2</td></tr>
<tr><td rowspan="5">Empty</td><td>in case no requested information</td><td rowspan="5">SW1</td><td rowspan="5">SW2</td></tr>
<tr><td>in case invalid parameter 'P1' / 'P2'</td></tr>
<tr><td>in case Lc was invalid</td></tr>
<tr><td>in case Le was invalid or too small</td></tr>
<tr><td>in case of error</td></tr>
</table>

5.15.6 Status-Codes SW1-SW2

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

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

SW1-SW2	Addressed Functional Unit P1	Specification	Meaning
'6700'	CT	Wrong (command) length parameter	Wrong Lc: Inconsistent command body or no Command Data field supported.
'6A00'		Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ P2 specifies not supported value
'6C00'		Wrong (information) length parameter	Wrong Le.
'6400'		Execution Error	
'6900'		Command not allowed	▪ Cardterminal Session: Admin Access Rights required. ▪ No open CT Session
'6A80'		Invalid Data Object	Invalid Data Object

		Incorrect parameters in the command data field	
'6A88'		Missing Data Object	
		Referenced data or reference data not found	Missing Data Object
'9000'		Command successful	Data Object adjusted: Value set.

5.16 Command SICCT INPUT

Das SICCT INPUT-Kommando dient dazu, eine Eingabe des Benutzers über eine vom Kartenterminal kontrollierte Eingabefunktionseinheit (Keyboard oder biometrische Einheit) im Datenfeld der Response zurückzuliefern.

5.16.1 Funktion

Der Parameter ’Le’ bestimmt die Anzahl der Eingabedaten (Bytes). Das Kommando liefert mit der Antwort den Ausgang der Operation sowie angeforderte Daten zurück.

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

- Le ist nicht vorhanden (absent) : Diese Option bietet die Möglichkeit, ggf. nach der Ausgabe einer Textnachricht, auf die Betsätigungstatste oder Abbruchtaste warten zu können.
- Le = ’00’: Bedeutet eine variable Eingabe, mit nicht vorgegebener Länge.
- Le = <n>: Bedeutet eine Eingabe von <n> Eingabezeichen (Bytes).

Zur Eingabeanforderung kann ein optionaler Benutzerdialog und ein optionale Timereinstellungen zur Zeitüberwachung gesetzt werden.

Ausführungsphasen SICCT INPUT				
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

Eingabe via Funktionseinheit vom Typ Keypad

Im Falle der Eingabe über ein Keypad steuern Zusatzoptionen den Benutzerdialog am Kartenterminal, sofern das Kartenterminal über ein Display verfügt.

- Damit der Benutzer seine Eingabe am Display verfolgen kann, soll optional eine Indikation über das Keypad eingegebener Zeichen dargestellt werden. Entweder erfolgt

diese im Klartext oder als Sequenz von Asterics (*)-Zeichen. Andere Zeichen sollen nicht erlaubt sein.

- Wird kein Datenobjekt (Application Label DO oder SICCT Message To Be Displayed DO) mit einer Anzeigenachricht übergeben, so erfolgt an einem SICCT mit Display die Darstellung der Standardnachricht Nummer 11 (s. 7.2) "Bitte Dateneingabe".
- Vor der Ausgabe einer Display-Nachricht wird das Display zuvor gelöscht.
- Die Ausgabe von Nachrichten an einem Terminal mit Display kann generell unterdrückt werden (s. Parameter P2).
- Für den Fall, dass keine Nachricht ausgegeben werden soll, d.h. P2='7x' (No message to be displayed), wird ebenfalls das Display gelöscht.
- Optional kann ein akustisches oder optisches Signal (s. Parameter P2) zu Beginn der angeforderten Dateneingabe ausgeben werden, sofern das Terminal diese Option unterstützt.
- Wird kein Datenobjekt (WAIT TIME DO) mit einer expliziten Timeout-Vorgabe entsprechend der Zeitperiode in Sekunden bis zur Eingabe des ersten Zeichens angegeben, erfolgt eine zeitliche Überwachung durch das Terminal mit Defaultwerten von bis zu max. 15 Sekunden bis zur ersten Eingabe und max. 5 Sekunden zwischen den einzelnen Eingaben bzw. Tastenbetätigungen.
- Werden ein oder zwei Datenobjekte (WAIT TIME DO) mit expliziten Timeout-Vorgaben angegeben, bestimmt das erste DO die Zeitperiode in Sekunden bis zur Eingabe des ersten Zeichens, sowie das zweite DO die zulässige Zeit zwischen den Eingabezeichen. Geben diese Werte den Wert '00' vor, so schaltet dieses die Zeitüberwachung im Terminal ab. Fehlt das zweite DO, verwendet das Terminal den Defaultwert von bis zu max. 5 Sekunden zwischen den einzelnen Eingaben bzw. Tastenbetätigungen.
- 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.
- Wird das Kommando mit variabler Eingabe-Länge verwendet, ist folgendes zu beachten:
  - Die Eingabe muss mit der Bestätigungstaste beendet werden.
  - Sollte nach der Eingabe des letzten Eingabezeichens die Timeout Zeit überschritten werden,

  so sind folgende Terminalaktionen standardkonform:
  (1.) Die Standardnachricht 12 "Abbruch" wird angezeigt, die Eingabe wird abgebrochen und der entsprechende Rückgabewert wird generiert.
  (2.) Die Standardnachricht 10 "Bitte Eingabe bestätigen" wird angezeigt und es wird erneut die eingestellte Timeoutzeit (Standard 5s) auf die Bestätigungstaste gewartet Wird innerhalb der Timeout Zeit die Bestätigungstaste nicht gedrückt, wird die Eingabe wie unter (1.) abgebrochen. Die Unterstützung der Standardnachricht Nummer 10 "Bitte Eingabe bestätigen" ist optional.
- Optional kann die finale Bestätigung der Eingabe durch den Benutzer per Bestätigungstaste explizit angefordert werden (s. Parameter P2). Im Fall eines Timeouts oder bei Erreichen der angeforderten Anzahl Eingabezeichen erscheint an einem Terminal mit Display die Standardnachricht Nummer 10 "Bitte Eingabe bestätigen". Die Unterstützung der Standardnachricht Nummer 10 "Bitte Eingabe bestätigen" ist optional.

- An einem Terminal mit Keypad endet eine angeforderte Dateneingabe generell durch die Betätigung der Bestätigungstaste. Das Terminal beendet die Eingabe und sendet die bis dahin ggf. eingelesene Information an den Aufrufer.

- An einem Terminal mit Keypad endet eine angeforderte Dateneingabe generell durch die Betätigung der Abbruchtaste. Das Terminal beendet die Eingabe und sendet keine ggf. eingelesenen Informationen an den Aufrufer. Bei einem Terminal mit Display erscheint die Standardnachricht Nummer 12 "Abbruch".

- An einem 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 löscht das letzte Zeichen und setzt die Eingabeposition entsprechend (nach links) zurück.

- Bei Überschreiten der Timeout-Grenzen beendet das Terminal die Dateneingabe, und stellt an eine Terminal mit Display die Standardnachricht Nummer 12 "Abbruch" dar. Das Terminal beendet die Eingabe und sendet keine ggf. eingelesenen Informationen an den Aufrufer.

Eingabe via Funktionseinheit vom Typ Biometrischer Sensor:

Im Falle der Eingabe bzw. Aufnahme eines biometrischen Merkmals über eine 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 gelten prinzipiell dieselben Abläufe wie bei der Keypad-Eingabe.

- Es gilt der erste Timeout bis zur Datenpräsentation des biometrischen Merkmals. Weitere Timeoutangaben werden ignoriert.

- 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" ist optional.

5.16.2 Anwendungsbedingungen

Das Kommando SICCT INPUT ist generell nur zulässig sofern das Kartenterminal die Präsenz von Eingabefunktionseinheiten anzeigt (s. SICCT GET STATUS). Dieses sind entweder ein Keypad oder eine biometrische Sensoreinheit.

Abbruchbedingung

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

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

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

5.16.3 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 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 (no cmd data, rsp data): no Lc, Le=1-256 Bytes or extended Le 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	'16'	SICCT INPUT

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)		
		'5'	Address FU of type Keypad Unit	
		'7'	Address FU of type Biometric Sensor Unit	
	bit4 .. bit1	'0'	bit 8 .. bit 5 = '5'	Standard Keypad: '50'
			bit 8 .. bit 5 = '7'	Standard Biometric Sensor Unit: '70' Fingerprint Sensor
		'1' : 'E'	bit 8 .. bit 5 = '5'	1st ... 14th additional Keypad
			bit 8 .. bit 5 = '7'	1st ... 14th specific Biometric Sensor Unit

P2	Command Qualifier:		
	Bit8	0	Input does not need Confirmation Key
		1	Input needs Confirmation Key
	Bit7.. bit5	'70'	Suppress all display messages during command execution
		'0x'	Display standard message no. 11 or message found in data field according 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 '000000' <= Lc <= '00FFFF'	0 <= Nc <= 65535

Data	Command Data		
	In case of Direct Coding of 'P1' (mandatory)		
	Empty	non-existent : in case Lc = '00': no Command Data provided	
	APPLICATION LABEL DO	Application Label Data Object	
	SICCT Message To Be Displayed DO	SICCT Message-To-Be-Displayed Data Object	
	WAIT TIME 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	
	APPLICATION LABEL DO	Application Label Data Object	
	SICCT Message To Be Displayed DO	SICCT Message-To-Be-Displayed Data Object	
	WAIT TIME DO	Waiting Time Data Object	

Le	Length of Requested Data Ne Return up to <Le> bytes of requested information		
	Empty	Le absent	No information requested: Ne = 0
			Usage: • Wait for confirmation or abort key without retun of imput 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 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 = variable input data length up to 65536

5.16.4 Data Objects

Das SICCT INPUT arbeitet mit den folgenden Daten Objekten.

Data Object	COMMAND RESPONSE	Desription	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10
FU CON DO	CMD	Functional Unit Context Data Object	
APPLICATION LABEL DO	CMD	Text / display message.	
SICCT Message To Be Displayed DO	CMD	Constructed TLV-DO containing one Character Set and one Application Label DO.	
WAIT TIME DO	CMD	Max. Waiting Time in seconds	

5.16.5 Response Structure

	Kodierung R-APDU			
	[Body:]		**Trailer**	
	[Requested Data / Information]		Status Byte 1	Status Byte 2
SICCT INPUT	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		

5.16.6 Status-Codes SW1-SW2

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

Das Statusbyte SW2 qualifiziert weitere Umstände in Abhängigkeit zu Kommandoparametern.

SW1-SW2	Addressed Functional Unit P1 of Type		Specification	Meaning
'6400'	'50' Keypad	'7x' Biometric Sensor	Nor or incomplete input in time	Abort condition raised by timeout.
'6401'			Process aborted by pressing of cancel key	Abort condition : User canceled input operation.
'6700'			Wrong (command) length parameter	Wrong Lc: Inconsistent command body or no Command Data field supported.
'6930'			Command with timer not supported.	Terminal does not support the timer option.
'6940'			Command with Display not supported.	Command with Display not supported.
'6941'			Functional Unit (FU) busy / not available.	The addressed FU (display, slot, ..) is busy at the moment or not available.
'6942'			Selected Character Set not supported.	The addressed display does not support the selected character set.
'6A00'			Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ P2 specifies not supported value
'6C00'			Wrong (information) length parameter	Wrong Le.
'9000'			Command successful	Captured Input data returned within body of command response.

5.17 Command SICCT OUTPUT

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.

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

- Display:
  Ausgabe einer Textnachricht zur Benutzerführung am Kartenterminal (die Anzeige bleibt erhalten, bis ein neuer Text gesetzt wird oder, ein optionaler Timer für die Ausgabe abgelaufen ist)

- biometrische Einheit:
  Daten zur Konfigurierung einer biometrischen Einheit.

- Druckerport:
  Ausgabe von Daten über optional vorhandene Ausgabe- oder Kommunikationsports des Kartenterminals: z.B. Druckdaten für einen ggf.(optional) am SICCT angeschlossenen Druckerport.

5.17.1 Funktion

Das Kommando führt die entsprechende Ausgabeaktion durch und liefert mit der Antwort den Ausgang der Operation zurück.

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

- Zur Ausgabe von Textnachrichten ist entweder ein Application Label oder ein SICCT Message To Be Displayed Datenobjekt zu übergeben.
- Vor der Ausgabe der Textnachricht wird das adressierte Display gelöscht.
- Die Übergabe einer leeren Textnachricht in Form eines Application Label oder ein SICCT Message To Be Displayed Datenobjekts der Länge Null, bedeutet ein Löschen des Displays.
- Die Darstellung einer Terminaleigenen StandardTextnachricht (Idle-Message) kann optional über den Parameter P2 erfolgen.
- Die Darstellungsdauer einer Textnachricht kann mittels eines Waiting Time Datenobjekts gesteuert werden, sofern das SICCT - Terminal diese Option unterstützt. Anderenfalls darf das Terminal das OUTPUT-Kommando mit Waiting Time-Angabe mit einem entsprechenden Status negieren.
- Die Funktion ist blockierend, d.h. die Anzeige bleibt erhalten, bis ein neuer Text gesetzt wird oder, der optional angegebene Timerwert für die Ausgabe abgelaufen ist.
- Keine Angabe eines Waiting Time DOs oder dass die explizite Angabe des Wertes Null in einem WTDO bedeutet die unmittelbare Textausgabe und sofortige Rückkehr des Kommandos.
- Eine Angabe eines Waiting Time DOs mit Wert ungleich Null bedeutet die unmittelbare Textausgabe und Rückkehr des Kommandos nach Ablauf der Wartezeit.

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

5.17.2 Anwendungsbedingungen

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

Abbruchbedingung
Kartenterminal mit Keypad: Die Ausführung des Kommandos SICCT OUTPUT kann durch den Anwender des Kartenterminals per Abbruchtaste abgebrochen werden.
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			
BCS	SICCT				Stage		
					1	2	3
✓	✓	✓	✓		✓	no	no

5.17.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	Lc	Data	[Le]
	'80'	'17'	Functional Unit	Command Qualifier	Length Command Data	Command Data	absent
SICCT OUTPUT	▪ 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 (cmd data, no rsp data:) Lc=1-255 Bytes or extended Lc no Le		

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 referenced by Functional Unit Index Data Object. FUI 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
		'1' : 'E'	bit 8 .. bit 5 = '4'	1st ... 14 th additional Display
			bit 8 .. bit 5 = '7'	1st ... 14th specific Biometric Sensor Unit

P2	Command Qualifier:		
	bit8 ... bit1	In case P1 addresses a FU other than Display type.	
		'00'	Default value = do not care
		'xx'	other values RFU
		In case P1 addresses a Display FU	
		'01'	Display the cardterminals' Idle Message
		'70'	Suppress all display messages during command execution
		'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 '000000' <= Lc <= '00FFFF'	0 <= Nc <= 65535

Data	Command Data		
	In case of Direct Coding of 'P1' (mandatory)		
	APPLICATION LABEL DO	Application Label Data Object	
	SICCT Message To Be Displayed DO	SICCT Message-To-Be-Displayed Data Object	
	WAIT TIME DO	Waiting Time Data Object	
	In case of Referenced Coding of 'P1'		
	FUI DO	Functional Unit Index Data Object	
	APPLICATION LABEL DO	Application Label Data Object	
	SICCT Message To Be Displayed DO	SICCT Message-To-Be-Displayed Data Object	
	WAIT TIME 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

5.17.4 Data Objects

Das SICCT OUTPUT arbeitet mit den folgenden Daten Objekten.

Data Object	COMMAND RESPONSE	Desription	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
APPLICATION LABEL DO	CMD	Text / display message.	See 5.5.10.19
SICCT Message To Be Displayed DO	CMD	Constructed TLV-DO containing one Character Set and one Application Label DO.	See 5.5.10.21
WAITING TIME DO	CMD	Max. Display Time in seconds, binary coding	See 5.5.10.22

5.17.5 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		

5.17.6 Status-Codes SW1-SW2

Nach erfolgreicher Ausführung des SICCT OUTPUT Kommandos enthält das erste Byte des Statusworts SW1 den Wert '90'.
Das Statusbyte SW2 qualifiziert weitere Umstände in Abhängigkeit zu Kommandoparametern.

SW1-SW2	Addressed Functional Unit (FU)			Specification	Meaning	
	P1					
	Of Type					
'6410'	'40' Display	'60' Printer	'7x' Biometric Sensor	Functional Unit (FU) unable to process the output data		
'6700'				Wrong (command) length parameter	Message too long.	
'6930'				Command with timer not supported.	Terminal does not support the timer option.	
‘6940’				Command with Display not supported.	Command with Display not supported.	
‘6941’				Functional Unit (FU) busy / not available.	The addressed FU (display, slot, ..) is busy at the moment or not available.	
‘6942’				Selected Character Set not supported.	The addressed display does not support the selected character set.	
'6A00'				Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ P2 specifies not supported value	
'6C00'				Wrong (information) length parameter	Wrong Le.	
'9000'				Command successful	Output data processed by addressed FU	

5.18 Command SICCT PERFORM VERIFICATION

5.18.1 Funktion

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

Das Kommando ermöglicht hierzu die optional zeitgesteuerte

- (optionale) Anzeige von Texten zur Bedienerführung am Kartenterminal mit Display,
- die Entgegennahme einer Abbruchinteraktion an einem Kartenterminal mit PIN-Tastatur,

- die Entgegennahme der PIN-Eingabe an einem Kartenterminal mit PIN-Tastatur,
- die (optionale) Erfassung biometrischer Messdaten an einem Kartenterminal mit biometrischer Sensoreinheit,
- die entsprechende Interaktion mit einer adressierten Chipkarte oder einem RFID Token

Die Interaktion mit der Chipkarte oder dem RFID Token besteht

- in dem Senden des im Datenfeld des SICCT PERFORM VERIFICATION-Kommandos übergebenen Kommandos (Data Object Command-to-Perform, see 5.5.10.23), nachdem die vom Terminal erfasste PIN, der Resetting Code oder ein vom Terminal aufbereiteter biometrischer Messdatensatz in das im DO 'Command-to-perform' angegebene Chipkartenkommando (APDU) an entsprechender Stelle (Insertion Position) eingesetzt wurde, und
- in dem Empfang und der Rückgabe des von der Chipkarte oder RFID-Token generierten Ergebniswerts.

Das Kommando gibt den Status (Erfolg, Abbruch) bzw. das Ergebnis der Verifikation (Erfolg, Fehlerkodierung der Chipkarte) zurück. Generell gilt: Das Kommando liefert mit der Antwort den Ausgang der Operation zurück. Wurde vom Terminal ein Kommando an die Chipkarte gesendet (Command To Perform), reicht das Terminal die von der Chipkarte zurückgegebene Statusinformation (SW1SW2) an den Aufrufer weiter.

Ausführungsphasen SICCT PERFORM VERIFICATION				
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 ▪ build ICC APDU (command to perform) ▪ perform ICC / PICC command
Postprocessing phase	Nachbereitungsphase	3		▪ display message ▪ send return value

5.18.2 Anwendungsbedingungen

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

Die Ausführung von Benutzerdialogen ist nur möglich, 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 stehen zur Verfügung 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 Displays kann zur Zeit ein Display angesprochen werden.

Numerische bzw. alphanumerische Eingaben entsprechend den nachfolgend aufgeführten PIN-/ Passwort-Formaten, sind nur möglich, sofern das Terminal über mindestens ein vom

Terminal kontrolliertes Keypad mit dem entsprechend benötigtem Tastensatz besitzt. Verfügt das SICCT-Terminal über mehrere Keypad kann zur Zeit ein Keypad zur Eingabe angefordert werden.

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

Standardausgabetext		Darstellung in	Änderungsmöglichkeit	
Default Message		Displayed at	Override	
4	"Bitte Geheimzahl eingeben"	Eingabephase	1st Application Label Data Object	1st SICCT Message-To-Be-Displayed Data Object
	"Please enter PIN"	Stage 1		
5	"Aktion erfolgreich"	Ausführungsphase	2nd Application Label Data Object	2nd SICCT Message-To-Be-Displayed Data Object
	"Action successful"	Stage 2		
6	" Geheimzahl falsch/gesperrt"	Nachbereitungsphase	3rd Application Label Data Object	3rd SICCT Message-To-Be-Displayed Data Object
	"PIN wrong or blocked"	Stage 3		
10	"Bitte Eingabe bestätigen"	Ende der Eingabephase bei variabler PIN-Länge	4th Application Label Data Object	4th SICCT Message-To-Be-Displayed Data Object
	"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 inkorrekter Länge	5th Application Label Data Object	5th SICCT Message-To-Be-Displayed Data Object
	"Abort"	Abort by ▪ pressing of Cancel Key before or during stage 1 ▪ card withdraw ▪ PIN entry too long/short		

Für eine Benutzerauthentisierung (CHV) nach ISO 9564-1, ISO/IEC 7816-4 oder CEN 726-3 (VERIFY-Kommando) stehen folgende PIN-Formate zur Verfügung:

Format 2 PIN Block	See 5.5.10.23
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), werden dem Terminal durch ein übergebenes Command-To-Perform Data Object (CMD DO) vorgegeben. Zusätzlich bestimmt das CMD DO über das Control-Byte, ob die Eingabe automatisch mit dem Erreichen der entsprechenden Anzahl Eingabezeichen oder zusätzlich durch die Betätigung der Bestätigungstaste am Keypad vom Karteninhaber zu bestätigen ist.

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

<table>
<tr><td>Aktion</td><td>Standard</td><td>Änderungsmöglichkeit</td><td>Bedingung</td></tr>
<tr><td>Action</td><td>Default Value</td><td>Override</td><td>Condition</td></tr>
<tr><td>Bis zur Eingabe des ersten Zeichens oder Abbruchtaste</td><td rowspan="2">15 Seconds</td><td rowspan="2">1st WAITING TIME Data Object</td><td>Start: Nach Darstellung der ersten Eingabeaufforderung</td></tr>
<tr><td>Until input of 1st character or cancel key</td><td>Starts after presentation of the 1st Display message</td></tr>
<tr><td>Bis zur Eingabe des jeweils nächsten Zeichens oder Abbruchtaste</td><td rowspan="4">5 Seconds</td><td rowspan="4">2nd WAITING TIME Data Object</td><td>Start: Nach der Eingabe des jeweils vorangegangenen Zeichens.</td></tr>
<tr><td>Until input of subsequent characters or cancel key</td><td>Starts after input of the presceding input character.</td></tr>
<tr><td>Bis zur Betätigung der Bestätigungstaste nach der letzten Zeicheneingabe oder Abbruchtaste</td><td>Start: Nach der Eingabe letzten Zeichens.</td></tr>
<tr><td>Until enter key or cancel key.</td><td>Starts after input of the last input character.</td></tr>
</table>

Hat der Karteninhaber die Eingabe vorgenommen, endet die Eingabephase

- mit dem Erreichen der vorgegeben Anzahl Eingabezeichen,
- mit der Bestätigung der Bestätigungstaste innerhalb der vorgegeben Zeitspanne oder
- mit dem Abbruch, in dem der Karteninhaber entweder die Karte dem Kartenterminal entzieht oder die Abbruchtaste verwendet.

Sollte das Kommando mit Variabler PIN-Länge verwendet werden so ist folgendes zu beachten:

- Die PIN-Eingabe muss mit der Bestätigungstaste beendet werden.
- Sollte nach der Eingabe der letzten PIN-Stelle die Timeout Zeit überschritten werden, so sind folgende Terminalaktionen standardkonform:

  1. Die Display Message 12 (Standard: "Abbruch") wird angezeigt, die PIN-Eingabe wird abgebrochen und der entsprechende Rückgabewert wird generiert.

- Die Display Message 10 (Standard: "Bitte Eingabe bestätigen") wird angezeigt und es wird erneut die eingestellte Timeoutzeit (Standard 5s) auf die Bestätigungstaste gewartet Wird innerhalb der Timeout Zeit die Bestätigungstaste nicht gedrückt, wird die PIN-Eingabe wie unter 1. Beschrieben abgebrochen.

Anmerkung:

SICCT Terminals, welche die Variante 1. bei der variablen PIN-Länge unterstützen, verwenden die Display Message 10 (Standard: "Bitte Eingabe bestätigen") nicht. Die Display Message 10 ist somit optional.

Sollen jedoch die Standard Messages durch Datenobjekte im Daten Teil des Kommandos überschrieben werden, so muss auch bei 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 Eingabephase wandelt das Terminal die Benutzereingabe intern in das eingestellte PIN-Format um, und fügt diese in das APDU- Kommando ein.

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

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

Während der Nachbereitungsphase signalisiert das Terminal dem Karteninhaber den Status oder Ausgang der Operation , indem entweder eine Standardmeldung (Standardmessage Nr. 5 oder 6) oder eine übergebene Meldung am eingestellten Display dargestellt wird.

No	Message	Phase	Condition
5	"Aktion erfolgreich"	Ausführungsphase	Chipcard Response equals '90 00'
		Stage 2	
6	" Geheimzahl falsch/gesperrt"	Nachbereitungsphase	Chipcard Response does not equal '90 00'
		Stage 3	

Anschliessend sendet das Terminal eine R-APDU (SW1SW2) an den Aufrufer.

Abbruchbedingung
Das Kommando SICCT PERFORM VERIFICATION hat nur dann eine direkte Auswirkung auf den Benutzer, sofern ein erkennbarer Statuswechsel an der Mensch-Maschine-

Schnittstelle entsteht. Die Ausführung, d.h. eine Anforderung einer Nutzereingabe, kann durch eine Anwenderinteraktion am Kartenterminal beeinflußt oder abgebrochen werden. Zulässige Interaktionen durch den Benutzer sind:

Eingabe der angeforderten Identifikationsdaten innerhalb eines spezifizierten Zeitrahmens während der Eingabephase	Successful Operation
Keine oder unvollständige Eingabe der angeforderten Identifikationsdaten innerhalb eines spezifizierten Zeitrahmens während der Eingabephase	Abort condition
Entnahme der Chipkarte oder des RFID-Tokens aus dem Wirkbereich des Kartenterminals vor und während der Eingabephase,	
Betätigung der Abbruchtaste am Keypad des Kartenterminals während der Eingabephase	

Nach Abbruch per Abbruchtaste gibt das Terminal Standardmeldung Nr. 12 aus.

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			
BCS	SICCT				Stage		
					1	2	3
✓	✓	✓	✓		✓	no	No

5.18.3 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 2 (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). 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 or RFID antenna	
		'x'	x <> '0', '1' : other values RFU	
	bit4 .. bit1	'0'	RFU	
		'1' : 'E'	bit 8 .. bit 5 = '0'	1st ICC-Interface ... 14th ICC-Interface
			bit 8 .. bit 5 = '1'	1st RFID-Token ... 14th RFID-Token

P2	Command Qualifier specifies the Functional Units which performs input data				
	Referenced Coding				
	Bit7 .. bit1	'FF'	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
				'51' : '5E'	1st ... 14th additional Keypad
		'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 <> 'FF' 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 '000000' <= Lc <= '00FFFF'	0 <= Nc <= 65535

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	Empty	non-existent : in case Lc = '00': no Command Data provided
	APPLICATION LABEL DO	Application Label Data Object
	SICCT Message To Be Displayed DO	SICCT Message-To-Be-Displayed Data Object
	WAIT TIME DO	Waiting Time Data Object
	Command-To-Perform 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

	APPLICATION LABEL DO	Application Label Data Object
	SICCT Message To Be Displayed DO	SICCT Message-To-Be-Displayed Data Object
	WAIT TIME DO	Waiting Time Data Object
	Command-To-Perform 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:

5.18.4 Data Objects

Das SICCT PERFORM VERIFICATION arbeitet mit den folgenden Daten Objekten.

Data Object	COMMAND RESPONSE	Desription	Remarks
FUI DO	CMD	See 5.5.10.9	see 5.5.10
FU CON DO	CMD	Functional Unit Context Data Object	
		Functional Unit Index Data Object(s) for command execution (Referenced Coded FU)	
		Selection of ▪ Keypad ▪ Display, ▪ Biometric Sensor Unit ICC or RFID.	
APPLICATION LABEL DO	CMD	Text / display message(s).	
		See 5.5.10.19	
		Up to five Application Label DataObjects (in case no SICCT Message-to-be-Displayed DOs given)	
SICCT Message To Be Displayed DO	CMD	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 SICCT Message-To-be-Displayed DataObjects	
		See 5.5.10.21	

WAIT TIME DO	CMD	Max. Waiting Time(s) in seconds See 5.5.10.22 Up to two Waiting Time DataObjects	
Command-To-Perform DO	CMD	Command To Perform	
		See 5.5.10.23	

5.18.5 Response Structure

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

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 infomation on performed command]	ICC returned status code	
			SW1	SW2

5.18.6 Status-Codes SW1-SW2

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

SW1-SW2	Addressed Functional Unit (FU) P1 Of Type		Specification	Meaning
'6400'	ICC<n>	RFID<n>	Nor or incomplete input in time	Timeout condition met during stage 1.
'6401'			Process aborted by pressing of cancel key	User aborted user entry by pressing the cancel key.
'64A1'			No card present	Verification not performed, because no card present
'64A2'			Card not activated	Verification not performed, because card not activated.
'6700'			Wrong (command) length parameter	Message too long.
'6930'			Command with timer not supported.	Terminal does not support the timer option.

SW1-SW2	Addressed Functional Unit (FU) P1 Of Type		Specification	Meaning
'6940'			Warning Command with Display not supported.	Command with Display not supported.
'6941'			Functional Unit (FU) busy / not available.	The addressed FU (display, slot, ..) is busy at the moment or not available.
'6942'			Selected Character Set not supported.	The addressed display does not support the selected character set.
'6983'			Verification Method Blocked.	Note: Chipcard generated status word in case the PIN verification failed, because the authentication method is blocked.
'6A00'			Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ P2 specifies not supported value
'6C00'			Wrong (information) length parameter	Wrong Le. If Le exists, here it has to be set to '00'.
'63Cx'			Verification unsuccessful. x = number of possible retries	Note: Chipcard generated status word in case the PIN verification failed.
'9000'			Command successful	Note: Chipcard generated status word in case the PIN verification was successful.
Note: In case the Terminal has sucessfully 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'.				

5.19 Command SICCT MODIFY VERIFICATION DATA

Das Kommando SICCT MODIFY VERIFICATION DATA stellt einen Spezialfall zur Änderung der Kartenauthentfikationsdaten, einer PIN, eines Kartenpassworts oder eines biometrischen Merkmals (PIN, Passwort, biometrisches Merkmal) dar.

5.19.1 Funktion

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

Das Kommando bedingt dieselben Anwendungsbedingungen und bietet prinzipiell gleiche Funktionalitäten wie das SICCT PERFORM VERIFICATION Kommando, bis auf die Tatsache, dass weitere Dialoge (Meldungen) und Benutzereingaben zur Eingabe des neuen Merkmals notwendig sind. Die Eingabe des neuen Merkmals ist nach der ersten Eingabe zu wiederholen, damit das Terminal einen internen Vergleich vornehmen kann.

Die Interaktion mit der Chipkarte oder dem RFID Token besteht bei diesem Kommando

- in dem Senden des im Datenfeld des SICCT MODIFY VERIFICATION DATA - Kommandos übergebenen Kommandos (Data Object Command-To-Perform, see 5.5.10.23), nachdem die vom Terminal erfasste alte und neue PIN, oder Resetting Code oder ein vom Terminal aufbereiteter biometrischer Messdatensatz in das im DO Command-To-Perform angegebene Chipkartenkommando (APDU) an entsprechender Stelle (Insertion Position) eingesetzt wurde, und
- in dem Empfang und der Rückgabe des von der Chipkarte oder RFID-Token generierten Ergebniswerts.

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

Das Kommando liefert mit der Antwort den Ausgang der Chipkartenoperation zurück.

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 ▪ build ICC APDU (command to perform) ▪ perform ICC / PICC command
Postprocessing phase	Nachbereitungsphase	3		▪ display message ▪ send return value

5.19.2 Anwendungsbedingungen

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

Die Ausführung von Benutzerdialogen ist nur möglich, 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 stehen zur Verfügung 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 Displays kann zur Zeit ein Display angesprochen werden.

Numerische bzw. alphanumerische Eingaben entsprechend den nachfolgend aufgeführten PIN-/ Passwort-Formaten sind nur möglich, sofern das Terminal über mindestens ein vom Terminal gesteuertes Keypad verfügt. Verfügt das SICCT-Terminal über mehrere Keypad kann zur Zeit ein Keypad zur Eingabe angefordert werden.

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

Standardausgabetext		Darstellung in	Änderungsmöglichkeit	
Default Message		Displayed at	Override	
4	"Bitte Geheimzahl eingeben"	Eingabephase	1st Application Label Data Object	1st SICCT Message-To-Be-Displayed Data Object
	"Please enter PIN"	Stage 1		

Standardausgabetext		Darstellung in	Änderungsmöglichkeit	
Default Message		Displayed at	Override	
5	"Aktion erfolgreich"	Ausführungsphase	2nd Application Label Data Object	2nd SICCT Message-To-Be-Displayed Data Object
	"Action successful"	Stage 2		
6	" Geheimzahl falsch/gesperrt"	Nachbereitungsphase	3rd Application Label Data Object	3rd SICCT Message-To-Be-Displayed Data Object
	"PIN wrong or blocked"	Stage 3		
7	"Neue Geheimzahl Eingeben"	Eingabephase	4th Application Label Data Object	4th SICCT Message-To-Be-Displayed Data Object
	"Please enter new PIN"	Stage 1		
8	"Eingabe wiederholen"	Eingabephase	5th Application Label Data Object	5th SICCT Message-To-Be-Displayed Data Object
	"Repeat input"	Stage 1		
9	"Geheimzahl nicht gleich. Abbruch"	Eingabephase	6th Application Label Data Object	6th SICCT Message-To-Be-Displayed Data Object
	"PIN not identical. Abort"	Stage 1		
10	"Bitte Eingabe bestätigen"	Ende der Eingabephase bei variabler PIN-Länge	7th Application Label Data Object	7th SICCT Message-To-Be-Displayed Data Object
	"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 inkorrekter Länge	8th Application Label Data Object	8th SICCT Message-To-Be-Displayed Data Object
	"Abort"	Abort by ▪ pressing of Cancel Key before or during stage 1 ▪ card withdraw ▪ PIN entry too long/short		

Für eine Benutzerauthentisierung (CHV) nach ISO 9564-1, ISO/IEC 7816-4 oder CEN 726-3 (VERIFY-Kommando) stehen folgende PIN-Formate zur Verfügung:

Format 2 PIN Block	See 5.5.10.23
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), werden dem Terimnal durch ein übergebenes Command-To-Perform Data Object (CMD DO) vorgegeben. Zusätzlich bestimmt das CMD DO über das Control-Byte, ob die Eingabe automatisch mit dem Erreichen der entsprechenden Anzahl Eingabezeichen oder zusätzlich durch die Betätigng der Bestätigungstaste am Keypad vom Karteninhaber zu bestätigen ist.

Die Eingabephase beginnt mit der Ausgabe einer ersten Meldung über das gewählte Display, welche den Kartenhalter zur Eingabe auffordert. Der Standardtext Nr. 4 wird dargestellt, sofern kein Application Label oder SICCT Message-To-Be-Displayed Data Object übergeben wurde.

Den zeitlichen Ablauf der Eingabe überwacht das Terminal über folgende vorgegebene Timer-Einstellungen, die ebenfalls über WaitingTime Data Objects überschrieben werden können.

<table>
<tr><td>Aktion</td><td>Standard</td><td>Änderungsmöglichkeit</td><td>Bedingung</td></tr>
<tr><td>Action</td><td>Default Value</td><td>Override</td><td>Condition</td></tr>
<tr><td>Bis zur Eingabe des ersten Zeichens oder Abbruchtaste</td><td rowspan="2">15 Seconds</td><td rowspan="2">1st WAITING TIME Data Object</td><td>Start: Nach Darstellung der ersten Eingabeaufforderung</td></tr>
<tr><td>Until input of 1st character or cancel key</td><td>Starts after presentation of the 1st Display message</td></tr>
<tr><td>Bis zur Eingabe des jeweils nächsten Zeichens oder Abbruchtaste</td><td rowspan="4">5 Seconds</td><td rowspan="4">2nd WAITING TIME Data Object</td><td>Start: Nach der Eingabe des jeweils vorangegangenen Zeichens.</td></tr>
<tr><td>Until input of subsequent characters or cancel key</td><td>Starts after input of the presceding input character.</td></tr>
<tr><td>Bis zur Betätigung der Bestätigungstaste nach der letzten Zeicheneingabe oder Abbruchtaste</td><td>Start: Nach der Eingabe letzten Zeichens.</td></tr>
<tr><td>Until enter key or cancel key.</td><td>Starts after input of the last input character.</td></tr>
</table>

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

- mit dem Erreichen der vorgegeben Anzahl Eingabezeichen,
- mit der Bestätigung der Bestätigungstaste innerhalb der vorgegenen Zeitspanne oder
- mit dem Abbruch, in dem der Karteninhaber entweder die Karte dem Kartenterminal entzieht oder die Abbruchtaste verwendet.

Sollte das Kommando mit Variabler PIN-Länge verwendet werden so ist folgendes zu beachten:

- Die PIN-Eingabe muss mit der Bestätigungstaste beendet werden.
- Sollte nach der Eingabe der letzten PIN-Stelle die Timeout Zeit überschritten werden, so sind folgende Terminal-Aktionen standardkonform:

  2. Die Display Message 12 (Standard: "Abbruch") wird angezeigt, die PIN-Eingabe wird abgebrochen und der entsprechende Rückgabewert wird generiert.

  3. Die Display Message 10 (Standard: "Bitte Eingabe bestätigen") wird angezeigt und es wird erneut die eingestellte Timout-Zeit (Standard 5s) auf die Bestätigungstaste gewartet.

Wird innerhalb der Timeout-Zeit die Bestätigungstaste nicht gedrückt, wird die PIN-Eingabe wie unter 1. Beschrieben abgebrochen.

Anmerkung:
SICCT Terminals, die die Variante 1. bei der variablen PIN-Länge unterstützen, verwenden die Display Message 10 (Standard: "Bitte Eingabe bestätigen") nicht. Die Display Message 10 ist somit optional.

Sollen jedoch die Standard Messages durch Datenobjekte im Datenteil des Kommandos überschrieben werden, so muss auch bei Terminals der Variante 1. die Reihenfolge der zu überschreibenden Displaytexte, wie in der weiter oben aufgefürten 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 wandelt das Terminal die Benutzereingabe intern in das eingestellte PIN-Format um, und fügt diese in das APDU- Kommando ein.

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

Nach Ablauf der gesamten Eingabephase sendet das Terminal das aufbereitete Chipkarten-kommando (Ausführungsphase), sofern sich die adressierte Chipkarte bzw. das RFID-Token im entsprechend aktivierten Betriebszustand befindet. Anderenfalls bricht das Terminal die Ausführung ab, verwirft das Kartenkommando und sendet einen entsprechenden Fehler-status an den Aufrufer. Die Ausführungsphase des Command-to-Perform endet mit dem Empfang des von der Chipkarte generierten Statusworts (SW1SW2).

Während der Nachbereitungsphase signalisiert das Terminal dem Karteninhaber den Status oder Ausgang der Operation , indem entweder eine Standardmeldung (Standardmessage Nr. 5 oder 6) oder eine übergebene Meldung am eingestellten Display dargestellt wird.

No	Message	Phase	Condition
5	"Aktion erfolgreich"	Ausführungsphase	Chipcard Response equals '90 00'
		Stage 2	
6	" Geheimzahl falsch/gesperrt"	Nachbereitungsphase	Chipcard Response does not

		Stage 3	equal '90 00'

Anschliessend sendet das Terminal eine R-APDU (SW1SW2) an den Aufrufer.

Abbruchbedingung
Das Kommando SICCT MODIFY VERIFICATION DATA hat nur dann eine direkte Auswirkung auf den Benutzer, sofern ein erkennbarer Statuswechsel an der Mensch-Maschine-Schnittstelle entsteht. Die Ausführung, d.h. eine Anforderung einer Nutzereingabe, kann durch eine Anwenderinteraktion am Kartenterminal beeinflußt oder abgebrochen werden.
Zulässige Interaktionen durch den Benutzer sind:

Eingabe der angeforderten Identifikationsdaten innerhalb eines spezifizierten Zeitrahmens während der Eingabephase	Successful Operation
Keine oder unvollständige Eingabe der angeforderten Identifikationsdaten innerhalb eines spezifizierten Zeitrahmens während der Eingabephase	Abort condition
Entnahme der Chipkarte oder des RFID-Tokens aus dem Wirkbereich des Kartenterminals vor und während der Eingabephase,	
Betätigung der Abbruchtaste am Keypad des Kartenterminals während der Eingabephase	

Nach Abbruch per Abbruchtaste gibt das Terminal Standardmeldung Nr. 12 aus.

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			
BCS	SICCT				Stage		
					1	2	3
✓	✓	✓	✓		✓	no	no

5.19.3 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 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 (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). FUI DO or 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 or RFID antenna		
		'x'	x <> '0', '1' : other values RFU		
	bit4 .. bit1	'0'	RFU		
		'1' : 'E'	bit 8 .. bit 5 = '0'	1st ICC-Interface ... 14th ICC-Interface	
			bit 8 .. bit 5 = '1'	1st RFID-Token ... 14th RFID-Token	

P2	Command Qualifier specifies the Functional Units which performs input data				
	Referenced Coding				
	Bit7 .. bit1	'FF'	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
				'51' : '5E'	1st ... 14th additional Keypad
		'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 <> 'FF' 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 '000000' <= Lc <= '00FFFF'	0 <= Nc <= 65535

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	Empty	non-existent : in case Lc = '00': no Command Data provided
	APPLICATION LABEL DO	Application Label Data Object
	SICCT Message To Be Displayed DO	SICCT Message-To-Be-Displayed Data Object

	WAIT TIME DO	Waiting Time Data Object
	Command-To-Perform 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
	APPLICATION LABEL DO	Application Label Data Object
	SICCT Message To Be Displayed DO	SICCT Message-To-Be-Displayed Data Object
	WAIT TIME DO	Waiting Time Data Object
	Command-To-Perform 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 MODIFY VERIFICATION DATA arbeitet mit den folgenden Daten Objekten.

Data Object	COMMAND RESPONSE	Desription	Remarks
FUI DO	CMD	See 5.5.10.9	see 5.5.10
FU CON DO	CMD	Functional Unit Context Data Object	
		Functional Unit Index Data Object(s) for command execution (Referenced Coded FU)	
		Selection of ▪ Keypad ▪ Display, ▪ Biometric Sensor Unit ICC or RFID.	
APPLICATION	CMD	Text / display message(s).	

LABEL DO		See 5.5.10.19	
		Up to eight Application Label DataObjects (in case no SICCT Message-to-be-Displayed Dos given)	
SICCT Message To Be Displayed DO	CMD	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	
		See 5.5.10.21	
WAIT TIME DO	CMD	Max. Waiting Time(s) in seconds	
		See 5.5.10.22	
		Up to two Waiting Time DataObjects	
Command-To-Perform DO	CMD	Command To Perform	
		See 5.5.10.23	

5.19.5 Response Structure

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

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 infomation on performed command]	ICC returned status code	
			SW1	SW2

5.19.6 Status-Codes SW1-SW2

Nach erfolgreicher Ausführung des SICCT MODIFY VERIFICATION DATA - Kommandos enthält das erste Byte des Statusworts SW1 den Wert '90'.
Das Statusbyte SW2 qualifiziert weitere Umstände in Abhängigkeit zu Kommandoparametern.

SW1-SW2	Addressed Functional Unit (FU)	Specification	Meaning
	P1		
	Of Type		

SW1-SW2	Addressed Functional Unit (FU) P1 Of Type		Specification	Meaning
'6400'	ICC<n>	RFID<n>	Nor or incomplete input time	
'6401'			Process aborted by pressing of cancel key	
'6402'			Process unsuccessful, new PIN not identical	
'6700'			Wrong (command) length parameter	Message too long.
'6930'			Command with timer not supported.	Terminal does not support the timer option.
'6940'			Warning Command with Display not supported.	Command with Display not supported.
'6941'			Functional Unit (FU) busy / not available.	The addressed FU (display, slot, ..) is busy at the moment or not available.
'6942'			Selected Character Set not supported.	The addressed display does not support the selected character set.
'6A00'			Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ P2 specifies not supported value
'64A1'			No card present	▪ Verification not performed, because no card present
'64A2'			Card not activated	▪ Verification not performed, because card not activated.
'6C00'			Wrong (information) length parameter	Wrong Le
'9000'			Command successful	Change of Verification Data successful

5.20 Command SICCT COMFORT AUTHENTICATION

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 Kartenterminal kann optional zwei weitere Kommandos zur Verfügung stellen:

SICCT COMFORT ENROLL
SICCT COMFORT AUTH

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

SICCT COMFORT AUTH stellt den erforderlichen Präsentationsprozess, quasi als Bestätigung des berechtigten Karteninhabers, dar. Erst nach der Präsentation und Überprüfung des Referenzmerkmals (PIN, Passwort oder biometrisches Merkmal), wird ein angeforderter Kartenprozess (z.B. Auslösen eines Signaturprozesses) ausgelöst. Das

Kartenterminal oder eine externe Entität stellen für den Vergleichsprozess die entsprechend vorgehaltenen Referenzdaten bereit.

5.20.1 Funktion

Um ein entsprechend geschütztes Kommando ausführen zu können, erhält das Kartenterminal vom Host das Kommando COMFORT_AUTHENTICATION mit dem eindeutigen Kartenikdentifikator (z.B. Kartenseriennummer) . Das Kartenterminal kann dann das entsprechende Benutzeridentifaktionsmerkmal (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

5.20.2 Anwendungsbedingungen

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

Das Kommando SICCT COMFORT AUTHENTICATION wird im Betriebsmodus BCS nicht unterstützt.

SICCT COMFORT AUTHENTICATION							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable			
BCS	SICCT				Stage		
					1	2	3
no	✓	✓	✓	no	no	no	no

5.20.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	[Lc]	[Data]	[Le]
SICCT COMFORT	'80'	'21'	Functional Unit	Command Qualifier	Length Command Data	Command Data	Length Requested Data

AUTHENTICATION	▪ CLA = Class ▪ INS = Instruction ▪ P1, P2 = Parameter 1 and 2 ▪ L_c = Length of command data field ▪ L_e = Length of expected ▪ SW1, SW2 = Status Bytes	

5.21 Command SICCT COMFORT ENROLL

Mittels einer Anlernfunktion wird ein als Referenzdatum bestehend aus einem biometrischen Merkmal des Karteninhabers sowie einem eindeutigen Kartenmerkmal, z.B. die Kartenseriennummer (ICCSN), gebildet und im Kartenterminal gespeichert.

5.21.1 Funktion

Zur Erzeugung des Referenzdatums liest das Kartenterminal ein eindeutiges Kartenmerkmal sowie ein Identitätsmerkmal des Karteninhabers ein. Optional erfolgt zuvor eine Authentifizierung des Kartenanwenders.

SICCTs mit biometrischem Sensor ermöglichen - falls dies für das betreffende Verfahren sinnvoll ist - auch einen Wechsel der biometrischen Daten (z.B. statt Daumen soll der Zeigefinger bei der Fingerabdruck-Verifikation verwendet werden).

Das Kartenterminal verwaltet und schützt die angelernten und gespeicherten Referenzdaten vor externen Zugriffen. Nach der Aufnahme von Referenzdaten initiiert das SICCT Terminal die zugvorige Authentifizierung des Karteninhabers gegen das Referenzdatum als verpflichtende Ausführungsbedingung für bestimmte Karten- oder Kartenterminalkommandos.

Ausführungsphasen SICCT COMFORT ENROLL				
Phase		**Stage**	**option**	**Description**
preprocessing phase	Vorbereitungsphase	1		▪
processing phase	Ausführungsphase	2		▪
postprocessing phase	Nachbereitungsphase	3		▪ send return value

5.21.2 Anwendungsbedingungen

Abbruchbedingung:
Das Kommando SICCT COMFORT ENROLL kann durch die steuernde Entität in der Ausführung abgebrochen werden.

Das Kommando SICCT COMFORT ENROLL wird im Betriebsmodus BCS nicht unterstützt.

SICCT COMFORT ENROLL							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable			
BCS	SICCT				Stage		
					1	2	3
no	✓	✓	no	no	no	no	no

5.21.3 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 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						

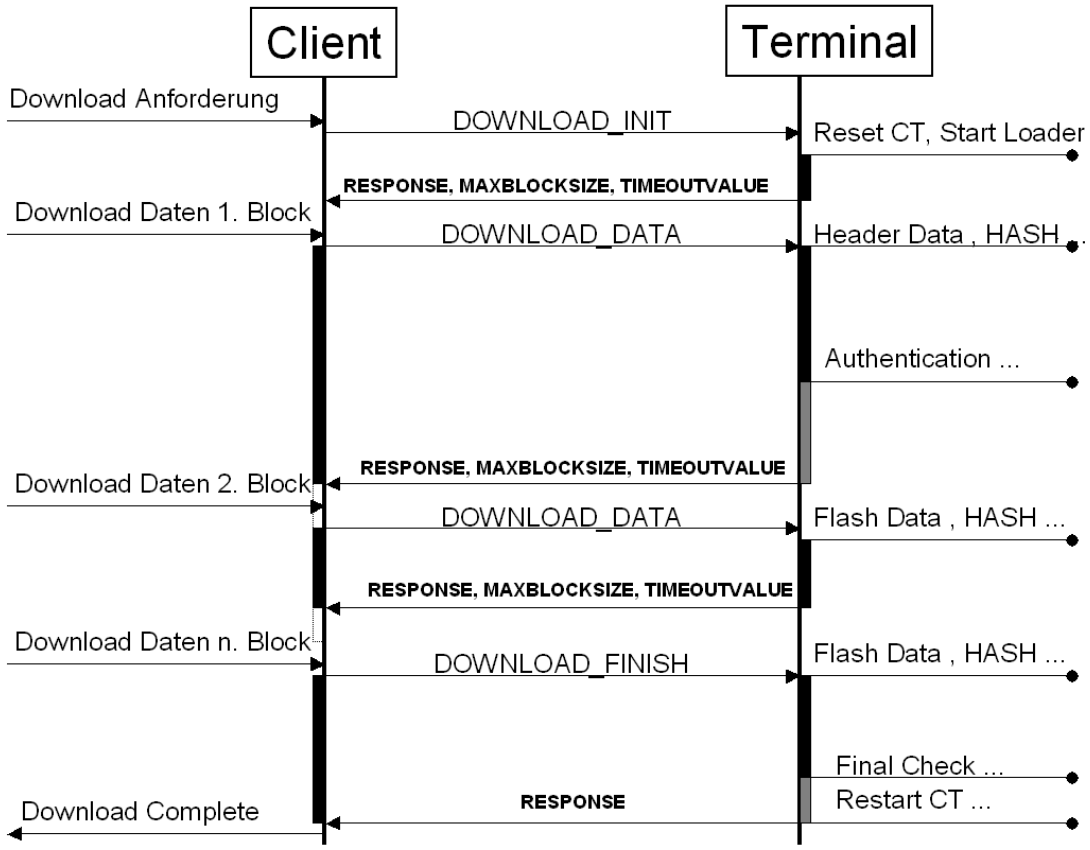
5.22 Command SICCT CT DOWNLOAD INIT

Die Kommandos SICCT CT Download INIT, SICCT CT Download DATA und SICCT CT Download FINISH implementieren eine Updatemöglichkeit des Kartenterminal-Firmware (FW) über die Kommandoschnittstelle.

Für den Download der Firmware wird ein einfacher Transportmechanismus spezifiziert. Downloadobjekte und Format sind herstellerspezifisch. Das Datenformat des Downloadobjekts wird nicht standardisiert werden, da Gerätesoftware und Downloadobjekt stets vom selben Hersteller geliefert werden.
Die Kommandoschnittstelle stellt Mechanismen bereit, um den FW-Download in aufeinanderfolgenden Transferblöcken gesteuert ablaufen zu lassen.
Da der Firmwaretausch für jedes Gerät ein kritischer Vorgang ist, wird der gesamte Downloadvorgang vom Absender zeitüberwacht.
Diese Zeitüberwachung muss variabel gestaltet sein, da z.B. die Lösch- und Programmierzeiten von Flashbausteinen und auch eine Signaturprüfung unterschiedlich lange Laufzeiten beinhalten, die für den proprietären Ladealgorithmus im Gerät individuell gelten.
Im Falle eines Timeout wird der Download abgebrochen. Wenn das Gerät nach dem Abbruch des Download-Vorgangs funktionsfähig geblieben ist, kann der Betrieb des Geräts ohne Update fortgesetzt werden oder der Download wiederholt werden.

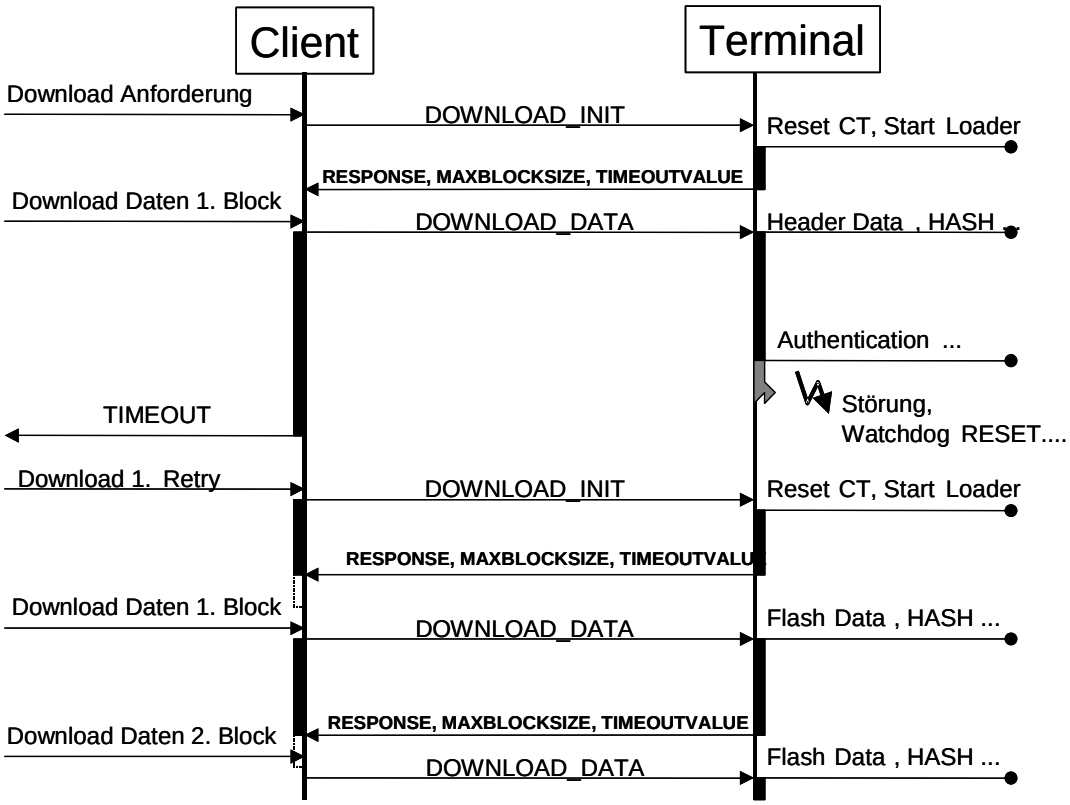
5.22.1.1 Ablauf eines FW-Downloads

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 Kartenterminal.



5.22.1.2 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 Kartenterminal.



5.22.2 Funktion

Das Kommando CT DOWNLOAD INIT initiiert einen Firmware-Download-Mechanismus am Kartenterminal. Nach erfolgreicher Ausführung beginnt eine sog. Download-Session, welche andauert bis zum SICCT CT DOWNLOAD FINISH Kommando. Innerhalb einer DOWNLOAD-Session akzeptiert das Kartenterminal nur die Kommandos

- SICCT CT DOWNLOAD DATA
- SICCT CT DOWNLOAD FINISH

Das Kartenterminal deaktiviert vor dem Eintritt in eine Download-Session die Chipkarten und beendet alle laufenden Transaktionen wie bei einem SICCT RESET CT. Abschließend quittiert das Terminal das Kommando mit einem Antwortdatensatz, bestehend aus den zurückgegebenen Download-Parametern 'RESPONSE, MAXBLOCKSIZE' und 'TIMEOUTVALUE'.

Diese Parameter haben dabei folgende Bedeutung und Codierung:

RESPONSE wie bei allen Kommandos kodiert "OK, Fehler, Busy“auf das vorhergehende Kommando.

MAXBLOCKSIZE, unsigned integer 16 Bit little Endian.
Der Absender des Downloads darf maximal MAXBLOCKSIZE Bytes Downloaddaten im folgenden Block senden.
TIMEOUTVALUE, unsigned integer 16 Bit little Endian. Zeit in Sekunden.
Dieser Parameter wird für die Zeitüberwachung des folgenden Vorgangs vom Absender verwendet.
Der Wert 0000 hält den Timeout Timer an.
Nach Initiierung des Downloads werden Datenblöcke mittels des Kommandos DOWNLOAD_DATA an das Terminal übertragen. Nach jedem Datenblock erfolgt die Verarbeitung im Terminal.
Nach Verarbeitung quittiert das Terminal mittels RESPONSE und setzt neue Werte für MAXBLOCKSIZE und TIMEOUTVALUE.

Der letzte an das Gerät übertragene Datenblock wird mit dem Kommando DOWNLOAD_FINISH gekennzeichnet.
Daraufhin beginnt das Gerät mit dem Abschluss des Downloads mit ggf. einer finalen Gültigkeitsprüfung.
Nach Ende der Prüfung erfolgt die anschließende RESPONSE, die damit auch das Ende des Downloads anzeigt.

Ausführungsphasen SICCT CT 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

5.22.3 Anwendungsbedingungen

SICCT CT DOWNLOAD INIT ist das erste Kommando und leitet den Start der Download-Session ein.

SICCT CT DOWNLOAD INIT							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable			
BCS	SICCT				Stage		
					1	2	3
no	✓	✓	no	no	no	no	no

5.22.4 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 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 (cmd data, no rsp data:) Lc=1-255 or extended Lc no Le		
					Case 3 (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(s) referenced by Functional Unit Index Data Object (FUI DO) or Functional Unit Context Data Object (FU CON DO). FUI DO or FU CON DO contained within Command Data Field.
	bit8 .. bit1	Direct Coding (mandatory)	
		'00'	Address Cardterminal

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 '000000' <= Lc <= '00FFFF'	0 <= Nc <= 65535

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	Empty	non-existent : in case Lc = '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		
	Empty	Le absent	No information requested Ne = 0
	variable length	Le short '01 <= Le <= 'FF'	1 <= Ne <= 255
		Le short Le = '00'	Ne = 256

5.22.5 Data Objects

Das SICCT DOWNLOAD INIT Kommando arbeitet mit den folgenden Daten Objekten.

Data Object	COMMAND RESPONSE	Desription	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10
DLPARAM DO	RSP	Download Parameter Data Object	see 5.5.10

5.22.6 Response Structure

SICCT DOWNLOAD INIT	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		
	DLPARAM DO	Download Parameter Data Object		

5.22.7 Status-Codes Sw1-SW2

SW1-SW2	Addressed Functional Unit (FU) P1	Specification	Meaning
'6200'	CT	Warning - download already started.	Download Session already started.
'6400'		Error	
'6501'		Memory Failure	
'6700'		Wrong (command) length parameter	Message too long.
'6900'		Command not allowed	▪ No open CT Session ▪ Cardterminal busy, Downlaod Session cannot start.
'6A00'		Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. P2 specifies not supported value
'6C00'		Wrong (information) length parameter	Wrong Le.
'6F00'		Communication with CT not possible.	
'9000'		Command successful	Download Initialisation was successful.

5.23 Command SICCT CT Download DATA

5.23.1 Funktion

Das SICCT CT DOWNLOAD DATA Kommando überträgt einzelne Download-Pakete während einer Download-Session an das Kartenterminal.

Ausführungsphasen SICCT CT 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

5.23.2 Anwendungsbedingungen

Das SICCT CT DOWNLOAD DATA Kommando darf nur innerhalb einer Download-Session, d.h. nach einem SICCT DOWNLOAD INIT Kommando gegeben werden.

SICCT CT DOWNLOAD DATA							
CT Mode		Conditions					
		CT ADMIN Session	CT CONTROL Session	Abortable			
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 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 1 (no cmd data, no rsp data) : no Lc, no Le		
					Case 2 (cmd data, no rsp data:) Lc=1-255 or extended Lc no Le		
					Case 3 (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	'25'	SICCT Download 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). FUI DO or FU CON DO contained within Command Data Field.
	bit8 .. bit1	Direct Coding (mandatory)	
		'00'	Address Cardterminal

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 '000000' <= Lc <= '00FFFF'	0 <= Nc <= 65535

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	Empty	non-existent : in case Lc = '00': no Command Data provided
	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	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:

5.23.4 Data Objects

Das SICCT DOWNLOAD DATA Kommando arbeitet mit den folgenden Daten Objekten.

Data Object	COMMAND RESPONSE	Desription	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10
DLDATA DO	CMD	Download Data Object	see 5.5.10.26
DLPARAM DO	RSP	Download Parameter Data Object	see 5.5.10.26

5.23.5 Response Structure

SICCT DOWNLOAD 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		
	DLPARAM DO	Download Parameter Data Object		

5.23.6 Status-Codes Sw1-SW2

SW1-SW2	Addressed Functional Unit (FU) P1	Specification	Meaning
'6400'	CT	Error	
'6501'		Memory Failure	
'6700'		Wrong (command) length parameter	Message too long.
'6900'		Command not allowed	▪ No pending Download Session.
'6A00'		Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ P2 specifies not supported value
'6C00'		Wrong (information) length parameter	Wrong Le
'6F00'		Communication with CT not possible.	
'9000'		Command successful	Transmission of Download Data Object was successful.

5.24 Command SICCT CT Download FINISH

5.24.1 Funktion

Das SICCT CT DOWNLOAD FINISH Kommando beendet eine bestehende Download-Session des Kartenterminals.

Ausführungsphasen SICCT CT 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

5.24.2 Anwendungsbedingungen

Das SICCT CT DOWNLOAD DATA Kommando darf nur innerhalb einer Download-Session, d.h. nach einem SICCT DOWNLOAD INIT bzw. SICCT DOWNLOAD DATA Kommando gegeben werden.

SICCT CT DOWNLOAD FINISH			
CT Mode	Conditions		
	CT ADMIN	CT	Abortable

BCS	SICCT	Session	CONTROL Session		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 FINISH	'80'	'26'	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 (cmd data, no rsp data:) Lc=1-255 or extended Lc no Le		
					Case 3 (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(s) referenced by Functional Unit Index Data Object (FUI DO) or Functional Unit Context Data Object (FU CON DO). FUI DO or FU CON DO contained within Command Data Field.
	bit8 .. bit1	Direct Coding (mandatory)	
		'00'	Address Cardterminal

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 '000000' <= Lc <= '00FFFF'	0 <= Nc <= 65535

Data	Command Data	
	In case of Direct Coding of 'P1' (mandatory)	
	Empty	non-existent : in case Lc = '00': no Command Data provided
	DLDATA DO	Download Data Object
	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		
	Empty	Le absent	No information requested Ne = 0
	variable length	Le short '01 <= Le <= 'FF'	1 <= Ne <= 255
		Le short Le = '00'	Ne = 256:

5.24.4 Data Objects

Das SICCT DOWNLOAD FINISH Kommando arbeitet mit den folgenden Daten Objekten.

Data Object	COMMAND RESPONSE	Desription	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10
DLTERM DO	RSP	Download Termination Data Object	see 5.5.10

5.24.5 Response Structure

SICCT DOWNLOAD FINISH	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		
	DLTERM DO	Download Termination Data Object		

5.24.6 Status-Codes Sw1-SW2

SW1-SW2	Addressed Functional Unit (FU)	Specification	Meaning
	P1		

SW1-SW2	Addressed Functional Unit (FU)	Specification	Meaning
	P1		
'6400'	CT	Error	
'6501'		Memory Failure	
'6700'		Wrong (command) length parameter	Message too long.
'6900'		Command not allowed	No pending Downlaod Session.
'6A00'		Wrong parameters P1, P2	▪ P1 addresses invalid Functional Unit. ▪ P2 specifies not supported value
'6C00'		Wrong (information) length parameter	Wrong Le.
'6F00'		Communication with CT not possible.	
'9000'		Command successful	Download Termination was successful.

6 Schnittstellenbeschreibung zum Host

6.1 Netzwerk Protokoll Festlegungen

Das Protokoll ist auf den Ebenen 5 bis 7 (Sitzungs-, Darstellungs-, Anwendungsschicht) des ISO/OSI-7-Schichtenmodells [ISO74981] bzw. der Anwendungsschicht des TCP/IP-Referenzmodells [RFC1122] angesiedelt, um den Einsatz in möglichst jedem Netz zuzulassen.

Der Schwerpunkt liegt bei Ethernet/IP basierten Netzen; für diese gilt im Speziellen:

Übertragungs-/Sicherungsschicht:
Ethernet Adapter müssen folgende Anforderungen erfüllen:

- 10 Mbit verpflichtend
- 100 Mbit und/oder 1Gbit optional
- Auto-negotiation Mode.

Vermittlungsschicht:
Das Protokoll IPv4 ([RFC791] und Erweiterungen) soll zum Einsatz kommen.
Die Adresse der Vermittlungsschicht bzw. der DNS Name des Terminals dienen zum eindeutigen Auffinden eines Terminals im Netz.

Transportschicht:
Die Protokoll-Suiten TCP und UDP sollen zum Einsatz kommen.

Anwendungsschicht:
Die Protokolle zum Auffinden eines Terminals sowie zur Kommunikation mit einem Terminal sind in der Anwendungsschicht angesiedelt; dazu gehören:

- DHCP/DNS – IP Adressen und Namensvergabe (siehe 0)
- Command Set – Das Protokoll beinhaltet die zu übertragenden Daten und Codes zur Beschreibung der auszuführenden Aktionen (siehe und Command Set).

Zeitkritische Kommandos/Timeouts und Fehlerbehandlung:
Die Einhaltung zeitlicher Grenzen bei der Kommunikation wird durch die jeweiligen Protokolle der durchwanderten Netzwerkschicht sichergestellt.
Die Behandlung von Verbindungs- und Übertragungsfehlern sollen nach Möglichkeit von den Netzwerkprotokollen der Schichten 1 bis 4 behandelt werden.

6.2 Discovery

Das Nutzen eines Terminals im Netzwerk macht drei Dinge notwendig:

1. Adress- und Namensvergabe am Terminal:
   Das Terminal erhält eine IP Adresse und ist optional auch über einen DNS Namen ansprechbar; des weiteren werden andere Netzwerkparameter gesetzt.
2. Dynamisches Auffinden von Terminals durch eine SICCT konforme Applikation (Service Discovery):
   Ein Broad-/Multicast basiertes Protokoll ermöglicht es Applikationen, die aktuellen IP Adressen und Namen der Terminals im Netzwerk zu erhalten. Dieser Schritt wird typischerweise vor der eigentlichen Kommunikation zwischen Anwendung und Terminal mittels der im Command Set definierten Kommandos ausgeführt.

3. Bekannt machen eines Dienstes (Service Announcement):
   Terminals haben zusätzlich die Möglichkeit sich selbstständig bei interessierten Clients bekannt zu machen.

Bei entsprechender statischer Konfiguration von Clients und Terminals können Service Discovery und Announcement auch entfallen. Für das dynamische Auffinden von Terminals muss das Service Discovery Protokoll implementiert sein; Service Announcement ist optional.

6.2.1 Adress- und Namensvergabe

Ein Terminal erhält seine Netzwerkkonfigurationsdaten durch einen der folgenden Mechanismen:

- DHCP [RFC2131]
- AUTO-IP [RFC3927]
- statische Konfiguration

Die Konfigurationsdaten umfassen folgende Werte:

- die IP Adresse des Terminals,
- Subnetzmaske des lokalen Netzwerks
- Gateway IP Adresse
- 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!

Bei Verwendung eines DHCP Servers werden dem Terminal alle nötigen Daten entweder aufgrund von statischen Listen oder dynamisch zugewiesen. Der am Terminal konfigurierbare DNS Name kann vom Terminal selbst einem DHCP Server, der auch die DNS-UPDATE [RFC2136] Erweiterungen unterstützt und für die Zusammenarbeit mit einem DNS Server konfiguriert ist, weitergegeben werden.

Auto-IP, das der konfigurationsfreien Vernetzung von Geräte in lokalen Rechernetzen 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 wird mit Hilfe eines Zufallszahlengenerators am Terminal eine IP Adresse aus dem von der IANA reservierten Bereich 169.254.1.0 bis 169.254.254.255 [RFC 3330] ausgewählt. Der RFC 3927 legt auch die Anzahl und zeitliche Abfolge der Adressierungsversuche, sowie das Verhalten bei Adresskonflikten fest.

Die statische Adresskonfiguration wird über eine der vom Terminal implementierten Managementschnittstellen durch einen Administrator vorgegeben. Die Oberfläche muss die Eingabe der oben genannten Konfigurationsdaten anbieten.

Ablauf der Netzwerkkonfiguration (siehe auch):
Ein Terminal ist entweder für die Verwendung der statischen Netzwerkkonfiguration oder für automatische Adressierung eingestellt. Welcher der beiden Mechanismen angewendet werden soll ist über die Managementschnittstellen einstellbar. Im Kapitel sind die Standardwerte festgelegt.
Bei der statischen Konfiguration trägt der Administrator die Verantwortung für dir Korrektheit der Einstellungen. Falsche Einträge können dazu führen, dass das Terminal nicht mehr über das Netzwerk ansprechbar ist.
Bei dynamischer Konfiguration des Terminals wird zuerst versucht, über das DHCP Protokoll die Konfigurationsdaten zu erhalten. Antwortet innerhalb einer bestimmten Zeit kein DHCP Server, so wird vorerst angenommen, dass keiner vorhanden ist. Als nächstes wird der Auto-

IP Mechanismus eingesetzt, 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.

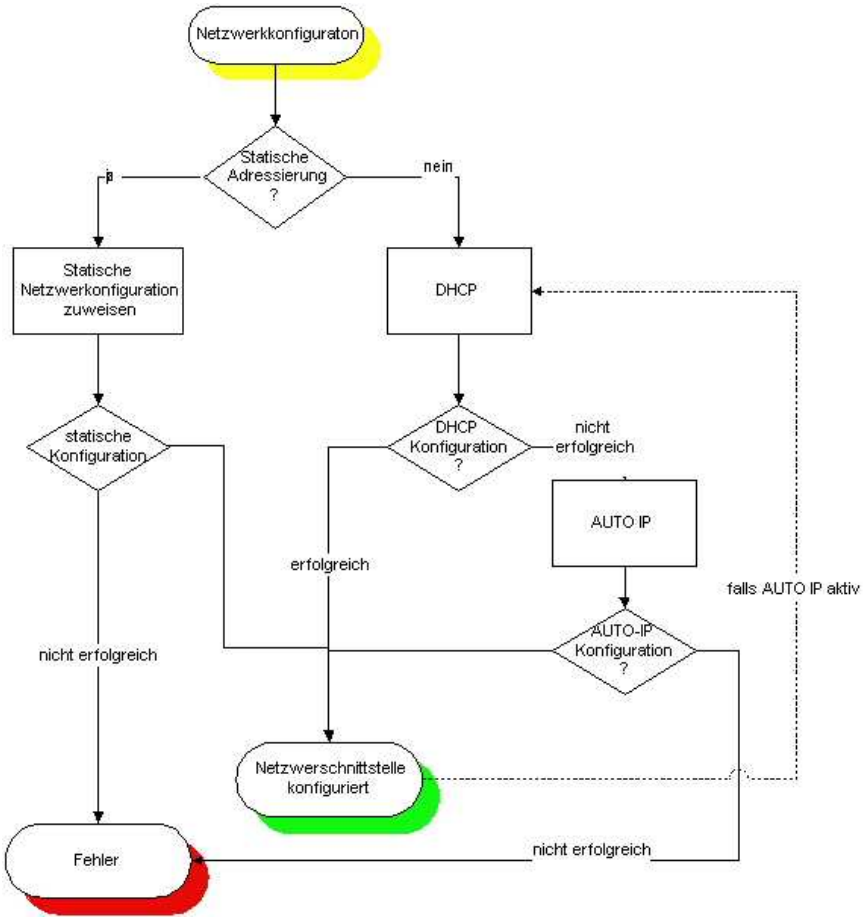


Abbildung 4 Netzwerkkonfiguration

6.2.2 Auffinden eines Terminals (Service Discovery)

Zum Auffinden eines Terminals im Netz durch ein SICCT kompatibles Anwendungsprogramm wird ein UDP/IP ([RFC768] / [RFC 791]) basiertes Broad-/Multicast Protokoll verwendet. Dieses Dokument beschreibt die Version 1.10 des Discovery-Protokolls.

Die Dienstanfrage erfolgt standardmäßig über den UDP Port 4742 (Dienstanfrageport):
Der Client versendet die Anfrage entweder als Broad- oder Multicast. Das Terminal wartet auf eingehende Dienstanfragen an einem Netzwerksocket, der entweder für den Empfang von Broadcast oder Multicasts konfiguriert ist. Es ist daher immer nur möglich diejenigen Dienstanfragen zu empfangen, deren Sendemodus dem Empfangsmodus des Terminals entspricht.

6.2.2.1 Dienstanfrage mittels Broadcast

Die Dienstanfrage mittels Broadcast ist standardmäßig zu unterstützen. Der Broadcast kann sowohl als gerichteter als auch lokaler Broadcast abgesetzt werden. Die Zieladresse bestimmt die Art des Broadcast:

- Für einen Broadcast im lokalen Subnetz wird 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 einstellbar sein. Zusätzlich muss durch die Administration des Netzes sichergestellt werden, dass einerseits der Broadcast bis zum Zielnetzwerk geroutet werden kann, und anderseits Datenpakete von dort antwortenden 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 sendende Teilnehmer üblicherweise das Broadcast-Paket selbst nicht mehr. Das Auffinden eines „lokal“ am Client laufenden SICCT Interpreters ist bei diesem Mechanismus gesondert zu behandeln.

6.2.2.2 Dienstanfrage mittels Multicast

Die Unterstützung der Dienstanfrage mittels Multicast ist optional und kann daher entfallen. Bei Verwendung eines Multicast ist die Multicastgruppenadresse durch die jeweilige Administration zu setzen.
Datenpakete an Multicastgruppen können bei entsprechender Schnittstelleninitialisierung auch von „lokal“ am Client laufenden SICCT Interpreter Instanzen empfangen werden. Auch hier muss die Administration der Netzwerke sicherstellen, dass die Multicastpakete zwischen den Subnetzen weitergeleitet werden.

6.2.2.3 Datenformat des Dienstanfragepakets

Die Daten des von der Client Anwendung versendeten Dienstanfragepakets sind wie folgt codierte TLV Objekte:

Protokoll Version 1.10: Dienstanfragepaket				
Datenfeld	**TAG (hex.)**	**Datenlänge (Bytes)**	**zwingend erforderlich**	**Beschreibung**
Protokoll Version	´80´	2	ja	Versionsnummer der Protokollspezifikation in network-byte-order.
Client IP Adresse	´81´	4	ja	IP Adresse des Clients in network-byte-order.
Client Port	´82´	2	ja	UDP Port am Client in network-byte-order an dem die Rückantwort der Terminals erwartet wird (standardmäßig UDP Port 4742).

Tabelle 8 Dienstanfragepaket

Der Client wartet standardmäßig auf dem UDP Port 4742 auf eingehende Dienstbeschreibungspakete der Terminals. Der Client kann auch so konfiguriert werden, dass die Antworten auf einem anderen UDP Port als 4742 angenommen werden sollen. Der verwendete Port wird immer im Feld „*Client Port*“ des Dienstanfragepakets entsprechend gesetzt.

6.2.2.4 Empfang des Dienstbeschreibungspakets

Nach dem Versand einer Dienstanfrage wartet der Client auf dem UDP Port 4742 (Dienstbeschreibungsport) oder auf dem, im Dienstanfragepaket angegebenen UDP Port, auf eingehende Dienstbeschreibungspakete der Terminals. Es liegt im Ermessen des Host, wie lange auf Antwortpakete der Terminals gewartet wird. Bei der Wahl der Wartezeit sind Geschwindigkeit und Auslastung der zu überbrückenden Netzwerke zu beachten. Für ein lokales 10Mbit Netzwerk sind 3 Sekunden ausreichend.

Jedes Terminal, das ein Dienstanfragepaket empfängt und bereit ist, Anfragen über das SICCT Protokoll zu verarbeiten, sendet ein Dienstbeschreibungspaket mittels UDP/IP an den Client: Die Zieladresse ist die im Dienstbeschreibungspaket angegebene IP Adresse. Zielport ist der vom Client im Dienstbeschreibungspaket im Feld „*Client Port*“ angegebene – standardmäßig ist das der UDP Port 4742.

6.2.2.5 Datenformat des Dienstbeschreibungspakets

Die Daten des vom Terminal versendeten Dienstbeschreibungspakets sind wie folgt codierte TLV Objekte:

Protokoll Version 1.10: Dienstbeschreibungspaket				
Datenfeld	**TAG (hex.)**	**Datenlänge (Bytes)**	**Datenfeld im Paket zwingend erforderlich**	**Beschreibung**
Protokoll Version	´80´	2	ja	Versionsnummer der Protokollspezifikation in network-byte-order.
Terminal IP Adresse	´81´	4	ja	IP Adresse des Terminals in network-byte-order.
Terminal MAC Adresse	´83´	6	ja	MAC Adresse des 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	´85´	n	nein	TLV codierte Beschreibung des zu verwendenden Sicherheitsprotokolls.

Tabelle 9 Dienstbeschreibungspaket

- *SICCT Kommandointerpreter Port:*
  Der Port an dem der SICCT Interpreter auf eingehende Verbindungen wartet, wird immer im Feld „*SICCT Kommandointerpreter Port*“ des Dienstbeschreibungspakets angegeben. Standardmäßig wartet der SICCT Kommandointerpreter auf dem TCP Port 4742 auf eingehende Clientverbindungen. Über entsprechende Managementschnittstellen kann ein Administrator diesen Port ändern. Clients werden

davon durch das entsprechend gesetzte Dienstbeschreibungspaketfeld in Kenntnis gesetzt.

- *Sicherheitsprotokoll:*
  Das Feld enthält die Beschreibung des zu verwendenden sichere Protokolls. Wird kein Protokoll aufgelistet, so wird nur eine „unverschlüsselte“ Verbindung unterstützt: Ein Terminal das unverschlüsselte Kommunikation nicht zulassen möchte listet dementsprechend nur das unterstützte sichere Protokoll. Ein Terminal das nur unverschlüsselte Verbindungen unterstützt listet kein „Sicherheitsprotokoll“.

Sicherheitsprotokolle:					
Protokoll	**TAG (hex.)**	**Datenlänge (Bytes)**	**Daten**	**Wert (hex.)**	**Beschreibung**
TLS	´8A´	1	Unterstützte Protokollversion (1 Byte)	´10´	TLS 1.0 [RFC2246]
				´11´	AES TLS Erweiterungen [RFC4346] [RFC3268]
...	...	...	...	...	...

Tabelle 10 Sicherheitsprotokolle

„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 Events).

TLS:
Mögliche TLS Cipher Suites umfassen die in den Spezifikationen für TLS 1.0 [RFC2246] und TLS 1.1 [RFC4346] beschriebenen, sowie die TLS Erweiterung um AES basiertes Cipher Suites [RFC3268].

Wird TLS implementiert, so muss zumindest die TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA Cipher Suite implementiert sein, um die Mindestanforderungen des TLS 1.0 Standards zu erfüllen (siehe RFC 2246, Kapitel 9: Mandatory Cipher Suites).

6.2.2.6 Konventionen des Dienstanfrage- und Dienstbeschreibungspakets

Die beim Protokoll übertragenen Daten sollen als TLV Objekte (siehe Tabellen) codiert werden: Das TAG und das Längenfeld sind je 1 Byte lang, das Datenfeld hat die im Längenfeld angegebene Länge. 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 unbekannten TAG Werten werden ignoriert.

Format des 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 wird im Discoveryprotokoll das Bit 47 der MAC Adresse auch zuerst übertragen.

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 erfolgt wiederum in *network-byte-order*, d.h. die Hauptversionsnummer wird in Bytestrom zuerst übertragen.
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 1.10	
Hauptversionsnummer	Nebenversionsnummer
15 ... Bit ... 8	7 ... Bit ... 0
0000 0001	0000 1010

Tabelle 11 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 Betriebsystem gängigen Byte-Reihenfolge.

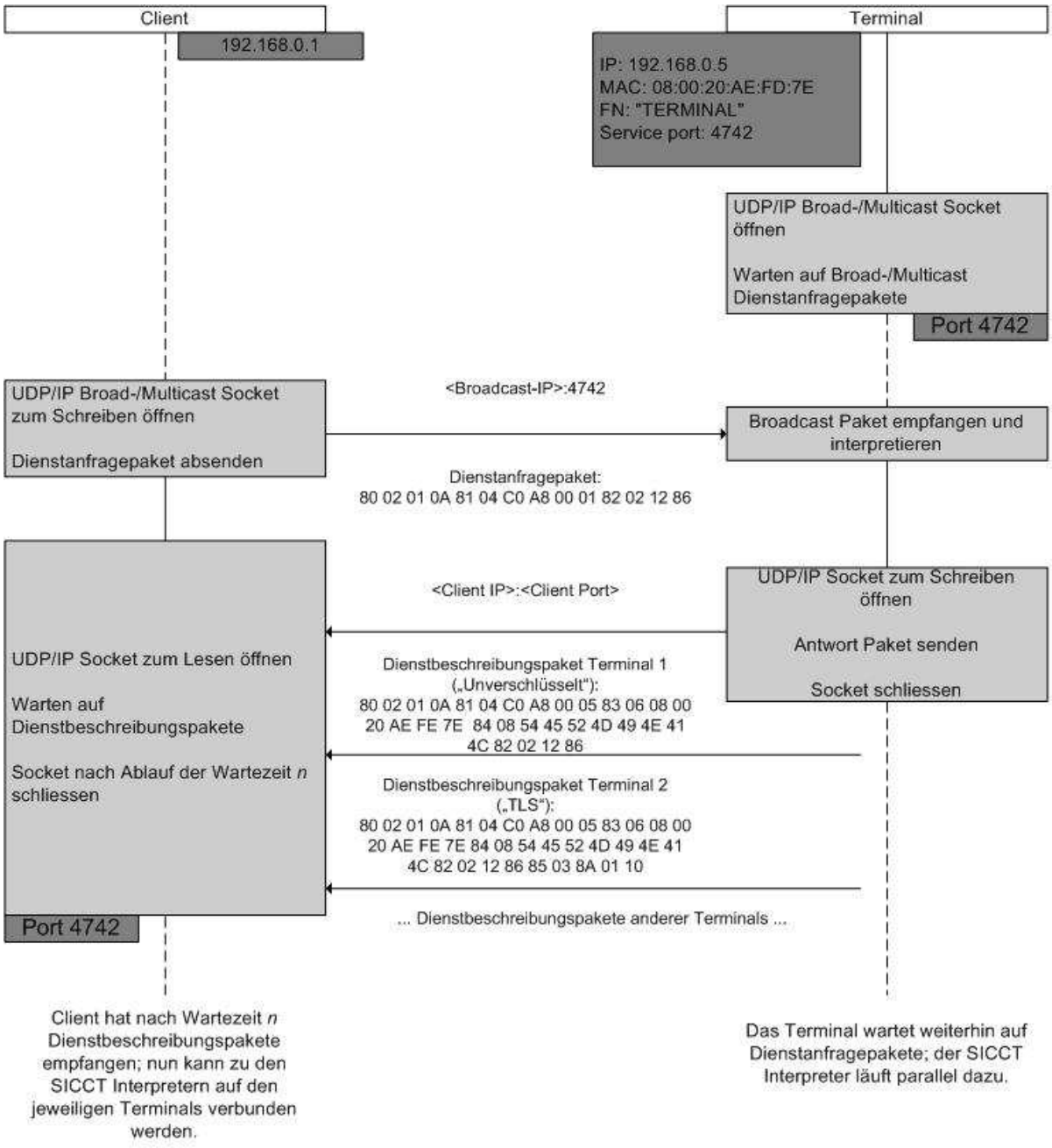


Abbildung 2 Service Discovery Broadcast/Multicast

6.2.3 Bekanntmachung eines Dienstes (Service Announcement):

Ein Terminal hat zusätzlich auch die Möglichkeit, sich selbst im Netzwerk bekannt zu machen: Zu diesem Zweck sendet es ein Dienstbeschreibungspaket mittels einem UDP/IP Broadcast an den Port 4742 (Service Announcement Port).
Die Adresswahl/-berechnung für den Broadcast erfolgt wie beim Versand einer Dienstanfrage.
An Service Announcements interessierte Clients warten am UDP Port 4742 auf eingehende Broadcasts.
Das Terminal erwartet keine Rückantwort von den Clients.

Das Terminal soll diese Nachricht nach dem Wirksamwerden relevanter Terminaleinstellungen senden. Relevante Änderungen umfassen:

- Wechsel der IP-Adresse
- Wechsel des SICCT Kommandointerpreter Ports
- Änderung des Terminalnamens

Die Änderung dieser Parameter erfolgt durch entsprechende administrative Aktionen über eine der Managementschnittstellen.

Das selbständige Versenden von Dienstbeschreibungspaketen durch die Terminals ermöglicht es den Clients schneller von der Anwesenheit des Terminals bzw. den Änderungen am Terminal zu erfahren, ohne selbst das Intervall der Service Discovery Broad-/Multicasts erhöhen zu müssen. Vom periodische Versenden dieser Nachricht ist abzusehen um das Netzwerk nicht unnötig zu belasten!

6.3 Kommandotransport und Namensvergabe

6.3.1 Adressierung

Ist die IP Adresse und der Port, an dem der SICCT Kommandointerpreter an einem 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 Terminals auch über ihren DNS Namen angesprochen werden. Vor allem bei Benutzerinteraktion sind DNS Namen den IP Adressen vorzuziehen, da sie für den Benutzer leichter merkbar sind.

Die Verbindung mit einem Terminal kann eine verschlüsselte TCP/IP Verbindung mit ein- oder zweiseitiger Authentifizierung (siehe Kapitel) sein: Signalisiert ein Terminal die Unterstützung für verschlüsselte Kommunikation, so werden die zum Aufbau der verschlüsselten Verbindung nötigen Schritte sofort nach erfolgreichem Aufbau der zugrundeliegenden TCP/IP Verbindung, aber noch vor der SICCT Protokoll basierten Kommunikation durchgeführt. Die Aushandlung der für die verschlüsselte Kommunikation erforderlichen Parameter (Algorithmen, Schlüssellängen, etc.) erfolgt über die im jeweiligen sicheren Protokoll vorgesehenen Mechanismen.
Der SICCT Kommandointerpreter stellt den Endpunkt der Netzwerkverbindung dar.
Ein SICCT Kommandointerpreter servisiert typischerweise genau einen Client zu einer Zeit. Bei Unterstützung konkurrierender Verbindungen ist jede Art von Ressourcenkonflikten vom Terminal selbst zu managen.

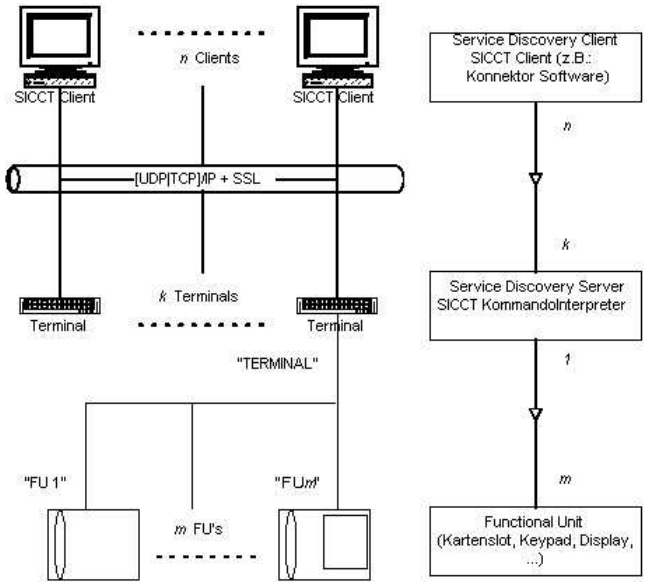


Abbildung 5 SICCT Netzwerkmodell

Ein 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 Terminal selbst ist eine FU die immer vorhanden ist.

Der Dispatcher-Teil des SICCT Kommando Interpreters am Terminal ist dafür zuständig, die eintreffenden SICCT Kommandos soweit zu interpretieren, um entweder das Kommando im Interpreter sofort ausführen zu lassen oder den Empfänger des Kommandos bestimmen zu können, und damit das Kommando an die richtige FU weiterzuleiten.
Die Adressierung und Benennung der FU eines Terminals erfolgt über das im *Kapitel 5.5.6.2 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 Terminal erfragen.

6.3.2 Envelope

Allen SICCT Kommandos (C- und den R-Kommandos, sowie Ereignisnachrichten) wird ein 10-Byte großer Header vorangestellt (=Envelope) der folgende Felder enthält:

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 (6Bh): Zieladresse in *network-byte-order*
			´0000´	SICCT Kommando APDU
			´0001´ – ´00FF´	ICC 1 – ICC 255 Karten APDU
			´1001´ – ´10FF´	RFID 1 – RFID 255 Karten APDU
				Falls R-Kommando (´83´): Quelladresse in *network-byte-order*
			´0000´	SICCT Kommando APDU
			´0001´ – ´00FF´	ICC 1 – ICC 255 Karten APDU
			´1001´ – ´10FF´	RFID 1 – RFID 255 Karten APDU
				Falls Ereignisnachricht (´50´): Quelladresse in *network-byte-order*
			´0000´	SICCT Terminal Ereignis
3	wSeq	2	´0000´ -´FFFF´	Sequenznummer des Kommandos in *network-byte-order*
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 12 SICCT Envelope

Die Felder des Envelope:

- *bMessageType*:
  Das Feld *bMessageType* gibt die Art der im *abCmd* Feld enthaltenen Daten an.
- *wSrcOrDesAddr*:
  Das Feld *wSrcOrDesAddr* spezifiziert die Absender- bzw Empfänger-FU des jeweiligen Kommandos; die Kodierung erfolgt über das in der *Ressourcentabelle* im *Kapitel 5.5.6.2* definierte Schema. Die Adresse wird in *network-byte-order* übertragen.

- *wSeq*:
  Die Sequenznummer *wSeq* wird vom Client aus dem in Kapitel 5.5.10.21 beschriebenen Wertebereichschema für jede abgesandte Nachricht vergeben: Hierbei sind die unterschiedlichen Wertebereiche für Kommandos und Event zu beachten. Der Wert wird in *network-byte-order* übertragen. Der Envelope des zugehörigen R-Kommando des Terminals enthält wiederum die gleiche Sequenznummer. 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 TERMINATE abgebrochen, so wird dessen Sequenznummer mit Empfang der positiven Abarbeitung des Terminate-Kommandos wieder frei.
  Das Terminal muss die gleichzeitige Abarbeitung von Kommandos mit gleicher Sequenznummer innerhalb einer Clientverbindung verweigern.
- *abRFU* :
  Das Feld *abRFU* ist für zukünftige Verwendung frei gehalten.
- *dwLength:*
  Das Feld *dwLength* gibt die Länge der im Feld *abCmd* gesendeten Daten an. Der Längenwert wird in *network-byte-order* übertragen. Die Länge des Envelope ist in dieser Längeninformation nicht enthalten. Diese Längenangabe dient dem korrekten Einlesen von Kommandos über das Netzwerk.

6.3.3 Kommandoabarbeitung

Das Terminal muss in der Lage sein SICCT Kommandos empfangen zu können, obwohl ein früheres SICCT Kommando noch in Abarbeitung ist (Multithreading): Jedes Kommando außer GET STATUS und SICCT TERMINATE) darf durch das Terminal mit dem Fehlercode TERMINAL_BUSY abgelehnt und verworfen werden. Es ist dem Terminal aber auch erlaubt, jedes SICCT Kommando entweder parallel zu bearbeiten oder intern zu queuen.

Die Empfänger FU des SICCT-Kommandos wird im Envelope angegeben:

- ISO 7816 Karten APDU's werden an der im Envelope angegebene Kontaktiereinheit ausgeführt.
- Ein SICCT Kommando APDU wird immer an das Terminal selbst gerichtet. Zur Kommandoausführung werden die in der SICCT Kommando APDU spezifizierte(n) FU(s) verwendet. Die Codierung der „Empfänger FU's" erfolgt mittels „direct bzw. referenced coding" im P1 Feld des SICCT Kommandos. Ob die Kommandoabarbeitung vollständig im Terminal oder aufgeteilt zwischen Terminal und FU abläuft, ist abhängig von der Architektur des Terminals und der Art des Kommandos. Der Typ bzw. die Daten im SICCT Kommando entscheiden dann darüber, welche Aktionen an der FU auszuführen sind.

Der Client wartet nach dem Versand eines SICCT-Kommandos an eine der FU's des Terminals auf eine entsprechend codierte Rückantwort: Der Envelope des R-Kommandos enthält die gleiche Adresse der FU sowie die gleiche Sequenznummer wie das zugehörige C-Kommando.
Der Client muss 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 wiederholt.
Jede Art von Ressourcenkonflikten sind im Terminal selbst zu managen.

Kommandos zur Abfragen und Manipulation von Einstellungen/Daten des Terminals selbst werden immer an die FU „Terminal“ (´0000´) gesendet. Zu diesen Kommandos gehören GET STATUS und TERMINATE.
Die Befehle GET STATUS und TERMINATE sind nicht kaskadierbar. D.h., dass der Zustand eines noch in Ausführung befindlichen GET STATUS Kommandos nicht durch ein weiteres GET STATUS Kommando erfragt werden kann. Genauso kann ein TERMINATE Kommando nicht durch ein weiteres TERMINATE Kommando abgebrochen werden.
Eine Kommandointerpreter Sitzung (Session) wird typischerweise clientseitig durch das CLOSE SESSION SICCT Kommando beendet. Unter bestimmtem Umständen, wie nach Konfigurationsänderungen und bei Auftreten bestimmter Fehler wie in 6.3.6) Fehlerbehandlung beschrieben, kann die Sitzung auch terminal-seitig beendet werden. In jedem Fall werden die gesteckten Karten deaktiviert, und die Initialzustände der FU’s wiederhergestellt.

6.3.4 Ereignisbenachrichtigung

Die Netzwerkverbindung, die zum Austausch der Command Set Kommandos dient, wird auch verwendet um den Client über Ereignisse am Terminal oder seiner FU’s zu informieren. Die Ereignisnachrichten werden aber nur unidirektional vom Terminal zum Client versandt. Das Terminal erwartet keine Antwort vom Client. Der Absender einer Ereignisnachricht ist immer die FU „Terminal“ (´0000´). Die Sequenznummer *wSeq* wird durch das Terminal verwaltet, da die Nummerierung der Ereignisnachrichten von den Kommandos unabhängig ist. Zwei aufeinanderfolgende Ereignisnachrichten müssen unterschiedliche Sequenznummern aufweisen.
Diese Nachrichten dienen dem Terminal dazu, die verbundenen Clients möglichst rasch über Statusänderungen zu informieren. Unabhängig davon kann ein Client immer Kommandos zur Statusabfrage an das Terminal absetzen.

6.3.4.1 Datenformat des Ereignisbenachrichtigungspakets

Die über diesen Kanal versandten Ereignisse sind:

- Protokollereignisse;
  - KEEP ALIVE: dient der terminalseitigen Detektierung des Netzwerkverbindungsstatus
  - TERMINAL SIGN OFF
  - PROTOCOL ERROR
- FU- und Kartenereignisse:
  - Hinzukommen und Entfernen von beliebigen FU’s
  - Slotstatusänderung: Einführen einer Karte in einen Slot bzw. Erkennung einer RFID Karte, sowie deren Entfernung.
- Kommandospezifische Ereignisse:
  - Tastaturereignisse

Die Daten der vom Terminal versendeten Ereignisnachrichten sind wie folgt codierte TLV Objekte:

Datenfeld	TAG (hex.)	Datenlänge (Bytes)	Beschreibung
Protokollereignisse			

KEEP ALIVE	´80´	2	Daten: FU Nummer des Terminals Paket zur Detektierung der Funktionstüchtigkeit der Netzwerkverbindung (siehe 1.1.1.1)	
TERMINAL SIGN OFF	´81´	2	Daten: FU Nummer des Terminals Ankündigung der terminalseitigen Beendigung der Kommandointerpreterverbindung. Das Terminal schließt die Netzwerkverbindung sofort nach Versand der Ereignisnachricht.	
PROTOCOL ERROR	´86´	1	Daten: Error Code (hexadezimal)	
			´00´	Timeout beim Lesen des Headers
			´01´	Timeout beim Lesen der im letzten Header angegebenen Anzahl Datenbytes
			´10´	Nicht unterstützter SICCT Nachrichten Typ
			´11´	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 14 Tastencodes)

Tabelle 13 Ereignisbenachrichtigung

Ein SICCT konformes Terminal muss alle Protokoll-, sowie FU- und Kartenereignisnachrichten aus Tabelle 13 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 5.5.6.2* definierte Schema.
Ereignisfelder gleichen Typs können hier beliebig oft und in beliebiger Reihenfolge vorkommen.
Ein Ereignisfeld enthält immer nur die Daten für ein Ereignis.

Beispiele (hexadezimal, ohne Envelope):

1 Slot „Slot1“ mit der FU-Nummer 0x0001 hinzugekommen (die Nummernzuordnung ist herstellerspezifisch und kann mit dem SICCT-Kommando GET STATUS ausgelesen werden):

```
82 02 00 01
```

1 Slot „Slot1“ mit der FU-Nummer ´0001´ entfernt:

```
83 02 00 01
```

1 Slot „Slot1“ mit FU-Nummer ´0001´ hinzugekommen/1 Slot „Slot2“ mit FU-Nummer ´0005´ entfernt:

```
82 02 00 01 83 02 00 05
```

oder

```
83 02 00 05 82 02 00 01
```

FU- und Kartenereignisse/Handhabung der Ereignisbehandlung logischer Slots von RFID Ressourcen:
Prinzipiell werden kontaktbehaftete und kontaktlose Slots gleich behandelt. Aus der Funktionsweise von RFID Ressourcen ergibt sich aber folgender Spezialfall:
Da jede RFID Ressource aus mehreren logische FU’s besteht, von denen jeweils so viele aktiv sind wie sich Karten in Reichweite befinden, muss immer mindestens eine FU exportiert werden. Diese FU wird exportiert unabhängig davon, ob gerade eine Karte in Reichweite ist oder nicht. Ab der zweiten hinzukommenden Karte wird auch jeweils eine weitere logische FU erzeugt. Mit dem Verschwinden einer Karte verschwindet auch die entsprechende FU. Dies ermöglicht dem Client das Auffinden von RFID FUs, auch wenn keine Karten in Reichweite sind. Weiter erspart dies die Verwaltung der maximalen Anzahl von möglichen logischen FU’s an einer RFID Ressource, wenn weniger als die maximale Anzahl an Karten in Reichweite sind.
Das Hinzukommen der ersten RFID Karte bewirkt nur eine Änderung des Slotstatus, da der Slot bereits existiert. Für alle weiteren RFID Karten werden entsprechend zwei Ereignisse – eines für die Erzeugung der FU „Slot“ und eine Statusänderung dieses Slots - generiert. Das Entfernen der letzten RFID Karte bewirkt wiederum nur eine Statusänderung des Slots, da der Slot bestehen bleiben muss.

Kommandospezifische Ereignisse/Tastaturereignisse:
Die Unterstützung von Tastaturereignissen ist für Terminals mit Display optional, wird jedoch für Terminals ohne Display empfohlen.
Ein Terminal mit Tastaturereignis-Unterstützung ist in der Lage mindestens die als nicht optional gekennzeichneten Tastaturereignisse und Tastencodes aus untenstehender Tabelle (Tabelle 14 Tastencodes) zu senden.
Tastaturereignisse sind nur während der Abarbeitung der SICCT Kommandos VERIFY und INPUT zulässig.
Die Tastaturcodes entsprechen den im PCSC 2.0 Standard (Part 10 Kapitel 2.10 [PCSC20]) für das SPE Kommando FEATURE_GET_KEY_PRESSED definierten Codes: Der Tastencode ´00´ (Kein Taste gedrückt seit letzter Abfrage) ist im Zusammenhang mit ereignisbasierter Benachrichtigung in SICCT nicht sinnvoll, und deshalb nicht in der Liste aufgeführt.

Tastencode (hex.)	Beschreibung
´2B´	‚0’ – ‚9’ Taste gedrückt
´??´	Menu Taste gedrückt „ESC“ ´1B´ bereits vergeben
´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 14 Tastencodes

6.3.5 Timing

- Jede Art von Timingverwaltung für die SICCT Kommandoabarbeitung entfällt auf Terminalseite.
- 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.
- Der Client kann zu jeder Zeit mit dem GET STATUS Kommando den Status der Kommandoabarbeitung am Terminal erfragen.
- Der Client kann Kommandos mit dem TERMINATE Kommando abbrechen. Das Abbrechen eines kann notwendig werden, wenn z.B. ...
    - die Bearbeitungszeit des Kommandos länger dauert als der Client dafür vorgesehen hat
    - ein Fehler- oder eine Abbruchbedingung auf Clientseite aufgetreten ist

6.3.6 Fehlerbehandlung

SICCT APDU Fehler:
Fehlercodes der jeweiligen SICCT Kommando APDU werden in den SW1/2 Feldern des R-Kommando APDU codiert.

SICCT Protokollfehler:
Ein mit falschen Werten befüllter SICCT Envelope soll mit einem PROTOCOL ERROR Event mit dem zutreffenden Error Code (siehe Tabelle 13 Ereignisbenachrichtigung (hexadezimal): ´10´, ´11´, ´12´) beantwortet werden. Das Terminal verwirft einen solchen SICCT Envelope und soviele nachfolgende Bytes wie es das Feld *dwLength* dieses Envelope's angibt. Die nachfolgenden Daten werden wieder als neue SICCT Nachrichten angesehen.

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 Lesen 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 Lesen von zumindest einem Teil der fehlenden Daten wird das *Block Read Timeout* wieder zurückgesetzt.
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.
Ein *Read Timeout* Wert von 5 Sekunden und ein *Message Timeout* von 5 Minuten ist als für den typischen Anwendungsfall ausreichend anzusehen.
Die effektiv benötigte Zeit zum Einlesen einer kompletten SICCT Nachricht ist von der Implementierung des Clients und des Kommandointerpreters, sowie von verschiedenen Netzwerkparametern abhängig; es empfiehlt sich daher die Timeout Werte entweder entsprechend groß vorzuwählen oder über die Terminalkonfiguration einstellbar zu machen.
Läuft eines dieser Timeouts ab, so sendet der Kommandointerpreter ein PROTOCOL ERROR Event mit zutreffenden Error Code (siehe Tabelle 13 Ereignisbenachrichtigung (hexadezimal): ´00´, ´01´).

Wurde das Einlesen einer SICCT Nachricht abgebrochen, so werden die nächsten empfangen Bytes wieder als neue SICCT Nachricht aufgefasst.
Der Kommandointerpreter muss die Netzwerkverbindung selbständig schließen, wenn solche Protokollfehler nacheinander in einer bestimmten Anzahl auftreten. Vor dem Schließen der Verbindung muss ein SIGN OFF Event abgesetzt werden.

Ist einer der hier beschriebenen Empfangsfehler auf eine fehlerhafte Netzwerkverbindung zurückzuführen, die im Empfangsmodus des Kommandointerpreters nicht dedektierbar ist, so wird das Absetzen des zugehörigen PROTOCOL ERROR Events diesen Fehler erkennen. Der Kommandointerpreter schließt in so einem Fall die Netzwerkverbindung unverzüglich ohne vorheriges Absetzen eines SIGN OFF Events.

Protokoll Standardwerte	
Block Read Timeout	5 Sekunden
Message Read Timeout	5 Minuten
Max. Anzahl Protokollfehler	5

Tabelle 15 Protokoll Standardwerte

Netzwerkfehler:
Fehler, die in der Sicherheitsprotokoll Schicht oder darunter liegenden Netzwerkschichten auftreten, werden primär durch deren Mechanismen behandelt: 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.

6.3.6.1 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. Terminal zu ermöglichen solche Fehler der Netzwerkschichten ehest baldig zu erkennen, ist es beiden erlaubt zu beliebigen Zeiten bestimmte Nachrichten als „Heartbeat“ zu versenden:

- Der Client versendet GET STATUS Nachrichten.
- Das Terminal versendet KEEP ALIVE Events.

Dafür darf nur der Status des Versendens eines SICCT Kommandos bzw. Events herangezogen werden.
Für den Client empfiehlt es sich, frühestens drei Sekunden nach Abschluss der letzten Kommunikation (SICCT Kommando, Eventnachricht) ein GET STATUS Kommando zu senden. Es wird empfohlen das der Client GET STATUS Kommandos nicht öfter als einmal alle 500ms, jedoch mindestens alle 10 Sekunden einmal absetzt.
Für das Terminal ist die Empfehlung analog, jedoch sollte das KEEP ALIVE Event erst frühestens vier Sekunden nach Abschluss der letzten Kommunikation versendet werden. Ein größeres Timeout auf Terminalseite begünstigt 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 kann auf das Versenden von GET STATUS Kommandos und KEEP ALIVE Events zur Prüfung des Netzwerkstatus zu solchen Zeitpunkten verzichtet werden.
Das Terminal darf die Netzwerkverbindung selbständig abbauen, selbst wenn die Netzwerkverbindung noch intakt ist, wenn es vom Client mindestens 120 Sekunden lang

keinerlei Kommandos empfangen hat. Dem Verbindungsabbau geht ein TERMINAL SIGN OFF Event voran.

6.3.7 Reset und Wiederaufsatz

Änderungen betreffend der Einstellungen zur Adressierung und Identifikation des Terminals und seiner FU's, sowie gewisse Fehlerzustände machen ein Beenden bestehender SICCT Kommunikationsverbindungen sowie das Zurücksetzen terminalinterner Stati nötig.

Zu diesen Einstellungsänderungen gehören:

- Ändern der IP Adresse
- Ändern des SICCT Kommandointerpreter Ports
- Ändern des Friendlynames einer oder mehrerer FU's
- Austausch des Terminalzertifikats
- Ändern der Sicherheitsprotokolleinstellungen

Zu diesen Fehlerzuständen gehören:

- Detektierung des Entfernens des Netzwerkkabels
- Fehler beim Senden eines SICCT Kommandos (Nachrichten des Keep Alive Mechanismus miteingeschlossen)

Wurden die Einstellungsänderungen über eine der administrativen Schnittstellen ausgelöst, so wird die Verbindung ordnungsgemäß beendet: D.h. es erfolgt ein regulärer Verbindungsabbau entsprechend dem verwendeten Sicherheits- bzw. Netzwerkprotokolls. Beiden Kommunikationspartnern ist es so möglich die Verbindung „sauber" zu beenden. Nachdem das Terminal die Verbindung beendet hat, werden die neuen Einstellungen aktiviert. Nun kann das Auffinden des Terminals durch Clients wieder durch Service Discovery stattfinden. Um Clients schneller davon zu informieren, dass das Terminal wieder verfügbar ist kann es sich wieder mittels Service Announcement im Netzwerk „melden".

Wurde einer der Fehlerzustände erkannt erfolgt terminalseitig die Beendung der Verbindung und das Zurücksetzen der internen Stati. Der Client und das 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 Netzwerkkabel am Terminal selbst entfernt so kann sich dieses, sobald es wieder eine Netzwerkverbindungen feststellen kann, mittels Service Announcement am Netz „melden".
Wurde das Kabel am Client gezogen, so kann dieser bei wiederhergestellter Konnektivität mittels Service Discovery verfügbare Terminals finden.
Bei anderen Netzwerkfehlern muss der Client darauf warten bis seine Service Discovery Nachrichten wieder vom Terminal empfangen und entsprechend beantwortet werden können.

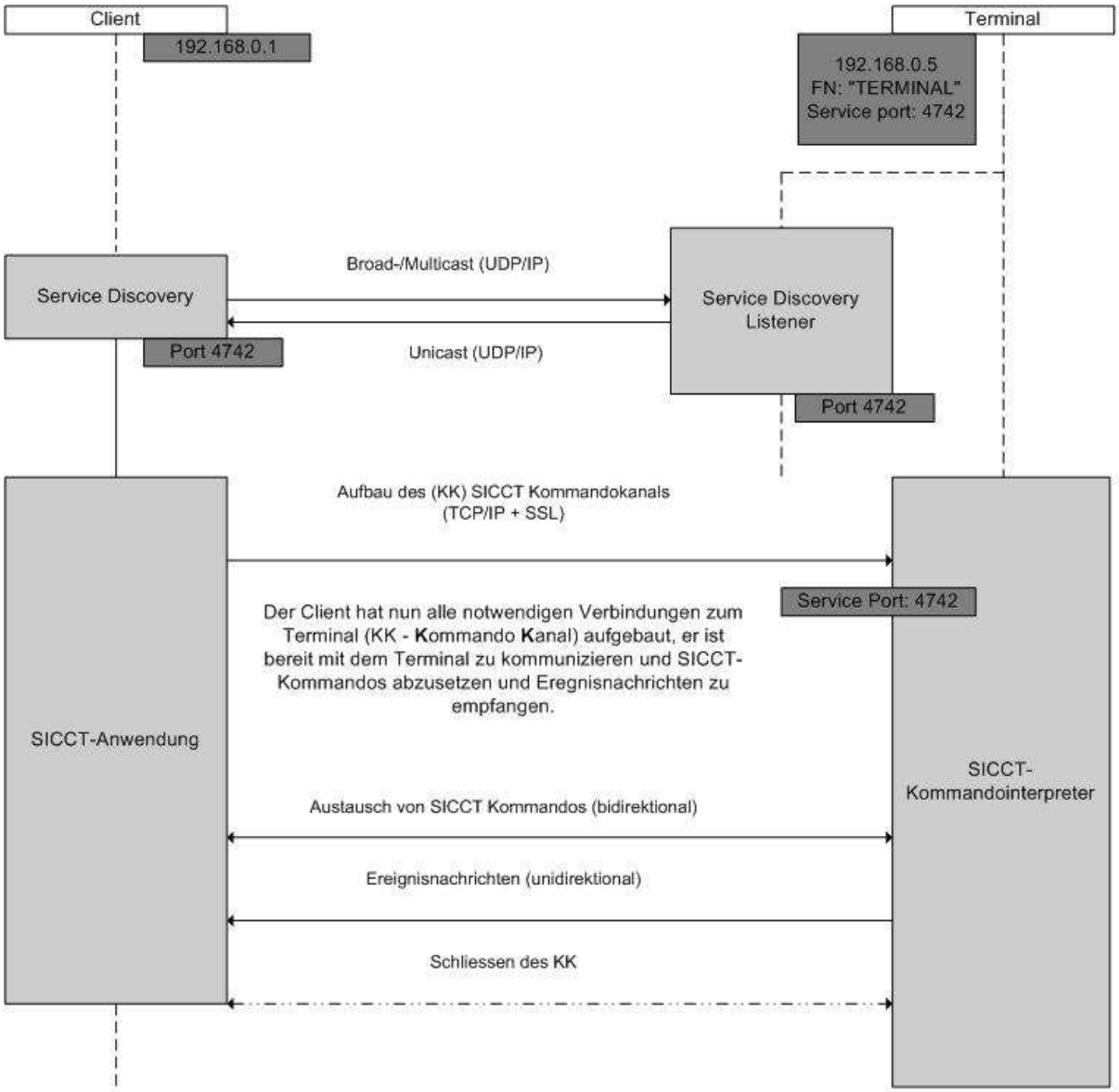


Abbildung 6 SICCT Netzwerkverbindungen

6.4 Auslieferungszustand

Der Auslieferungszustand eines Terminals umfasst:

- Standardwerte für die Netzwerkkonfiguration
  - DHCP ist aktiv
  - IP Adresse und Netzmaske undefiniert oder mit Werten aus einem nicht routebaren Adressbereich [RFC1918] vorbelegt.
  - DNS Name
- Standardwerte für das Service Discovery:
  - Dienstanfrage:
    - Port : 4742 (UDP, Broad-/Multicast)
    - optional: Multicast Adresse
  - Service Announcement:
    - Port : 4742 (UDP, Broadcast)
- Standardwerte für den SICCT Kommandointerpreter:

  - SICCT Kommandointerpreter Port: 4742 (TCP)
  - Name für das Terminal
  - Namen für die FU's entsprechend einer hersteller- oder fachspezifischen Sicht
- Terminal Zertifikat/Schlüssel

6.5 Terminal Managementverfahren

Es soll die Möglichkeit bestehen, ein Terminal über folgendende Schnittstellen über das Netzwerk administrieren zu können:

1. Web-basiertes Management Interface.
2. SICCT-Kommandos.
3. Weitere administrative Schnittstellen sind zulässig, aber herstellerspezifisch.

Für alle Varianten sind entsprechende Methoden zur sicheren Übertragung der Daten zu benutzen (siehe).

Die folgende Tabelle listet die in den jeweiligen Managementschnittstellen zulässigen Aktionen:

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 Terminals	A	opt.	-	opt.
Dienstanfrage und Service Announcement:				
Ändern der Ports	A	opt.	-	opt.
Ändern der Sende-/Empfangsmodi	A	opt.	-	opt.
Ändern der Multicastadresse	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 Terminals	A	mand.	mand.	opt.
Firmware Update	A	opt.	mand.	opt.
Zurücksetzen der Konfiguration in den Werkszustand	A	mand.	-	opt.
„-„ ... nicht zulässig „mand.“ ... mandatory „opt.“ ... optional	Modi: U ... User Modus A ... Admin Modus			

Tabelle 16 Managementaktionen

Die Managementschnittstellen erlauben das Abfragen und Ändern der Konfiguration erst nach erfolgreicher Authentifizierung. Dabei werden zwei Rollen unterschieden:

- Benutzer
- Administrator.

Der Benutzer ist nur berechtigt, die aktuellen Einstellungen anzuzeigen und sein eigenes Kennwort zu ändern. Das Einsehen der aktuellen Konfiguration ist zu schützen, da die Daten der Terminalkonfiguration Informationen enthalten können, die für DoS und ähnliche Attacken benutzt werden können. Der Administrator darf Einstellungen zur Benutzerverwaltung, Netzwerkkonfiguration, den Terminal und Slotnamen sowie Zertifikate ändern.

6.5.1 Web-basierte Managementschnittstelle

Die Organisation und Präsentation der verschiedenen Konfigurationsdaten eines Terminals ist herstellerspezifisch und orientiert sich an den technischen Möglichkeiten und Kapazitäten der Terminalhard- und software (Webserver). Die Webmanagementschnittstelle ist nicht für eine tool-unterstützte automatisierte Geräteverwaltung gedacht; dafür sind geeignete Terminal-Kommandos im Command-Set (siehe) definiert.
Grundlegende Richtlinien an eine solche Benutzerschnittstelle sind in *Kapitel 7 Schnittstelle zum Benutzer* zu definieren.

6.5.2 Administrative SICCT Kommandos

Gewisse Parameter aus der Liste der Managementaktionen können auch über Kommandos des SICCT Protokolls gelesen und verändert werden. So kann eine client-seitige Applikation Befehle aus dem Command Set benutzen, um Terminals zu konfigurieren.
Für die Veränderung dieser Daten ist, so wie bei anderen Managementschnittstellen auch, eine vorangehende erfolgreiche Authentifizierung nötig: Die Authentifizierung erfolgt mittels des INIT SESSION SICCT Kommandos. Dieses Kommando wechselt bei erfolgreicher Authentifizierung den Status der SICCT Kommandointerpreterverbindung vom „User“ in den „Administrator“ Modus. Nach dem Verbindungsaufbau befindet sich die Kommandointerpreterverbindung immer im „User“ Modus. Nach Beendigung der administrativen Kommandos kann mit dem SICCT Kommando CLOSE SESSION wieder in den „User“ Modus zurückgewechselt werden. Im „Administrator“ Modus können nur die entsprechenden administrativen Kommandos abgesetzt werden, aber keine anderen SICCT Kommandos.

6.5.3 Herstellerspezifische Managementschnittstellen

Die Beschreibung der Ausprägung herstellerspezifischer Managementschnittstellen ist nicht Teil dieses Standards. Solche Schnittstellen können beispielsweise mit Hilfe der am Terminal vorhandenen Ein-/Ausgabemöglichkeiten realisiert werden, oder auch andere dem Gerät entsprechende Schnittstellen zur Benutzerinteraktion nutzen.
Bei Verwendung von Einheiten des Terminals, die auch über die SICCT Kommandos der Kommandointerpreterverbindung gesteuert werden können (wie Keyboard und Display), sind im speziellen die Anforderungen bzgl. der Handhabung von Ressourcenkonflikten aus Kapitel 1.1.1 zu beachten.

6.5.4 Allgemeine Anforderungen an den Ablauf des Terminalmanagement

Die Ausführung administrativer Tätigkeiten über andere Schnittstellen als die Kommandointerpreterverbindung sollen den laufenden Betrieb des Terminals so lange wie möglich nicht stören oder beeinflussen. Die Möglichkeit parallel herkömmliche SICCT Kommandos abzuarbeiten und administrative Tätigkeiten auszuführen hängt von der jeweils verwendeten Administrationsschnittstelle und den Fähigkeiten des Geräts ab. Werden bei einer solchen parallelen Abarbeitung gleiche Ressourcen des Terminals beansprucht, so muss die ausführende Logik im 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. Über die SICCT Kommandointerpreterverbindung kann eine bereits verwendete Ressource ihren Status mittels einer BUSY Statusmeldung bekannt geben.

Um verbundenen Clients über einen terminalseitig initiierten Abbau der Kommandointerpreterverbindung zu informieren muss das TERMINAL SIGN OFF Event (siehe 1.1.1.1) versenden. Ein solches Event ist beispielsweise vor dem effektiven Ändern

von Terminaleinstellungen, welche den Zustand der Kommandointerpreterverbindung beeinflussen, nötig.

6.6 Sicherer Kanal

6.6.1 Sicherheitsprotokolle

Das Terminal hat die Möglichkeit über das Feld „*Sicherheitsprotokolle*“ im Dienstbeschreibungspaket dem Client die Unterstützung von Sicherheitsprotokollen anzuzeigen.

Das Terminal erwartet, das beim Verbindungsaufbau versucht wird das Protokoll mit der höchsten Priorität zuerst zu benutzen. Schlägt eine entsprechende Einrichtung der Verbindung fehl, so wird das Protokoll mit der nächstniedrigeren Priorität erwartet. Diese Schritte werden solange wiederholt bis entweder ein Protokoll erfolgreich eingerichtet werden konnte, oder keine weiteren Protokolle mehr zur Verfügung stehen. Konnte kein Protokoll eingerichtet werden so wird die zu Grunde liegende TCP/IP Verbindung geschlossen.

6.6.1.1 TLS

Für die Verschlüsselung der Kommunikation mittels TLS (Transport Layer Security) gilt die Protokollversion 1.0 [RFC2246] sowie die Erweiterungen um AES basierte Cipher Suites [RFC 3268].

Zeigt das Terminal TLS 1.0 Unterstützung an, so sind nur Ciphers Suites aus dem TLS 1.0 Standard relevant. Wird AES Unterstützung angezeigt, so sind auch Cipher Suites aus der erweiterten Liste zulässig. Welche Cipher Suites konkret durch ein Terminal angeboten werden, ergibt sich aus dessen Konfiguration zur Erfüllung fachspezifischer Sicherheitsvorgaben (Protection Profile, Policy).
Bei TLS verschlüsselter Verbindung erfolgt die Aushandlung der zu verwendenden Cipher Suite und ihrer Parameter im Rahmen des im TLS Standard definierten Handshakeverfahrens.

Beim Einsatz von TLS ist eine einseitige Authentifizierung des Terminals gegenüber dem Client verpflichtend. Das Terminal kann optional auch eine Authentifizierung des Clients vornehmen (zwei-seitige Authentifizierung). Dementsprechend muss es dem Terminal möglich sein die Gültigkeit des Client-Zertifikats zu prüfen. (#tbd# Aufbringen/Konfigurieren von CRL, Einsatz von OCSP (Online Certificate Status Protokoll), Klärung der Integration in die eHealth PKI)

Der Einsatz von TLS erfordert die Installation passender Zertifikate am Terminal und Client (siehe 1.1.1).

6.6.2 Zertifikate

Es muss möglich sein ein nach dem X.509v3 Standard [RFC3280] ausgestelltes Zertifikat auf dem 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.

Die Bindung des Zertifikats an das Terminal erfolgt durch Angabe der Terminal MAC Adresse als Common Name (*CN*). Die im Service Discovery Protokoll übermittelte MAC Adresse kann auf Clientseite zur Verifizierung des Zertifikats herangezogen werden (*x.509v3 Extensions Subject Alternative Name*).

Das Zertifikate verwendet als Signaturalgorithmus mindestens SHA-1/RSA mit einer Schlüssellänge von mindestens 1024 Bit.

Die Gültigkeit des Zertifikats ist vom Einsatzgebiet des Terminals abhängig; um einen möglichst langen Einsatz des Terminals ohne Zertifikatstausch zu gewährleisten soll ein Zeitraum von 5 Jahren nicht unterschritten werden.

Für die Webmanagementoberfläche und die Verbindung des Command Sets wird ein gemeinsames Zertifikat verwendet. 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 Contraints*).

Optional kann es möglich sein das Zertifikat am Terminal auszutauschen, um die jeweilige Vertrauensstellung an die Einsatzumgebung des Terminals anzupassen. Das Verändern bzw. das Tauschen des Zertifikats darf nur durch den „Administrator“ (siehe Benutzerrollen in Terminal Managementverfahren) erfolgen: Dementsprechend sind die entsprechenden Aktionen nur nach erfolgreicher Authentifizierung an der Webmanagementoberfläche bzw. im Command Set erlaubt.

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	Terminalherstellerdaten (*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 17 SICCT konformes x.509 Zertifikat

6.6.3 Sicherung des Service Discovery/Announcement

Die Kommandos zum Auffinden der Terminals (Service Discovery/Announcement) werden über unverschlüsselte Verbindungen gesendet.

6.6.4 Sicherung der Verbindung zum SICCT Kommandointerpreter (SICCT Kommunikationskanal)

Wird der SICCT Kommunikationskanal mit einem der zur Verfügung stehenden Sicherheitsprotokolle abgesichert, so wird erst mittels SICCT Kommandos kommuniziert nachdem das Sicherheitsprotokoll erfolgreich eingerichtet wurde.

6.6.5 Sicherung der Managementschnittstellen

SICCT Kommandos:
Es empfiehlt sich die administrativen SICCT Kommandos nur bei verschlüsseltem SICCT Kommunikationskanal anzuwenden.

Webmanagementschnittstelle:
Für den Zugriff auf das Terminal mittels Webmanagementschnittstelle muss immer TLS verwendet werden.
Der Aufbau der TLS Verbindung erfolgt nach der im HTTPS Protokoll definierten Vorgehensweise.

Herstellerspezifische Managementschnittstellen:
Die Sicherung der Daten erfolgt in einer der Art der Schnittstelle angepassten Weise.

6.7 Firmware Download

Das Terminal soll die Möglichkeit bieten, den SICCT Kommandointerpreter, sowie andere „Systemkomponenten“ zu aktualisieren. Die Form der dafür nötigen Daten hängt von der konkreten Implementierung des Terminals ab, und kann somit einer Firmware für einen bestimmten Mikroprozessor, als auch einer PC-ähnlichen Softwarearchitektur entsprechen. Je nach Ausbau des Terminals kann die Aktualisierung der Software demnach deren vollständigen oder auch einen Komponentenweisen Austausch bedeuten. Das Format in dem die Daten bereit gestellt werden, ist demnach herstellerspezifisch. Bei der Übertragung selbst werden die Daten und eine angehängte Signatur in das jeweils verwendete Protokoll eingebettet.

6.7.1 Vorgeschriebener Download Mechanismus

Jedes Terminal muss die Kommandos des Command-Set zum Firmware Download unterstützen.

Optionale Download Mechanismen können zusätzlich vom Hersteller implementiert werden und die Firmware z.B. über Ethernet per http oder über RS-232 oder USB transportieren.

6.7.2 Integrität und Authentizität der Daten

Die Dateien, die ein Firmware Download ausmachen, dürfen nur nach erfolgreicher Überprüfung der Integrität und Authentizität eingespielt werden, um falsche oder böswillig veränderte Daten vorzeitig zu erkennen: Die zur Sicherstellung der Integrität und Authentizität verwendeten Verfahren sind herstellerspezifisch.

6.8 Referenzen

[ISO74981] *ISO standard 7498-1:1994*; http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip.

[RFC791] *RFC 791: Internet Protocol*; DARPA; J. Postel; 1981.

[RFC791] *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.

[RFC4346] *RFC 4346: The Transport Layer Security (TLS) Protocol Version 1.1*; RTFM Inc.; http://www.ietf.org/rfc/rfc4346.txt; 2006;

[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.

[PCSC20] *Interoperability Specification for ICC and Personal Computer Systems Part 10: IFD with Secure Pin Entry Capabilities*; PCSC Workgroup; 2005.

7 Schnittstellenbeschreibung zum Benutzer

7.1 Tastatur

Der Chipkartenleser kann eine Tastatur besitzen. Falls eine Eingabetastatur vorhanden ist, sind folgende Regelungen zu beachten:

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 ist zu achten.

Die Anordnung der Tasten erfolgt in Anlehnung an CEN EN 1332-3 „Identifikationskartensysteme – Schnittstelle Mensch Maschine – Teil 3: Tastenfelder“.

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, ist die Größe von mindestens 2 Zeilen mit je minimal 16 Zeichen zur Darstellung bereitzustellen. Bei einem virtuellem Terminal (z.B. ein PC mit einem Chipkartenleser in der Tastatur) sogt der Software-Proxi für eine Emulation ein solches Display auf dem Monitor. Außer der deutschen Sprachanzeige können weitere Sprachen zur Anzeige von Meldetexten implementiert werden. Weitere Symbole und Zeichen zur Benutzerführung (z. B. Sicherheitsmodus) sind erlaubt. Bei Anzeigetexten mit nachfolgender Tastatur-Eingabe soll ein blinkendes Cursor-Zeichen die Position des Cursors anzeigen. Für den oben beschriebenen Sicherheitsmodus sind Standardtexte im Chipkartenleser vorzuhalten.

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[4]	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	✓	✓

[4] Standard-Anzeigetext Nr. 3 entfällt im SICCT Betriebsmode.

7	Neue Geheimzahl eingeben	Please enter new PIN	✓	✓
8	Eingabe wiederholen	Repeat input	✓	✓
9	Geheimzahl nicht gleich. Abbruch	PIN not iden-tical. Abort	✓	✓
10	Bitte Eingabe bestätigen	Please confirm input	✓	✓
11	Bitte Dateneingabe	Please enter data	✓	✓
12	Abbruch	Abort	✓	✓

Tabelle 25 Standard-Anzeigetexte für SICCT- und BCS Betriebsmode.

CT Idle Message
Zusätzlich zu den den dargestellten Standardtexten verfügt ein SICCT Terminal über eine sog. CT Idle-Message, welche herstellerspezifisch ausfallen über die Mechanismen der Geräteadministration angepaßt werden kann. Die Darstellung der Idle-Message erfolgt nach dem Kommandos, die ein Rücksetzen des Terminals bedeuten, oder per Option des SICCT OUTPUT Befehls.

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 beinhalten ein TLV Objekt.

Kommandos und Daten Objekte sind im Kapitel Command Set definiert

7.2.1.1 Transparente Displayansteuerung „BCS mode“

Bei Verwendung des Kommandos OUTPUT (BCS kompatibel) wird davon ausgegangen, dass ein 2 zeiliges Display mit je 16 Zeichen zur Verfügung steht. Als Zeichenvorrat sind Groß- und Kleinbuchstaben inklusive Umlaute sowie die Sonderzeichen entsprechend DIN 66003, insbesondere inklusive „Stern“, zu unterstützen. Als Steuerzeichen ist in einem Anzeigetext nur CR (0Dh, Wechsel in die zweite Zeile) zulässig.

Falls eine Zeile mit weniger als 16 Zeichen beschrieben wird sind die restlichen Zeichen der Zeile zu löschen.

Als Kommandodaten wird das DO Application Label Data Object erwartet.

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 wird das "SICCT Message To Be Displayed DO" verwendet. 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 2 Zeiliges Display mit je 16 Zeichen zur Verfügung steht. Als Zeichenvorrat sind Groß- und Kleinbuchstaben inklusive Umlaute sowie die Sonderzeichen entsprechend DIN 66003, insbesondere inklusive „Stern“, zu unterstützen. Als Steuerzeichen ist in einem Anzeigetext CR (0Dh, Wechsel in die zweite Zeile) und BEL (07h, Ausgabe eines Beep) zulässig.

Falls eine Zeile mit weniger als 16 Zeichen beschrieben wird sind die restlichen Zeichen der Zeile zu löschen.
Wenn mehr druckbare Zeichen ausgegeben werden sollen als das Display in einer Zeile darstellen kann werden überzählige Zeichen nicht dargestellt bzw. durch eine Scrollfunktion dargestellt.

7.2.2 Leuchtdioden

Optional können durch Leuchtdioden Informationen über den Betriebszustand des Chipkartenleser signalisiert werden.

Eine Leuchtdiode in einer ersten Farbe (vorzugsweise grün) signalisiert den Zustand nach einer korrekten Initialisierung des Chipkartenlesers nach Anlegen der Versorgungsspannung. Der Betriebszustand nach Aktivierung (kontaktbehaftet) oder Selektierung (kontaktlos) der Chipkarte wird durch eine Leuchtdiode in einer zweiten Farbe (vorzugsweise gelb) angezeigt. Eine blinkende Leuchtdiode in der zweiten Farbe oder einer dritten Farbe signalisiert den Fehlerfall.

Falls nur eine Leuchtdioden-Anzeige vorhanden ist, zeigt diese an, 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 können 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 gelangen dazu nicht in die unsichere Umgebung des Hosts.

7.5 Benutzerauthentisierung bei der Komfortsignatur

Eine Authentisierung eines Signaturkarteninhabers erfolgt meist über die Verifikation einer PIN.

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 Terminal kann über eine Anlernfunktion für die vereinfachte Authentisierung verfügen.

Dabei wird z.B. die Kartennummer aus der Signaturkarte ausgelesen werden und zusammen mit einem biometrische Merkmal des Signaturkarteninhabers oder einer eindeutigen Information aus dem sicheren Objekt im Terminal gespeichert.

Wenn dann eine Komfortsignatur durchgeführt werden soll, erhält das Kartenterminal vom Host das Kommando SICCT COMFORT AUTHENTICATION mit der Kartennummer der Signaturkarte. Das Kartenterminal verifiziert dann das biometrische Merkmal bzw. die Anwesenheit des sicheren Objekts.

7.6 Sicherer Modus

Sicherheitstechnische Applikationen erfordern authentische Ein- und Ausgaben. So wird beispielsweise dem Benutzer signalisiert, dass derzeitig 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, ist dieses dem Anwender eindeutig zu signalisieren.

Dazu sind akustische, optische und/oder andere deutlich wahrzunehmenden Signale zur Verfügung zustellen. 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 ist sicherzustellen, dass das Signal nicht unbefugt ansteuerbar ist und nur von der Firmware des Chipkartenlesers bedient wird.
Die Nutzung der Signalisierung ist dem Benutzer in der Dokumentation eindeutig darzustellen.

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

SICCT_R1: 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.

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: Das Kartenterminal stellt eine sichere Komponente für die Verwendung von Chipkarten bereit und signalisiert dem Kartenanwender einen erkennbar sicheren und betraubaren Betrieb.

SICCT_R4: Das Kartenterminal bietet technische Funktionen zur Anwenderauthentifizierung (KeyPad, biometrische Sensorik). Authentisierungsdaten werden vom Kartenterminal nicht an das Host-System weitergegeben, intern nicht gespeichert und fallbezogen angewendet.

SICCT_R5: Das 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 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 (FW) kann über die Lebensdauer des Kartenterminals aktualisiert werden, um

- die Einführung neuer Funktionen
- die Beseitigung von erkannten Sicherheitsrisiken
- eine generelle Fehlerbehebung vorzunehmen.

Voraussetzung hierfür ist, dass das Terminal mit der neuen Firmware bereits validiert ist.

SICCT_R8: Das Kartenterminal eignet 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.

SICCT_R9: Das Kartenterminal basiert auf bestehenden Industriestandards, die additiv für den Betrieb im eHealth 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 Kartenterminal soll konform zu einer gut verständlichen technischen Spezifikation gehalten werden, welche u.a. die Sicherheitsanforderungen beschreibt.

8.2 Funktionale Sicherheit

8.2.1 Mechanik

Anforderungen an die Kartenkontaktiereinheit wie Steckzyklenzahl und Entnahmeschutz sind im Kapitel 4 beschrieben.

8.2.2 Hardware

Sicherheitstechnische Anforderungen an die Hardware sind:

- Kurzschlusssicherheit
- Automatische Deaktivierung der ID-1 ICC beim Ziehen der Karte
- ESD Schutz der ICC Kontakte im Terminal

8.3 Informationssicherheit

8.3.1 Mechanik

Zur Erkennbarkeit von hardware-technischen Manipulationen sind geeignete Methoden zu verwenden (z.B. Versiegelung mit fälschungssicherem Sicherheitsaufkleber, welcher sich bei Entfernung zerstört und damit nur einmal verwendbar ist)

8.3.2 Firmware

Folgende sicherheitstechnische Anforderungen sind durch die Firmware der Terminals zu gewährleisten:

- 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.
- Sichere PIN Eingabe
- Eindeutige Terminal ID
    - bei USB z.B. über Seriennummer im Descriptor (iserial Number).
    - bei Ethernet z.B. mittels MAC Adresse.
- Unterstützung von Rollenprofilen (Administrator, User)
- Wenn Firmware download möglich, dann über sicheren, zertifizierbaren Weg
- Komfort PIN, RF/ID oder Biometrie, wenn vom BSI/Bundesnetzagentur freigegeben.

8.3.3 Kommunikationsschnittstelle

Die Kommunikationsschnittstellen stellen die Anbindung des Terminals an die Infrastruktur dar. Dabei ergeben sich je nach Schnittstellentyp spezifische sicherheitstechnische Anforderungen:

- Seriell
    - Keine
- USB
    - Keine

- Ethernet
    - Kryptografische Mechanismen zur Sicherung einer Ende-zu-Ende Kommunikation; z.B. SSL für Authentisierung und Verschlüsselung der Kommunikation zwischen Terminal und Host (einseitig oder zweiseitig)

8.4 Schützenswerte Elemente

SICCT_O1: Datenverkehr zwischen Terminal und Host

Anhand von Analysen der Anwendungen und Anwendungsumgebungen ist festzulegen, ob und mit welchen Methoden der Datenverkehr zwischen Terminal und Host 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: Authentifikationsdaten (PIN)

Mittels der sicheren PIN Eingabe erfassen die Terminals Identifikationsdaten (PIN) und übermitteln sie auf sicherem Weg an sichere Signaturerstellungseinheiten (Signatur-Chipkarten) gemäß SigG §2 Nummer 10.

Die Firmware hat zu garantieren, dass die Kommandos zum Verifizieren und Modifizieren der PIN erkannt werden und durch Sicherheitsfunktionen zur „Sicheren PIN-Eingabe“ bearbeitet werden. Diese Sicherheitsfunktionen haben zu gewährleistet, dass die PIN-Eingabe über das Keypad des Chipkartenlesers sicher an die Chipkarte weitergeleitet wird, ohne vom Host ausgespäht werden zu können.
Die Speicherbereiche der PIN-Daten sind so durch die Firmware aufzubereiten, dass kein Angriff auf gespeicherte Daten möglich ist.
Die Anforderungen an die Einsatzumgebung sind dahingehend in Handbüchern zu formulieren, dass der Anwender seine Identifikationsdaten unbeobachtet eingeben können muss.
Das Terminal erkennt die von der Host-Software übermittelten Kommandos zur PIN-Eingabe und fügt die über das Keypad eingegebenen Nummern als PIN an die entsprechenden Stellen des Kommandos an die Chipkarte ein. Die PIN selbst verlässt den Leser nie in Richtung Host.
Nachfolgende Liste der 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

Der Mode der sicheren PIN Eingabe ist dem Anwender optisch (LED oder Display) zu visualisieren (siehe Kapitel 7.6).

SICCT_O3: Authentizität und Integrität der Firmware

Wenn das 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.

9 Anmerkungen zum Dokumentenstand

9.1 Dokumentenstand und Haftungsausschluss

Die vorliegende Zwischenversion der SICCT-Spezifikation gibt den momentanen Stand der Arbeiten wieder, aus dem Zielsetzung und Lösungsansätze ersichtlich sind. Das Dokument in der jetzigen Form noch nicht geeignet, als sichere Basis für andere Spezifizierungsarbeiten oder technische Entwicklungen zu 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 Änderungen unterworfen sein kann.

9.2 Zu bearbeitende Punkte

Folgende Punkte sind in der SICCT-Spezifikation noch nicht oder noch nicht abschließend bearbeitet worden:

In Kapitel 5:

- Differenzierung ISO 7810-10 TYPE 1 / TYPE 2: SDA 1 Byte-Adressierung, SDA mit 2 Byte-Adressierung, 2WB, 3WB
- o Differenzierung PICC nach ISO 14443A/B
- o PICC nach ISO 14443A/B, die nicht nach ISO 14443-4 betrieben werden

Command-To-Perform Data Object
- Beispiele

SICCT RESET CT / ICC
- Bedeutung / Verhalten WARM RESET / COLD RESET PICC
- Bedeutung / Verhalten WARM RESET / COLD RESET für ICC nach ISO 7816-10
- Verhalten PPS für PICC

Command SICCT Request ICC / Funktion
- Bedeutung / Verhalten COLD RESET für PICC
- Bedeutung / Verhalten PPS für PICC

Command SICCT Request ICC / Anwendungsbedingungen
- automatisches PPS-Verfahren
- Interner Ablauf der Chipkartenbehandlung im Kartenterminal

Command SICCT Eject ICC / Funktion
- Bedeutung / Verhalten EJECT für PICC

10 Abbreviations

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 Comission
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 Personal Computer 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 References

[STD1] ISO/IEC 7810 (Nov. 2003): Identification cards - Physical characteristics

[STD2] ISO/IEC 7816-1 (Oct. 1998): Identification Cards – Integrated circuit(s) cards with contacts – Part 1: Physical characteristics

[STD3] ISO/IEC 7816-1, Amendment 1 (Nov. 2003): Identification Cards – Integrated circuit(s) cards with contacts – Part 1: Physical characteristics

[STD4] ISO/IEC 7816-2 (March 1999): Identification Cards – Integrated circuit(s) cards with contacts – Part 2: Dimensions and location of the contacts

[STD5] ISO/IEC 7816-2, Amendment 1 (June 2004): Identification Cards – Integrated circuit(s) cards with contacts – Part 2: Dimensions and location of the contacts – Assignment of contacts C4 and C8

[STD6] ISO/IEC 7816-3 (Dec. 1997): Identification Cards – Integrated circuit(s) cards with contacts – Part 3: Electronic signals and transmission protocols

[STD7] ISO/IEC 7816-3, Amendment 1 (June 2002): Identification Cards – Integrated circuit(s) cards with contacts – Part 3: Electronic signals and transmission protocols – Electrical characteristics and class indication for intergrated circuit(s) cards operating at 5V, 3V and 1,8V

[STD8] ISO/IEC 7816-4 (Jan. 2005): Identification Cards – Integrated circuit(s) cards with contacts – Part 4: Organization, security and commands for interchange

[STD9] ISO/IEC 7816-10 (Nov. 1999): Identification Cards – Integrated circuit(s) cards with contacts – Part 10: Electronic signals and answer to reset for synchronous cards

[STD10] ISO/IEC 14443-1 (Apr. 2000): Identification Cards – Contactless integrated circuit(s) cards – Proximity cards – Part 1: Physical characteristics

[STD11] ISO/IEC 14443-2 (July 2001): Identification Cards – Contactless integrated circuit(s) cards – Proximity cards – Part 2: Radio frequency power and signal interface

[STD12] ISO/IEC 14443-2 FDAM2, Amendment 2: Identification Cards – Contactless integrated circuit(s) cards – Proximity cards – Part 2: Radio frequency power and signal interface – Bit rates of $f_c/64$, $f_c/32$ and $f_c/16$

[STD13] ISO/IEC 14443-3 (Febr. 2001): Identification Cards – Contactless integrated circuit(s) cards – Proximity cards – Part 3: Initialization and anticollision

[STD14] ISO/IEC 14443-3 FDAM1, Amendment 1: Identification Cards – Contactless integrated circuit(s) cards – Proximity cards – Part 3: Initialization and anticollision – Bit rates of $f_c/64$, $f_c/32$ and $f_c/16$

[STD15] ISO/IEC 14443-4 (Febr. 2002): Identification Cards – Contactless integrated circuit(s) cards – Proximity cards – Part 4: Transmission protocol

[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

[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)

[MKT_10] Multifunktionale KartenTerminals - MKT, MKT-Version 1.0, 15.04.1999 TeleTrusT Deutschland e.V., http://www.teletrust.de/publikat.asp?id=41220&Sprache=D_&HomePG=0

[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 anzufordern

[IND3] Infineon-Datenbuch, beim Hersteller anzufordern

[gematik_KT] Einführung der Gesundheitskarte, eHealth-Terminal, V0.5, http://www.gematik.de/download/gematik_KT_Spezifikation_V0_5_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 2246] The TLS Protocol Version 1.0 , http://www.ietf.org/rfc/rfc2246.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

Anhang A (normativ)

ISO / EMV Vergleichstabelle

Der folgende Vergleich zeigt die unterschiedlichen Parameter (z.B. Werte, Anforderungen, Verhaltensweisen) folgender beiden Normen auf:

- ISO 7816-3, siehe [STD6] und [STD7]
- 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 das Terminal 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, soll 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**
A1: Electrical Characteristics						
1. VCC class selection	4.2.2	see 4.2.2, Fig. 1	5.5.6	note 4	like ISO, starting with 5V	
2. VCC class A	4.3.2, 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	4.3.2, Tab. 1	2.7...3.3V max. 50mA	5.5.6	2.76...3.24V min. 55mA	like EMV	
4. VCC class C	4.3.2, Tab. 1	1.62...1.98V max. 30mA	5.5.6	1.66...1.94V min. 35mA	like EMV (note 1)	

Description	ISO 7816-3		EMV 4.1		Value to be used by SICCT	
	Ref.	Value	Ref.	Value	ISO Mode	EMV Mode
5. VCC class A spikes	4.3.2, Tab. 2	20nAs @ 400ns/100mA	5.5.6	30nAs @ 400ns/100mA	like EMV	
6. VCC class B spikes	4.3.2, Tab. 2	10nAs @ 400ns/50mA	5.5.6	17,5nAs @ 400ns/50mA	like EMV	
7. VCC class C spikes	4.3.2, Tab. 2	6nAs @ 400ns/50mA	5.5.6	11,1nAs @ 400ns/50mA	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)	4.3.3, Tab. 3	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)	4.3.3, Tab. 3	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)	4.3.3, Tab. 3	max. 1µs	5.5.2.1	max. 0.8µs	like EMV	
13. I/O signal perturbations low voltage (VOL in terminal trans-mission mode)	4.3.3, Tab. 3	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 trans-mission mode)	4.3.3, Tab. 3	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)	4.3.3, Tab. 3	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)	4.3.3, Tab. 3	min. 0V max. 0.15*Vcc @ +1000µA (class C: +500µA)	5.5.2.2	min. 0V max. 0.2*Vcc	like EMV max. current 500µA (class A: 1000µA)	

Description	ISO 7816-3		EMV 4.1		Value to be used by SICCT	
	Ref.	Value	Ref.	Value	ISO Mode	EMV Mode
17. I/O rise and fall times (terminal reception mode)	4.3.3, Tab. 3	max. 1µs	5.5.2.2	max. 1.2µs	like EMV	
18. I/O current limit	note 3		5.5.2	max. ± 15mA	like EMV	
19. CLK high voltage	4.3.4, Tab. 4	min. 0.7*Vcc @ -20µA max. Vcc @ 100µA (note 5)	5.5.4	min. 0.8*Vcc @ 50µA max. Vcc	like EMV	
20. CLK low voltage	4.3.4, Tab. 4	min. 0V @ -100µA max. 0.5V (Class C: 0.2*Vcc) @ 20µA (note 5)	5.5.4	min. 0V max. 0.15*Vcc @ -50µA	like EMV	
21. CLK rise and fall times	4.3.4, Tab. 4	max. 9% of CLK period	5.5.4	max. 8% of CLK period	like EMV	
22. CLK signal perturbations low voltage	4.3.4, Tab. 4	min. -0.3V	5.5.4	min. -0.25V max. 0.15*Vcc	like EMV	
23. CLK signal perturbations high voltage	4.3.4, Tab. 4	max. Vcc+0.3V	5.5.4	min.0.8*Vcc max. Vcc+0.25V	like EMV	
24. CLK signal duty cycle	4.3.4	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. +/- 1%	like EMV	
26. RST high voltage	4.3.5, Tab. 5	min. 0.8*Vcc @ -20µA max. Vcc @ 150µA (note 5)	5.5.4	min. 0.8*Vcc @ 50µA max. Vcc	like EMV	
27. RST low voltage	4.3.5, Tab. 5	min. 0V @ -200µA max. 0.12*Vcc @ 20µA note 5	5.5.4	min. 0V max. 0.15*Vcc @ -50µA	like EMV	
28. RST rise and fall times	4.3.5, Tab. 5	max. 1µs	5.5.4	max. 0.8µs	like EMV	
29. RST signal perturbations low voltage	4.3.5, Tab. 5	min. -0.3V	5.5.4	min. -0.25V max. 0.15*Vcc	like EMV	
30. RST signal perturbations high voltage	4.3.5, Tab. 5	max. Vcc+0.3V	5.5.4	min.0.8*Vcc max. Vcc+0.25V	like EMV	
31. VPP	4.3.6	see Tab. 6	5.5.3	C6 isolated (>10MOhm) or 0...1.05*VCC (class A terminal only)	C6 isolated or like ISO (no VPP generation needed)	

Description	ISO 7816-3		EMV 4.1		Value to be used by SICCT	
	Ref.	Value	Ref.	Value	ISO Mode	EMV Mode
32. contact resistance	note 3		5.5.7	max. 500 mOhm	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	5.2	RST = State L	6.1.2	RST, CLK, I/O = VOL VCC ≤ 0.4V	like EMV	
2. activation sequence	5.2	1. VCC powered 2. I/O in reception mode 3. VPP = VCC (class A only) 4. CLK activated	6.1.2	1. VCC powered 2. I/O in reception mode 3. CLK activated	like ISO (Vpp only if supported)	
3. cold reset: time tb (see ISO Fig. 2)	5.3.2	tb ≥ 400Tclk	6.1.3.1	40000Tclk ≤ tb ≤ 45000Tclk	like EMV	
4. cold reset: time tc (see ISO Fig. 2)	5.3.2	400Tclk ≤ tc ≤ 40000Tclk	6.1.3.1	terminal reception window: 380Tclk ≤ tc ≤ 42000Tclk	like EMV	
5. cold reset: time to deactivation without ATR	5.3.2	min. 40000Tclk	6.1.3.1	min. 42001Tclk max. 42000Tclk + 50ms	like EMV	
6. warm reset: time te (see ISO Fig. 3)	5.3.3	te ≥ 400Tclk	6.1.3.2	40000Tclk ≤ te ≤ 45000Tclk	like EMV	
7. warm reset: time tf (see ISO Fig. 3)	5.3.3	400Tclk ≤ tf ≤ 40000Tclk	6.1.3.2	terminal reception window: 380Tclk ≤ tf ≤ 42000Tclk	like EMV	
8. warm reset: time to deactivation without ATR	5.3.3	min. 40000Tclk	6.1.3.2	min. 42001Tclk max. 42000Tclk + 50ms	like EMV	
9. clock stop mode	5.3.4		note 4		like ISO	
10. PPS	7		note 4		like ISO	
11. deactivation sequence	5.4	1. RST = State L 2. CLK = State L 3. VPP deactive 4. I/O = state A 5. VCC deactive	6.1.5	1. RST = State L 2. CLK and I/O = State L 3. VCC deactive	like ISO (Vpp only if supported)	

Description	ISO 7816-3		EMV 4.1		Value to be used by SICCT	
	Ref.	Value	Ref.	Value	ISO Mode	EMV Mode
12. voltage levels after deactivation	5.4	RST, CLK = State L I/O = state A	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	6.3.2	12etu	8.4	11.8etu	like EMV	
2. max. delay between ATR bytes (initial wait time)	6.3.2	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	16etu (T=0) or 22etu (T=1)	like EMV	

Description	ISO 7816-3		EMV 4.1		Value to be used by SICCT	
	Ref.	Value	Ref.	Value	ISO Mode	EMV Mode
11. T=0 parity error signalling during ATR	6.3.3	optional	9.2.3	not allowed	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	6.5.2	default TA1='01' or 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)	note 3		8.3.3.2	reject ATR (note 6)	like EMV	
4. TB1 is absent or TB1 with any value (warm ATR)	note 3		8.3.3.2	accept ATR as though TB1='00' (default)	like EMV	
5. TD1 containing a protocol unequal T=0 or T=1	6.2	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	6.5.7	capability to change mode of operation	8.3.3.5	b8 should be ignored	like EMV	
7. TA2 is present (specific mode), use of b5	6.5.7	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	6.5.7	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	6.5.4		8.3.3.6	reject ATR (note 6)	like EMV	
10. TC2 = '00' (T=0 only)	8.2	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)	6.2	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)	9.5.2.3	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

Description	ISO 7816-3		EMV 4.1		Value to be used by SICCT	
	Ref.	Value	Ref.	Value	ISO Mode	EMV Mode
13. allowed values for low nibble of TB3 (CWI) (T=1 only)	9.5.3.1	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)	9.5.3.2	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)	9.5.3.1, 9.5.3.2	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)	9.5.4	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. Tai (clock stop and voltage class)	6.5.5, 6.5.6		note 4		like ISO	
A5: T=0 Protocol, Timing and Error Handling						
1. min. delay between received bytes	6.5.3	12etu	9.2.2.1	11.8etu	like EMV	
2. max. delay WWT	8.2	WWT=960*D*WI [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	note 3		9.2.2.1	16etu	like EMV	
6. number of character repetitions after parity error signalling (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	note 3		9.2.3	9600etu (invalid byte to RST=L)	like EMV	

Description	ISO 7816-3		EMV 4.1		Value to be used by SICCT	
	Ref.	Value	Ref.	Value	ISO Mode	EMV Mode
procedure or status byte						
9. definition of CLA byte in command GET RESPONSE	7.6.1 [STD8]		9.3.1.3	CLA='00'	like ISO	like EMV
A6: T=1 Protocol, Timing and Error Handling						
1. min. delay between received bytes	6.5.3	11etu	9.2.4.2 .2	10.8etu	like EMV	
2. max. delay between received bytes (CWT)	9.5.3.1	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)	9.5.3.2	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	9.7.3.2	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)	9.7.2.3	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)	9.5.3.3	22etu	9.2.2.1	reception within 21etu	like EMV	

Description	ISO 7816-3		EMV 4.1		Value to be used by SICCT	
	Ref.	Value	Ref.	Value	ISO Mode	EMV Mode
12. min. guard time after reception to transmission (BGT)	9.5.3.3	22etu	9.2.2.1	22etu	like EMV	
13. NAD	9.4.2.1, 9.6.1	NAD used for node addressing and VPP state control	9.2.4.1 .1	NAD='00', otherwise error handling	like EMV	
14. LEN, I-block with LEN='00'	9.4.2.2	allowed	9.2.4.1 .1	not supported	like ISO	
15. IFSD	9.5.2.2	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	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	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	9.4.2.2	allowed	9.2.4.1 .1	not supported, otherwise error handling	like EMV	

Notes:

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

1 ECard Framework

1.1 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.

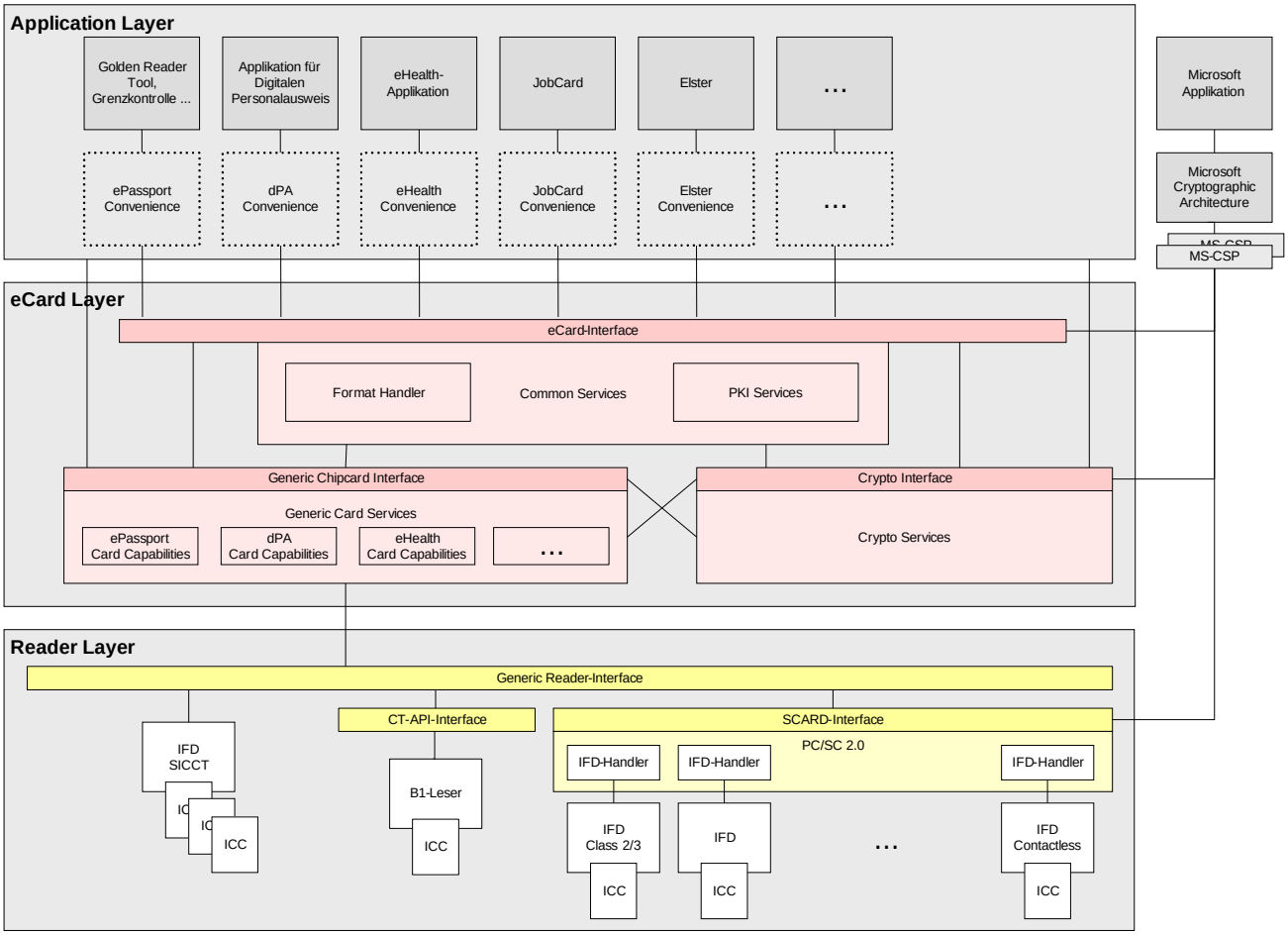


Abbildung 10: eCard-Framework

.1.1 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.

.1.2 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 (ePassport, 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

.1.3 Reader Layer

Die generelle Karten- und Kartenleseransteuerung soll über eine einheitliche Schnittstelle bereitgestellt werden, durch die von der konkreten Ausprägung des Lesegerätes abstrahiert wird. Hierbei sollen unterschiedlichste Lesegeräte

(konktaktbehaftete 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 wieder finden, 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.

1.2 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 gekapsel

Anhang C (normativ)

SICCT Status Codes

1 SICCT Status Code Vergleichstabelle

Complete SICCT Status Code Overview		
SW1 SW2	Meaning	
Warning		
'6200'	SICCT SELECT CT MODE	Warning - Specified Mode already set.
'6200'	SICCT TERMINATE COMMAND	Warning - Specified command not found
'6200'	SICCT INIT CT SESSION	Warning - CT Session already set.
'6200'	SICCT CLOSE CT SESSION	Warning - No pending / open CT Session found.
'6200'	SICCT REQUEST ICC	Warning: No card presented in time
'6201'	SICCT REQUEST ICC	Warning: Reset successfu, ICC alrady inserted and activated.
'620x'	Sequence Number Data Object Command Sequence Status Word	Sequence Number found: command in processing state '0x'
'63Cx'	SICCT PERFORM VERIFICATION	Verification unsuccessful. x = number of possible retries
General Execution Errors		
'6400'	SICCT TERMINATE COMMAND	Not terminated at preprocessing phase
'6400'	SICCT INIT CT SESSION	Error - Opening CT Session was not successful
'6400'	SICCT CLOSE CT SESSION	Error - Closing CT Session was not successful
'6400'	SICCT RESET CT / CC	Reset not successful
'6400'	SICCT REQUEST ICC	Reset not successful
'6400'	SICCT GET STATUS	Execution Error
'6400'	SICCT SET STATUS	Execution Error
'6400'	SICCT INPUT	Nor or incomplete input in time
'6400'	SICCT PERFORM VERIFICATION	Nor or incomplete input in time
'6400'	SICCT MODIFY VERIFICATION DATA	Nor or incomplete input in time
'6400'	SICCT CT Download INIT	Error
'6400'	SICCT CT Download DATA	Error
'6400'	SICCT CT Download FINISH	Error
'6401'	SICCT TERMINATE COMMAND	Not terminated at processing phase

Complete SICCT Status Code Overview		
SW1 SW2	Meaning	
'6401'	SICCT REQUEST ICC	Process aborted by pressing of CANCEL key
'6401'	SICCT INPUT	Process aborted by pressing of CANCEL key
'6401'	SICCT PERFORM VERIFICATION	Process aborted by pressing of cancel key
'6401'	SICCT MODIFY VERIFICATION DATA	Process aborted by pressing of cancel key
'6402'	SICCT TERMINATE COMMAND	Not terminated at postprocessing phase
'6402'	SICCT MODIFY VERIFICATION DATA	Process unsuccessful, new PIN not identical
'640x'	Sequence Number Data Object Command Sequence Status Word	Sequence Number found: processing state found at stage '0x' - state cannot be changed.
'64A1'	SICCT RESET CT / CC	No Card present
'64A1'	SICCT GET STATUS	No Card present
'64A1'	SICCT PERFORM VERIFICATION	No Card present
'64A1'	SICCT MODIFY VERIFICATION DATA	No Card present
'64A2'	SICCT PERFORM VERIFICATION	Card not activated
'64A2'	SICCT MODIFY VERIFICATION DATA	Card not activated
'6501'	SICCT CT Download INIT	Memory Failure
'6501'	SICCT CT Download DATA	Memory Failure
'6501'	SICCT CT Download FINISH	Memory Failure
General Checking Errors		
'6700'	ALL SICCT Commands	Wrong (Command Length) length
		Too less / many Data (Objetcs) given within command.
'6900'	SICCT INIT CT SESSION	Command not allowed. Open CT Session found.
'6900'	SICCT CLOSE CT SESSION	Command not allowed. No open CT Session
'6900'	SICCT GET STATUS	Command not allowed ▪ Cardterminal Session: Admin Access Rights required. ▪ No open CT Session
'6900'	SICCT SET STATUS	Command not allowed ▪ Cardterminal Session: Admin Access Rights required. No open CT Session
'6900'	SICCT CT Download INIT	Command not allowed ▪ No open CT Session ▪ Cardterminal busy, Downlaod Session cannot start.
'6900'	SICCT CT Download DATA	Command not allowed ▪ No pending Download Session.

Complete SICCT Status Code Overview		
SW1 SW2	Meaning	
'6900'	SICCT CT Download FINISH	Command not allowed ▪ No pending Download Session.
'6930'	SICCT REQUEST ICC	Command with timer not supported. ▪ Terminal does not support the timer option.
'6930'	SICCT EJECT ICC	Command with timer not supported. Terminal does not support the timer option.
'6930'	SICCT INPUT	Command with timer not supported. Terminal does not support the timer option.
'6930'	SICCT OUTPUT	Command with timer not supported. Terminal does not support the timer option.
'6930'	SICCT PERFORM VERIFICATION	Command with timer not supported. Terminal does not support the timer option.
'6930'	SICCT MODIFY VERIFICATION DATA	Command with timer not supported. Terminal does not support the timer option.
'6940'	SICCT REQUEST ICC	Command with Display not supported.
'6940'	SICCT INPUT	Command with Display not supported.
'6940'	SICCT OUTPUT	Command with Display not supported.
'6940'	SICCT PERFORM VERIFICATION	Command with Display not supported.
'6940'	SICCT MODIFY VERIFICATION DATA	Command with Display not supported.
'6941'	SICCT REQUEST ICC	Functional Unit (Display, Slot, Keypad) busy / not available. ▪ The addressed FU is busy and at the moment not available.
'6941'	SICCT INPUT	Functional Unit (Display, Slot, Keypad) busy / not available. ▪ The addressed FU is busy and at the moment not available.
'6941'	SICCT OUTPUT	Functional Unit (Display, Slot, Keypad) busy / not available. ▪ The addressed FU is busy and at the moment not available.
'6941'	SICCT PERFORM VERIFICATION	Functional Unit (Display, Slot, Keypad) busy / not available. ▪ The addressed FU is busy and at the moment not available.
'6941'	SICCT MODIFY VERIFICATION DATA	Functional Unit (Display, Slot, Keypad) busy / not available. ▪ The addressed FU is busy and at the moment not available.
'6942'	SICCT REQUEST ICC	Selected Character Set not supported. ▪ The addressed display does not support the selected character set.

Complete SICCT Status Code Overview		
SW1 SW2	Meaning	
'6942'	SICCT EJECT ICC	Selected Character Set not supported. ▪ The addressed display does not support the selected character set.
'6942'	SICCT INPUT	Selected Character Set not supported. ▪ The addressed display does not support the selected character set.
'6942'	SICCT OUTPUT	Selected Character Set not supported. ▪ The addressed display does not support the selected character set.
'6942'	SICCT PERFORM VERIFICATION	Selected Character Set not supported. ▪ The addressed display does not support the selected character set.
'6942'	SICCT MODIFY VERIFICATION DATA	Selected Character Set not supported. The addressed display does not support the selected character set.
'6A00'	ALL SICCT Commands	Wrong parameters P1, P2 ▪ P1 addresses invalid Functional Unit. ▪ Invalid FUI DO referenced by P1 (command data) P2 specifies not supported value
'6A80'	SICCT GET STATUS	Invalid Data Object ▪ Incorrect parameters (data object) in the command data field
'6A80'	SICCT SET STATUS	Invalid Data Object ▪ Incorrect parameters (data object) in the command data field
'6A88'	SICCT GET STATUS	Missing Data Object
		Referenced data or reference data not found
'6A88'	SICCT SET STATUS	Missing Data Object
		Referenced data or reference data not found
'6C00'	ALL SICCT Commands	Wrong length Le
'6D00'	ALL SICCT Commands	Wrong instruction
'6E00'	ALL SICCT Commands	Class not supported
'6F00'	Sequence Number Data Object Command Sequence Status Word	Sequence Number not found: no information given on processing state
'6F00'	SICCT REQUEST ICC	Communication with ICC not possible
'6F00'	SICCT RESET CT / CC	Communication with ICC not possible
'6F00'	SICCT CT Download INIT	Communication with CT not possible.
'6F00'	SICCT CT Download DATA	Communication with CT not possible.

Complete SICCT Status Code Overview		
SW1 SW2	Meaning	
'6F00'	SICCT CT Download FINISH	Communication with CT not possible.
Normal Processing		
'9000'	Sequence Number Data Object Command Sequence Status Word	Sequence Number found: Command queued and not in processing state
'9000'	SICCT SELECT CT MODE	Mode Selection was successful ▪ The specified CT mode has been selected. In order to activate the mode the cardterminal has to be reset.
'90xx'	SICCT TERMINATE COMMAND	Command successful ▪ The specified command with the given sequence number has been terminated at stage 'xx'
'9000'	SICCT TERMINATE COMMAND	Command successful ▪ Command terminated before execution.
'9000'	SICCT INIT CT SESSION	Opening CT Session was successful ▪ The specified CT session has been opened.
'9000'	SICCT CLOSE CT SESSION	Closing CT Session was successful ▪ The specified CT session has been closed.
'9000'	SICCT RESET CT / ICC	Reset successful ▪ Cardterminal and all functional units successfully set to reset state. All chipcards deactivated.
'9000'	SICCT RESET CT / ICC	Reset successful, synchronous ICC ▪ Synchronous chipcard detected, activated and successfully set to demanded reset state.
'9000'	SICCT RESET CT / ICC	Reset successful ▪ Cardterminal and all functional units successfully set to reset state. All chipcards deactivated.
'9000'	SICCT RESET CT / ICC	Reset successful, synchronous ICC ▪ Synchronous chipcard detected, activated and successfully set to demanded reset state.
'9000'	SICCT EJECT ICC	Command successful ▪ Chipcard deactivated but not removed.
'9000'	SICCT GET STATUS	Command successful ▪ Data Object query successful.
'9000'	SICCT SET STATUS	Command successful ▪ Data Object adjusted: Value set.
'9000'	SICCT INPUT	Command successful ▪ Captured Input data returned within body of command response.
'9000'	SICCT OUTPUT	Command successful ▪ Output data processed by addressed FU

Complete SICCT Status Code Overview		
SW1 SW2	Meaning	
'9000'	SICCT PERFORM VERIFICATION	Command successful ▪ Note: Chipcard generated status word in case the PIN verification was successful.
'9000'	SICCT MODIFY VERIFICATION DATA	Command successful ▪ Note: Chipcard generated status word in case the PIN verification was successful.
'9000'	SICCT CT Download INIT	Command successful ▪ Download Initialisation was successful.
'9000'	SICCT CT Download DATA	Command successful ▪ Transmission of Download Data Object was successful.
'9000'	SICCT CT Download FINISH	Command successful ▪ Download Termination was successful.
'9001'	SICCT TERMINATE COMMAND	Command terminated at preprocessing phase
'9001'	SICCT RESET CT / ICC	Reset successful, asynchronous ICC ▪ Asynchronous chipcard detected, activated and successfully set to demanded reset state.
'9001'	SICCT REQUEST CT / ICC	Reset successful, asynchronous ICC Asynchronous chipcard detected, activated and successfully set to demanded reset state.
'9001'	SICCT EJECT ICC	Command successful Chipcard deactivated and removed.
'9002'	SICCT TERMINATE COMMAND	Command successful Command terminated at processing phase
'9003'	SICCT TERMINATE COMMAND	Command successful Command terminated at postprocessing phase
'900F'	SICCT TERMINATE COMMAND	Command successful Command abortion forced.