

SICCT **Secure Interoperable ChipCard Terminal**

ERRATA zu [SICCT_123]

Datei: [SICCT_123_Errata_1_0_20210505]
Version: 1.0
Datum: 05.05.2021
Editoren: Frank Osthoff / Jürgen Atrott

Hinweise zum Dokumentenstand und Haftungsausschluss befinden sich in Kapitel 2.

Autoren

Jürgen Atrott	TÜV Informationstechnik GmbH
Frank Osthoff	Ingenico Healthcare GmbH
Alfred Fiedler	gematik GmbH
Ursel Hagedorn	achelos GmbH
Marcel Stienemeier	achelos GmbH
Sebastian Schraml	Cherry GmbH

© 2005 - 2021, TeleTrusT

All product names are trademarks, registered trademarks, or service marks of their respective owners.

Vorwort	5
1 Anwendungsbereich	5
2 Anmerkungen zum Dokumentenstand	5
2.1 DOKUMENTENSTAND UND HAFTUNGSAUSSCHLUSS	5
2.2 ERRATA-BESCHREIBUNG	5
3 Änderungen und Korrekturen	6
5.13 COMMAND SICCT REQUEST ICC	10
5.13.1 Funktion	10
5.13.2 Anwendungsbedingungen	11
5.13.3 Command Structure	12
5.13.4 Data Objects	13
5.13.5 Response Structure	14
5.13.6 Status-Codes SW1SW2	14
5.18.6 COMMAND SICCT OUTPUT	17
5.20 COMMAND SICCT MODIFY VERIFICATION DATA	20
6.2.3.1 DIENSTANFRAGE (SERVICE DISCOVERY)	23
6.4.1.1 TLS	23
6.4.2 ZERTIFIKATE	24
10 Referenzen	25
ÄNDERUNGSNACHWEISE DES ERRATA-DOKUMENTS	25

Vorwort

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

1 Anwendungsbereich

Das Dokument beschreibt bekannte und von der SICCT-WG bearbeitete Errata-Korrekturen zu der Spezifikation "Secure Interoperable Chipcard Terminal" [SICCT_123] der Version V1.2.3 vom 30.09.2016.

Geänderte Inhalte zur [SICCT_123] werden in diesem Dokument auszugsweise dargestellt und wurden im Text **gelb** markiert.

2 Anmerkungen zum Dokumentenstand

2.1 Dokumentenstand und Haftungsausschluss

Die veröffentlichte Version 1.2.3 der SICCT-Spezifikation [SICCT_123] gibt den momentanen Stand der Arbeiten wieder, aus dem Zielsetzung und Lösungsansätze ersichtlich sind. Das Dokument [SICCT_123] in der jetzigen Form kann als Basis für andere Spezifizierungsarbeiten oder technische Entwicklungen dienen.

Für die Richtigkeit, Vollständigkeit und Aktualität der Inhalte können die Autoren keine Gewähr übernehmen.

Die folgenden Angaben fokussieren einige wichtige Punkte, die zum Zeitpunkt der Erstellung als bekannt galten. Die Gesamtheit der Korrekturen erhebt keinerlei Anspruch auf Vollständigkeit mit der Konsequenz, dass das Dokument an jeder Stelle grundlegend oder im Detail noch späteren Erweiterungen unterworfen sein kann.

2.2 Errata-Beschreibung

Das vorliegende Dokument beinhaltet die SICCT-Basispezifikation sowie alle bis zum Ausgabedatum bekannten und von der TeleTrusT-AG "SICCT" bearbeiteten Errata-Korrekturen zu der Spezifikation "Secure Interoperable Chipcard Terminal" [SICCT_121] in der Version V1.2.3 vom 30.09.2016. Zukünftige ERRATA-Dokumente erhalten einen eigenen Versionsbezug (Version X_Y_Z) sowie eine Versionsreferenz zur SICCT-Basispezifikation (z.B. [SICCT_123_Errata_X_Y_Z_YYYYMMDD]) im Dateinamen.

3 Änderungen und Korrekturen

5.5.10.17 Display Capabilities Data Object

Korrektur (19.02.2021): 5.5.10.17 Ergänzung der Beschreibung des Display Capabiliy Data Objects [DSPLC DO] um ein Beispiel zur Klarstellung, wie die Fähigkeit zur Darstellung multipler Character Sets von einer Functional Unit Display abgefragt und zurückgemeldet werden können.

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

Display Capabilities Data Object (DSPLC DO)					
TAG	'67'	One byte tag according SICCT-Specifications:			
		Tag coding according ASN.1 BER see ...			
		BER-Coding : Application context, constructed, Tag-Number = 7 ('07')			
LEN	LEN coding see ...				
	one or two bytes coding - LEN in the range of : 0 <= LEN <= 255				
	'00' ... '7F'	0 <= LEN <= 127		One byte coding	
'81'	'80' ... 'FF'	128 <= LEN <= 255		Two byte coding	
VALUE	Interface Device Protocol Options				
	Tag	LEN			
	'80'	'01'	Number of visible characters per line		
			b8...b1	Number of visible characters per line: '10' <= m <= 'FF'	
			'00'	none	
	'81'	'01'	Number of visible lines		
			b8 ..b1	Number of visible lines: '02' <= n <= 'FF'	
			'00'	none	
	'82'	'01'	Number of virtual characters per line (panning)		
			b8...b1	Number of virtual characters per line (panning): '10' <= p <= 'FF'	
			'00'	none	
	'83'	'01'	Number of virtual lines 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
			b6..b8	Number of LEDs: '0' <= u <= '7'	
	'00'	None			
	One or sequence of Character Set Data Objects				
	'85'	'01'	Character Set Data Object		
			b8..b1	1st supported Character Set	see ...
	'85'	'01'	Character Set Data Object		
			⋮	2nd supported Character Set	see ...
	⋮	⋮	⋮	⋮	
	'85'	'01'	Character Set Data Object		
		⋮	<nth> supported Character Set	see ...	

Beispiel - SICCT GET STATUS an FU(Display) zur Abfrage des [DSPLC DO]

```
[CAPDU] = [80 13 40 67 00] SICCT GET STATUS, address FU '40', query [DSPLC DO]
[RAPDU] = [ [DSPLC DO] | [SW1SW2] ]
[RAPDU] = [ [67 18 [ [80 01 17]
                [81 01 06]
                [82 01 17]
                [83 01 06]
                [84 01 91]
                [85 01 00] display supports ISO 646 char. set
                [85 01 30] display add. supports ISO 18004 QR
                [85 01 40] ] ] ] display add. supports ISO 16022 DataMatrix
                [9000] ]
```

5.5.10.20 Character Set Data Object

Korrektur (19.02.2021): 5.5.10.20 Ergänzung der Beschreibung des Character Set Data Objects [CS DO] um zwei weitere Werte zur Codierung einer Nachricht zur Darstellung als grafischen 2-Code (QR- bzw. DataMatrix-Code).

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

Character Set Data Object [CS DO]			
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'		Latin-4			

Character Set Index Value	Normative Reference	Description
	ISO / IEC 8859-4	Die Codetabelle dieser Kodierung enthält einige Zeichen der Sprachen Estnisch, Lettisch und Litauisch. Vergleichen Sie diese Kodierung auch mit ISO 8859-10, deren Codetabelle sehr ähnlich ist.
'05'	ISO / IEC 8859-5	8859-5 Die Codetabelle dieser Kodierung enthält kyrillische Zeichen. Sie deckt weitgehend die Sprachen Bulgarisch, Mazedonisch, Russisch, Serbisch und Ukrainisch ab.
'06'	ISO / IEC 8859-6	8859-6 Die Codetabelle dieser Kodierung enthält Zeichen arabischer Schrift. Die Darstellung der Zeichen in der folgenden Tabelle ist jedoch "abstrakt", da die Zeichen in der Schriftpraxis variieren, je nachdem, ob sie am Anfang, in der Mitte oder am Ende eines Wortes oder einzeln stehen. Arabisch zeichnet sich weiterhin dadurch aus, dass die Schriftrichtung von rechts nach links ist.
'07'	ISO / IEC 8859-7	8859-7 Die Codetabelle dieser Kodierung enthält die Zeichen der neugriechischen Schrift.
'08'	ISO / IEC 8859-8	8859-8 Die Codetabelle dieser Kodierung enthält die Zeichen der neugriechischen Schrift.
'09'	ISO / IEC 8859-9	Latin-5 Diese Kodierung ist speziell für Türkisch gedacht. Die Codetabelle basiert auf ISO 8859-1, enthält jedoch anstelle der isländischen Sonderzeichen türkische Zeichen.
'0A'	ISO / IEC 8859-10	Latin-6 Die Codetabelle dieser Kodierung enthält speziell Zeichen für die Sprachen Grönländisch (Inuit) und 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 (<i>Latin-4</i> eher baltisch) und ISO 8859-10 (<i>Latin-6</i> eher nordisch) denen einige Zeichen fehlten.
'0E'	ISO / IEC 8859-14	Latin-8 ISO 8859-14 versucht alle Sonderzeichen keltischer und einiger anderer westeuropäischer Sprachen 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.
'30'	ISO 18004	ISO 18004 für die Darstellung des übergebenen Textes als QR-Code
'40'	ISO 16022	ISO 16022 für die Darstellung des übergebenen Textes als Data-Matrix-Code

5.5.10.21 SICCT Message-To-Be-Displayed Data Object

Korrektur (19.02.2021): 5.5.10.21 Ergänzung des SICCT Message-To-Be-Displayed Data Objects zur Darstellung als grafischen 2-Code (QR- bzw. DataMatrix-Code).

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

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

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

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

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

SICCT Message-To-Be-Displayed Data Object (SMTBD DO)					
TAG	'A0'		One byte tag according 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				
	'00' ... '7F'		0 <= LEN <= 127	One byte coding	
	'81'	'80' ... 'FF'	128 <= LEN <= 255	Two byte coding	
	'82'	'01' .. 'FF'	'00' .. 'FB'	256 <= LEN <= MAX	Three byte coding
VALUE	One Character Set DO and corresponding Message (OCTETSTRING DO)				
	Character Set Data Object		See 5.5.10.20		
	'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'		0 <= L <= 127	One byte coding
		'81'	'80' ... 'FF'	128 <= LEN <= 255	Two byte coding
		'82'	'01' .. 'FF'	'00' .. 'FB'	256 <= LEN <= MAX
Byte Sequence formatted according the selected character set giving the message to be displayed.					
MAX = abhängig von der gesamten SICCT Kommandolänge					

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

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

5.13 Command SICCT REQUEST ICC

Korrektur (21.04.2021): Editorische Korrekturen und Eliminierung doppelter Unterkapitel in [SICCT_123#5.13]:

Command Structure: 5.13.7 ersetzt 5.13.3, Data Objects: 5.13.8 ersetzt 5.13.4,

Response Structure: 5.13.9 ersetzt 5.13.5, Status-Codes SW1SW2: 5.13.10 ersetzt 5.13.6.

Nachfolgend der komplette Textabschnitt SICCT REQUEST ICC, ohne inhaltliche Änderungen.

5.13.1 Funktion

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

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

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

Folgende Optionen können generell gewählt werden:

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

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

- 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 Wirkungsbereich beinhalten, muss die Parametrierung der Chipkartenkontaktiereinheit sowie nachfolgend den

elektrisch / logischen Reset der im Zugriff befindlichen Chipkarte(n) bedeuten. Das CT-Kommando muss stets einen Cold - Reset an einer ICC – Funktionseinheit bewirken.

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

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

Ausführungsphasen SICCT REQUEST ICC				
Phase		Stage	option	Description
preprocessing phase	Vorbereitungsphase	1		<ul style="list-style-type: none"> ▪ display message ▪ perform user dialog ▪ wait for card insertion or user abort via keypad
processing phase	Ausführungsphase	2		<ul style="list-style-type: none"> ▪ check CT / ICC / PICC state ▪ Reset CT / ICC / PICC ▪ Perform PPS with ICC / PICC
postprocessing phase	Nachbereitungsphase	3		<ul style="list-style-type: none"> ▪ return ICC / PICC PPS Response ▪ return CT / ICC / PICC Reset Information ▪ send return value

5.13.2 Anwendungsbedingungen

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

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

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

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

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

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

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

5.13.3 Command Structure

SICCT Kommando	Kodierung C-APDU						
	CLA	INS	P1	P2	[Lc]	[Data]	[Le]
SICCT REQUEST ICC	'80'	'12'	Functional Unit	Command Qualifier	Length Command Data	Command Data	Length Requested Data
	<ul style="list-style-type: none"> ▪ CLA = Class ▪ INS = Instruction ▪ P1, P2 = Parameter 1 and 2 ▪ Lc = Length of command data field ▪ Le = Length of expected ▪ SW1, SW2 = Status Bytes 				Case 1 (no cmd data, no rsp data) : no Lc, no Le		
					Case 2 (no cmd data, rsp data): no Lc, Le=1-256 bytes		
					Case 3 (cmd data, no rsp data): Lc=1-255 Bytes no Le		
				Case 4 (cmd data, rsp data): Lc=1-255Bytes Le=1-256 Bytes			

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

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

P2	Command Qualifier:	
	bit8..bit5	Request handling Instructions
		SICCT with no display: don't care
		SICCT with display
		'0'
'F'	No message to be displayed ¹	

¹ Suppress all internal ('Abort', ..) and external messages.

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

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

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

5.13.4 Data Objects

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

Data Object	COMMAND RESPONSE	Description	Remarks
FUI DO	CMD	Functional Unit Index Data Object (Referenced Coded FU)	see 5.5.10.9
FU CON DO	CMD	Functional Unit Context Data Object	see 5.5.10.10
APPL DO	CMD	Text / display message.	see 5.5.10.19
SMTBD DO	CMD	Constructed TLV-DO containing one Character Set and one Application Label DO.	see 5.5.10.21
ATR DO	RESP	Answer To Reset Information (Cold ATR)	see 5.5.10.4
HB DO	RESP	Part of ATR Information: Historical Byte Data Object	see 5.5.10.5
WT DO	CMD	Max. Waiting Time in seconds	see 5.5.10.22

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

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

5.13.5 Response Structure

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

5.13.6 Status-Codes SW1SW2

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

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

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

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

SW1SW2	Addressed Functional Unit		Priority Priorität	Specification	Meaning / Bedeutung
	P1				
	Of Type				
	ICC <n>	RFID <n>			
'6200'	x	x	11	Warning: No card presented in time	<ul style="list-style-type: none"> No card presented within specified time. / Innerhalb der vorgegebenen Zeit wurde Keine Karte eingeführt
'6201'	x	x	11	Warning: Reset successful	<ul style="list-style-type: none"> Card (ICC) already present and activated. / Karte (ICC) war bereits eingeführt und aktiviert.
'6400'	x	x	12	Reset not successful	<ul style="list-style-type: none"> Error occurred during chipcard reset. In case the ICC / PICC does not react on reset or the ICC / PICC reacts on reset but does not send correct or incomplete reset information (ATR), or the ICC / PICC does not provide any terminal supported protocol. State of addressed chipcard set to deactivated state..
'6401'	x	x	12	Process aborted by pressing of CANCEL key	<ul style="list-style-type: none"> Process aborted by pressing of CANCEL key. / Der Prozeß via ABRUCH-Taste abgebrochen
'64A1'	x	x	12	No Card present	<ul style="list-style-type: none"> Addressed ICC Interface / slot did not contain or contact a chipcard / RFID token.
'64A2'	x	x	11	Card not activated	<ul style="list-style-type: none"> "Warm reset or PPS procedure with given PPSR DO not processed because card not activated"
'6700'	x	x	3	Wrong (command) length	<ul style="list-style-type: none"> Wrong Lc: Inconsistent command body or no Command Data field supported.
'6900'	x	x	9	Command not allowed.	<ul style="list-style-type: none"> The terminal cannot perform the command because has there is no valid (open) CT Session TLS connection
'6941'	x	x	10	Functional Unit busy / not available	<ul style="list-style-type: none"> At the moment FU_CT is busy / unavailable. Funktionseinheit FU_CT ist z.Zt. belegt / nicht verfügbar
'6942'	x	x	8	Selected Character Set not supported.	<ul style="list-style-type: none"> The addressed display does not support the selected character set. / Die adressierte Anzeige unterstützt den gewählten Zeichensatz nicht.
'6A00'	x	x	5	Wrong parameters P1, P2	<ul style="list-style-type: none"> P1 addresses invalid Functional Unit. In case of Referenced Coding – Invalid / ungültiges FUI DO referenced by P1 (command data). P2 specifies not supported value
'6A80'	x	x	6	Invalid Data Object	<ul style="list-style-type: none"> Incorrect data objects / parameters in the command data field In case of Referenced Coding – The data field contains an invalid DO / Das Datenfeld enthält ein unzulässiges DO. Invalid FUI DO / Unzulässiges FUI DO
'6A88'	x	x	7	Missing Data Object	<ul style="list-style-type: none"> In case of Referenced Coding – The data field does not contain a FUI DO / Das Datenfeld enthält kein FUI DO.
'6A89'	x	x	8	Too many Data Objects	<ul style="list-style-type: none"> Too many Data Objects (FUI DO, ...) of same type. Es sind zu viele Datenobjekte gleichen Typs (FUI DO, ...) enthalten
'6C00'	x	x	4	Wrong Length Le	<ul style="list-style-type: none"> Wrong Le / falscher Wert Le:
'6D00'	x	x	2	Wrong Instruction	<ul style="list-style-type: none"> Wrong / unbekanntes Instruction-Byte
'6E00'	x	x	1	Class not supported	<ul style="list-style-type: none"> Falsches / unbekanntes Class-Byte

'6F00'	x	x	11	Communication with ICC not possible	<ul style="list-style-type: none"> ▪ the ICC / PICC reacts on reset but does not send correct or incomplete protocol parameter selection (PPS) information ▪ the ICC / PICC reacts on reset but generates protocol errors.
'9000'	x	x	11	Reset successful, synchronous ICC	<ul style="list-style-type: none"> ▪ Synchronous chipcard detected, activated and successfully set to demanded reset state.
'9001'	x	x	12	Reset successful, asynchronous ICC	<ul style="list-style-type: none"> ▪ Asynchronous chipcard detected, activated and successfully set to demanded reset state.

5.18.6 Command SICCT OUTPUT

Korrektur (19.02.2021): 5.18.6 Ergänzung der Funktionalität sowie die Menge der Statusworte des SICCT OUTPUT Kommandos zur Darstellung von Nachrichten als grafischer 2-Code (QR- bzw. Data-Matrix-Code). Angabe eines Aufrufbeispiels.

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

:

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

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

5.18.6 Status-Codes SW1SW2

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

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

SW1SW2	Addressed Functional Unit			Priority Priorität	Specification	Meaning / Bedeutung
	P1					
	Of Type					
'40' Display	'60' Printer	'7x' Biometric Sensor				
'6200'				11		
'6400'	x	x	x	12	Execution Error	<ul style="list-style-type: none"> An undetermined error occurred during execution of the command. / Unbestimmter Fehler während der Kommandoausführung.
					Output not successful	<ul style="list-style-type: none"> Error occurred during output operation. Rendered 2D-code image is too big
'6401'	x	x	x		Process aborted by pressing of cancel key	<ul style="list-style-type: none"> Abort condition : User canceled input operation. / Abbruchbedingung: Anwender brach Ausführung der Ausgabe ab.
'6410'	x	x	x	11	Functional Unit (FU) unable to process the output data	<ul style="list-style-type: none"> the addressed FU does not support the quantity or type of provided output data
'6700'	x	x	x	3	Wrong (command) length	<ul style="list-style-type: none"> Wrong Lc: Inconsistent command body or command data (message, ...) is too long.
'6900'	x	x	x	9	Command not allowed.	<ul style="list-style-type: none"> The terminal cannot perform the command because there is no valid (open) CT Session (Admin Access Rights required.) TLS connection

SW1SW2	Addressed Functional Unit			Priority Priorität	Specification	Meaning / Bedeutung
	P1					
	Of Type					
	'40' Display	'60' Printer	'7x' Biometric Sensor			
'6930'	x	x	x		Command with timer not supported	<ul style="list-style-type: none"> Terminal does not support the timer option. / Das Terminal unterstützt die Timer-Option nicht.
'6940'	x	x	x	8	Command with Display not supported	<ul style="list-style-type: none"> Command with not supported for addressed FU (display, printer, biometric sensor)
'6941'	x	x	x	10	Functional Unit busy / not available	<ul style="list-style-type: none"> At the moment FU_CT or addressed FU is busy / unavailable. Funktionseinheit FU_CT oder die adressierte FU ist z.Zt. belegt / nicht verfügbar
'6942'	x	x	x	8	Selected Character Set not supported	<ul style="list-style-type: none"> The addressed FU (display, printer, ...) does not support the selected character set regarding 2D-Code are not supported / Die adressierte FU unterstützt nicht den angegebenen Zeichensatz bzw. es werden keine 2D-Code unterstützt
'6A00'	x	x	x	5	Wrong parameters P1, P2	<ul style="list-style-type: none"> P1 addresses invalid Functional Unit. In case of Referenced Coding – Invalid / ungültiges FUI DO referenced by P1 (command data). P2 specifies not supported value
'6A80'	x	x	x	6	Invalid Data Object	<ul style="list-style-type: none"> Incorrect data objects / parameters in the command data field In case of Referenced Coding – The data field contains an invalid DO / Das Datenfeld enthält ein unzulässiges DO. Invalid FUI DO / Unzulässiges FUI DO In case e.g. output date (Display message, ..) too long.
'6A88'	x	x	x	7	Missing Data Object	<ul style="list-style-type: none"> Referenced data or reference data not found In case of Referenced Coding – The data field does not contain a FUI DO / Das Datenfeld enthält kein FUI DO.
'6A89'	x	x	x	8	Too many Data Objects	<ul style="list-style-type: none"> Too many Data Objects (FUI DO) of same type. Es sind zu viele Datenobjekte gleichen Typs (FUI DO, ...) enthalten
'6C00'	x	x	x	4	Wrong Length Le	<ul style="list-style-type: none"> Wrong Le / falscher Wert Le: Le has to absent / Le darf nicht angegeben sein.
'6D00'	x	x	x	2	Wrong Instruction	<ul style="list-style-type: none"> Wrong / unbekanntes Instruction-Byte
'6E00'	x	x	x	1	Class not supported	<ul style="list-style-type: none"> Falsches / unbekanntes Class-Byte
'6F00'	X	-	-	11	Communication with CT not possible	<ul style="list-style-type: none"> In case the terminal detects general protocol errors or the terminal detects errors within CT Session or TLS connection

SW1SW2	Addressed Functional Unit			Priority Priorität	Specification	Meaning / Bedeutung
	P1					
	Of Type					
	'40' Display	'60' Printer	'7x' Biometric Sensor			
	x	x	x	11	Communication with addressed FU not possible	<ul style="list-style-type: none"> the addressed FU has reported a general error.
'9000'	x	x	x	12	Command successful	<ul style="list-style-type: none"> Output data processed by addressed FU / Ausgabedaten wurden von der adressierten FU verarbeitet.

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

Ausgabe der Nachricht " the brown fox jumps over the lazy dogs back. " am Terminal-Display für max. 50 Sekunden als QR-Code.

[CAPDU] = [[SICCT OUTPUT] | [SMTBD DO] | [WT DO]]

[CAPDU] = [[80 17 40 00] 38

[A0 33 [

[85 01 30]

[04 2E

207468652062726F

776E20666F78206A

756D7073206F7665

7220746865206C61

7A7920646F677320

6261636B2E20] |

[80 01 32]]]

[RAPDU] = [SW1SW2] = [9000]

5.20 Command SICCT MODIFY VERIFICATION DATA

Korrektur (19.02.2021): 5.20.6 Korrektur bzw. Ergänzung des in [SICCT_123] fehlenden Statuswortes '6402' für das Kommando SICCT MODIFY VERIFICATION DATA. Angabe eines Aufrufbeispiels.

:

5.20.6 Status-Codes SW1SW2

Nach erfolgreicher Ausführung des SICCT MODIFY VERIFICATION DATA - Kommandos muss das erste Byte des Statuswortes SW1 den Wert '90' enthalten.
Das Statusbyte SW2 kann weitere Umstände in Abhängigkeit zu Kommandoparametern qualifizieren.

SW1SW2	Addressed Functional Unit P1		Priority Priorität	Specifica- tion	Meaning / Bedeutung
	Of Type				
	ICC<n>	RFID<n>			
'63Cx'	x	x		Verification unsuccessful. x = number of possible retries	Note: Chipcard generated status word in case the PIN verification failed.
'6400'	x	x	12	Nor or incomplete input in time	Abort condition raised by timeout (during stage 1).
'6401'	x	x	12	Process aborted by pressing of CANCEL key	Abort: User canceled input operation by pressing of CANCEL key. / Abbruch - Der Benutzer beendete die Eingabe via ABBRUCH-Taste.
'6402'	x	x	12	Process unsuccessful, new PIN not identical	Fail: The conformation of the PIN entry was unsuccessful. / Fehler – Die Bestätigung der PIN-Eingabe war nicht erfolgreich.
'64A1'	x	x		No Card present	Verification not performed, because no card present Addressed ICC Interface / slot did not contain or contact a chipcard / RFID token.
'64A2'				Card not activated	Verification not performed, because no activated card.
'6700'	x	x	3	Wrong (command) length	Wrong Lc: Inconsistent command body or no Command Data field supported. Message too long.
'6900'	x	x	9	Command not allowed.	The terminal cannot perform the command because has there is no valid (open) CT Session TLS connection
'6930'	x	x	8	Command with timer not supported.	Terminal does not support the timer option. / Das Terminal unterstützt die Option Zeitüberwachung nicht.
'6940'	x	x	8	Command with Display not supported.	Terminal does not support the display option. / Das Kommando unterstützt die Option Display (Anzeige) nicht.
'6941'	x	x	9	Functional Unit (FU) busy / not available.	The addressed FU (display, slot, ..) is busy at the moment or not available. / Die adressierte Funktionseinheit (Anzeige, Konaktiereinheit, ..) ist im Moment belegt und ist nicht verfügbar.
'6941'	x	x	10	Functional Unit busy / not available	At the moment FU_CT is busy / unavailable. Funktionseinheit FU_CT ist z.Zt. belegt / nicht verfügbar
'6942'	x	x	8	Selected Character Set not supported.	The addressed display does not support the selected character set. / Die adressierte Anzeige unterstützt den gewählten Zeichensatz nicht.
'6983'	x	x	12	Verification Method Blocked.	Note: Chipcard generated status word in case the PIN verification failed, because the authentication method is blocked.
'6A00'	x	x	5	Wrong parameters P1, P2	P1 addresses invalid Functional Unit. In case of Referenced Coding – Invalid / ungültiges FUI DO referenced by P1 (command data). P2 specifies not supported value Invalid combination of P1 and P2

'6A80'	x	x	6	Incorrect parameters in the command data field or Invalid Data Object	Incorrect data objects / parameters in the command data field General data error In case of CMD DO e.g. non-allowed 'INS' code. In case of Referenced Coding – The data field contains an invalid DO / Das Datenfeld enthält ein unzulässiges DO. Invalid FUI DO / Unzulässiges FUI DO Invalid FU CON DO / Unzulässiges FU CON DO
'6A88'	x	x	7	Missing Data Object	In case of Referenced Coding – The data field does not contain a FUI DO or FU CON DO/ Das Datenfeld enthält kein FUI DO oder FU CON DO
'6A89'	x	x	8	Too many Data Objects	Too many Data Objects (CMD DO, FUI DO, FU CON DO, APPL DO, SMTBD DO, WT DO) of same type. Es sind zu viele Datenobjekte gleichen Typs (CMD DO, FUI DO, FU CON DO, APPL DO, SMTBD DO, WT DO) enthalten
'6C00'	x	x	4	Wrong Length Le	Wrong Le / falscher Wert Le: If Le exists, here it has to be set to '00'.
'6D00'	x	x	2	Wrong Instruction	Wrong / unbekanntes Instruction-Byte
'6E00'	x	x	1	Class not supported	Falsches / unbekanntes Class-Byte
'6F00'	x	x	11	Communication with addressed FU not possible	In case the terminal detects general protocol errors or the terminal detects errors within CT Session or TLS connection
'9000'	x	x	12	Command successful	Note: Chipcard generated status word in case the change of Verification Data was successful.
Note: In case the SICCT-Terminal has successfully generated and sent an APDU (command to perform) to the chipcard, the returned status word has been taken from the chipcards' response. Due to a variety of cards there might be more corresponding values (SW1SW2) than '9000', '6983' and '63Cx'.					

6.2.3.1 Dienstanfrage (Service Discovery)

Korrektur (05.10.2020 / 21.04.2021): 6.2.3.1 Ergänzung der Tabelle Sicherheitsprotokolle um TLS V1.3

Sicherheitsprotokolle:					
Protokoll	TAG (hex.)	Datenlänge (Bytes)	Daten	Wert (hex.)	Beschreibung
TLS	'8A'	1	Unterstützte Protokollversion (1 Byte)	'10'	TLS 1.0 [RFC2246]
				'11'	TLS 1.0 [RFC2246] + AES-TLS Erweiterungen [RFC3268]
				'20'	TLS 1.1 [RFC4346]
				'30'	TLS 1.2 [RFC5246]
				'40'	TLS 1.3 [RFC8446]
...

- "Unverschlüsselt":
Eine "unverschlüsselte" Verbindung entspricht einer TCP/IP Verbindung ohne Anwendung weiterer Mechanismen zur Identifikation oder Authentifikation der Kommunikationspartner, sowie zur verschlüsselten Übertragung der Protokoll Daten (SICCT Kommandos und Ereignisse).
- TLS:
Wird das optionale Sicherheitsprotokoll TLS unterstützt, so muss TLS 1.2 [RFC5246] unterstützt werden. TLS 1.3 [RFC8446] kann optional unterstützt werden. Ältere TLS Cipher Suites dürfen nicht mehr unterstützt werden.
Die Aushandlung der Cipher Suite sowie deren Parameter muss im Rahmen des TLS Handshake Verfahrens während des Aufbaus der SICCT Kommandointerpretiererverbindung erfolgen. Für weitere Festlegungen siehe Kapitel 6.4.1.1 "TLS".

6.4.1.1 TLS

Korrektur (21.04.2021): 6.4.1.1 Änderung der Minimalanforderung für optionale TLS-Sicherheitsprotokolle

Für die Verschlüsselung der Kommunikation mittels TLS (Transport Layer Security) muss mindestens die Protokollversion 1.2 [RFC5246] angeboten werden. TLS 1.3 [RFC8446] kann optional unterstützt werden. Die älteren TLS Cipher Suites TLS 1.0 und TLS 1.1 dürfen nicht mehr unterstützt werden.

Welche Cipher Suites konkret durch ein SICCT-Terminal angeboten werden, ergibt sich aus dessen Konfiguration zur Erfüllung fachspezifischer Sicherheitsvorgaben (Protection Profile,

Policy).

Wird TLS implementiert und gibt es keine fachspezifischen Vorgaben bezüglich der zu unterstützenden Cipher Suites, so muss aber zumindest die Cipher Suite `TLS_RSA_WITH_AES_128_CBC_SHA` implementiert sein, um die Mindestanforderungen des TLS 1.2 Standards zu erfüllen (siehe [RFC5246#9, Mandatory Cipher Suites]).

Die Aushandlung der konkret zu verwendenden Cipher Suite und ihrer Parameter erfolgt im Rahmen des im TLS Standard definierten Handshakeverfahrens.

Beim Einsatz von TLS muss eine einseitige Authentifizierung des SICCT-Terminals gegenüber dem Client verpflichtend unterstützt werden. Das SICCT-Terminal darf optional auch eine Authentifizierung des Clients vornehmen (zwei-seitige Authentifizierung). Dementsprechend **soll** es dem SICCT-Terminal möglich sein, die Gültigkeit des Client-Zertifikats zu prüfen.

Beim Einsatz von TLS muss die Installation passender Zertifikate am SICCT-Terminal und Client möglich sein.

6.4.2 Zertifikate

Korrektur (21.04.2021): 6.4.2 Anhebung der Minimalanforderung für X.509v3-Zertifikate

Es muss möglich sein, ein nach dem X.509v3 Standard [RFC3280] ausgestelltes Zertifikat auf dem SICCT-Terminal abzulegen. Das verwendete Dateiformat zum Aufbringen bzw. zur Speicherung des Zertifikates am Terminal ist dem Terminalhersteller überlassen.

Die Standardfelder spezifizieren den Terminalhersteller näher: Herstellername, Supportadresse, Gerätetypbezeichnung.

Das Zertifikat muss als Signaturalgorithmus mindestens SHA-256/RSA mit einer Schlüssellänge von mindestens 2048 Bit verwenden.

Für die Webmanagementoberfläche und die Verbindung des Command Sets muss ein gemeinsames Zertifikat verwendet werden. Dementsprechend muss das Zertifikat für Webserverauthentifizierung (x.509v3 Standard Extensions Key Usage) verwendbar sein.

Ein Terminalzertifikat darf nicht zur Signierung anderer Zertifikate verwendet werden (Standard Extensions Basic Constraints).

Optional darf es möglich sein, das Zertifikat am SICCT-Terminal auszutauschen, um die jeweilige Vertrauensstellung an die Einsatzumgebung des SICCT-Terminals anzupassen. Das Verändern bzw. das Tauschen des Zertifikats darf nur durch den "Administrator" (siehe Benutzerrollen in Terminal Managementverfahren) erfolgen: Dementsprechend muss eine erfolgreiche Authentifizierung an der Webmanagementoberfläche bzw. im Command Set erfolgen, um die entsprechendverfahrenen Aktionen zu erlauben.

:

10 Referenzen

Korrektur (21.04.2021): Ergänzung RFC8446 TLS1.3 und Streichung alter TLS-RFCs für TLS 1.0, 1.1.

- :
[RFC 2246] The TLS Protocol Version 1.0, <http://www.ietf.org/rfc/rfc2246.txt>
- :
[RFC4346] ~~RFC 4346: The Transport Layer Security (TLS) Protocol Version 1.1; RTFM Inc.; <http://www.ietf.org/rfc/rfc4346.txt>; 2006;~~
- :
[RFC8446] RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3; RTFM Inc.; <https://tools.ietf.org/html/rfc8446.txt>; 2018;
- :
:

Änderungsnachweise des Errata-Dokuments

Errata-Version	Release-Datum	Editor	
V1.0	05.10.2020	F. Osthoff	Initial-/Draftversion für Abstimmung mit SICCT-AG
	19.02.2020	F. Osthoff	Korrekturen und Ergänzungen am Draft-Dokument nach Rückmeldungen der SICCT-AG bis zum 15.01.2021
	21.04.2021	F. Osthoff	Korrekturen und Ergänzungen am Draft-Dokument nach Rückmeldungen der SICCT-AG bis zum 15.01. und Beschlüssen vom 19.03.2021.
	05.05.2021	J. Atrott	Korrektur E-Mail-Adresse, Versionierung, Finalisierung