

CEA Standard

National Renewable Security Standard (NRSS)

CEA-679-C R-2013

October 2005



CEA
Consumer Electronics Association®

Documents
www.CE.org
Document Sharing Hub

NOTICE

Consumer Electronics Association (CEA[®]) Standards, Bulletins and other technical publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for his particular need. Existence of such Standards, Bulletins and other technical publications shall not in any respect preclude any member or nonmember of CEA from manufacturing or selling products not conforming to such Standards, Bulletins or other technical publications, nor shall the existence of such Standards, Bulletins and other technical publications preclude their voluntary use by those other than CEA members, whether the standard is to be used either domestically or internationally.

Standards, Bulletins and other technical publications are adopted by CEA in accordance with the American National Standards Institute (ANSI) patent policy. By such action, CEA does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the Standard, Bulletin or other technical publication.

Note: The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights.

By publication of this standard, no position is taken with respect to the validity of this claim or of any patent rights in connection therewith. The patent holder has, however, filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Details may be obtained from the publisher.

This document does not purport to address all safety problems associated with its use or all applicable regulatory requirements. It is the responsibility of the user of this document to establish appropriate safety and health practices and to determine the applicability of regulatory limitations before its use.

This document is copyrighted by the Consumer Electronics Association (CEA[®]) and may not be reproduced, in whole or part, without written permission. Federal copyright law prohibits unauthorized reproduction of this document by any means. Organizations may obtain permission to reproduce a limited number of copies by entering into a license agreement. Requests to reproduce text, data, charts, figures or other material should be made to CEA.

(Formulated under the cognizance of the CEA **R4.8 DTV Interface Subcommittee.**)

Published by
©CONSUMER ELECTRONICS ASSOCIATION 2013
Technology & Standards Department
www.CE.org

All rights reserved

FOREWORD

The National Renewable Security Standard provides a means for renewable security to be employed with digital consumer electronics devices such as digital television receivers and digital VCRs. Renewable security encompasses upgradeable, extensible, removable and replaceable security, and may be implemented with a variety of embedded and removable approaches that may vary in complexity, feature set and security level. This standard does not address intended applications for either Part A or Part B devices.

The National Renewable Security Standard provides two physical designs, one in part A and one in part B. Part A defines a removable and renewable security element form factor that is an extension of the ISO-7816 standard. Part B defines a removable and renewable security element based on the PCMCIA ("PC Card") form factor. The common attributes allow either an NRSS-A or NRSS-B device to provide security for applications involving pay and subscription cable or satellite television services, telephony, and all forms of electronic commerce.

The main differences between NRSS-A and NRSS-B devices are the range of capabilities and the capacity for extension. The NRSS-A interface is limited to 8 electrical contacts using serial communication, whereas the NRSS-B interface uses 68 electrical contacts and parallel communication. Potentially, the NRSS-A device could be smaller and less complex, while the NRSS-B device could be more robust and extensible.

The video distribution industries (cable TV, DBS, MMDS, etc.) are utilizing a variety of implementations for securing digital video: a) embedded, b) totally removable and replaceable, c) hybrid, and d) split.

Embedded security involves circuitry that is permanently installed within a consumer electronics host device (e.g., cable converter, DBS receiver, TV receiver).

Removable security involves circuitry that may be totally extracted from the host device; when security is removable, the consumer electronics device can be "secret-free".

A hybrid security system uses embedded circuitry but, if necessary, this circuitry may be upgraded or overridden and replaced with replaceable elements. Thus, embedded security may nonetheless be replaceable.

A split security system has both embedded circuitry that supports some security functions (e.g., descrambling) and replaceable circuitry that supports other security functions (e.g., key management).

Both NRSS-A and NRSS-B support all options except the embedded security.

The consumer electronics industry is looking for a digital standard "cable ready" television set, receiving both broadcast and cable digital services, that is secret-free and mass merchandiseable. The availability of NRSS-A and/or NRSS-B security elements allows support for the consumer electronics needs.

This discussion is provided in order to explain the environment in which this standard has been developed. This standard was developed under the auspices of the CEMA Technology & Standards/NCTA JEC National Renewable Security Standard (NRSS) Subcommittee.

(This page intentionally left blank.)

PART A CONTENTS

1 Introduction	1
2 Scope and Field of Application	1
3 Normative References	1
3.1 Normative Reference List	1
3.2 Normative Reference Acquisition	1
4 Definitions.....	2
5 General Electrical and Physical Characteristics	2
6 Exceptions and Additional Physical Characteristics	3
7 Exceptions and Additional Electrical Characteristics.....	3
7.1 High-speed Data-in	4
7.2 High-speed Clock	4
7.3 High-speed Data-out.....	4
7.4 Functional Operations on High-speed Data	5
7.5 Communications Protocol and Flow Control	5
7.5.1 CA Module Implementation of Flow Control.....	5
7.5.2 Host Implementation of Flow Control.....	5
8 Commands.....	6
9 General Commands	7
9.1 Answer-to-Reset	7
9.2 Get CA Module Configuration	7
9.3 Get CA Module Status	8
9.4 Put CE device ID	10
9.5 Get Suggested ID.....	10
9.6 Put Host Configuration/Status	10
9.7 Get Requested Homing Time	11
10 Purchase and Cancel IPPV Commands.....	11
10.1 Put Purchase Action	11
10.2 Get Number of Purchase Offers.....	12
10.3 Get Purchase Offer Data	12
10.4 Get Purchase Offer Title	13
10.5 Put Purchase Action	13
10.6 Put Purchase Cancellation	14
10.7 Get Purchase Status	14
10.7.1 CA Status Byte.....	14
10.7.2 Program Status Byte	15
10.7.3 Status Byte Interpretation.....	15
10.8 Get Special Reason Message.....	15
11 Purchase History Commands.....	16
11.1 Get Purchase Record Length	16
11.2 Get Purchase Record	16
11.3 Put Purchase Data.....	17

11.4 Get Updated Purchase Slot Info.....	17
12 Chaining Commands	17
12.1 Get Number of Chaining Blocks	17
12.2 Get Block	18
12.3 Put Block	18
13 Reportback Commands.....	18
13.1 Get Open Reportback Request	18
13.2 Get Reportback Values	19
13.3 Get Reportback Message.....	19
13.4 Put Reportback Buffer	19
13.5 Put Reportback Status	19
14 Locks and Limits Commands	20
14.1 Get PIN	20
14.2 Put PIN	20
14.3 Get PIN Action Result	21
14.4 Get Lock State.....	22
14.5 Get User Parameter	22
14.6 Put User Parameter	22
15 Service Changes and Related Commands.....	23
15.1 Put Add-Service Packet Identification (PID) Numbers.....	23
15.2 Put Remove-Service Packet Identification (PID) Numbers	23
15.3 Get Decode Status for PIDs.....	23
15.4 Put CE Device PowerState.....	24
15.5 Get Forced Service Change.....	24
15.6 Put EMM PID.....	25
16 Display and Menu Commands	25
16.1 Get Display Message Size	25
16.2 Get Display Message.....	25
16.3 Get Alpha Box Request.....	26
16.4 Get Alpha Box Parameters	26
16.5 Put Alpha Box Input	26
16.6 Get Menu Request	26
16.7 Get Menu Parameters.....	27
16.8 Put User Response.....	27
17 Mail Handling Commands	27
17.1 Get Mail Message Length	27
17.2 Get Mail Message Text	27
18 Intercard Message Commands.....	28
18.1 Get Intercard Message Sizes.....	28
18.2 Get Intercard Message	28
18.3 Put Intercard Message	28
18.4 Put Card Numbers	28
19 NVM Storage Commands	28
19.1 Put NVM Available	28
19.2 Get Allocation Size	29
19.3 Put Allocation Tag	29
19.4 Get Operation and Tag	29
19.5 Put Operation Results	29

19.6 Put Read Data	29
19.7 Get Write Data	30
20 Copy Protection Framework	30
20.1 Service Provider [Informative]	30
20.2 Back Channel [Informative]	30
20.2.1 Service Provider Processing [Informative]	30
20.2.1.1 Checking Validity of Host Information [Informative]	30
20.2.1.2 Host Revocation [Informative]	30
20.3 CP System Negotiation	31
20.4 CA Module/Host Binding	31
20.4.1 Discrete Logarithm DH	31
20.4.2 Elliptic Curve DH	31
20.4.3 Needham-Schroeder Key Exchange Protocol	32
20.4.4 Computation of Kcpss and Kvalidated_cpss	32
20.5 Copy Control Information Definition	32
20.5.1 private_CCI	33
20.5.2 APS (Analog Protection System) Information	33
20.5.3 Copy Generation Management System (CGMS-A) Information	33
20.6 Scrambling	33
20.6.1 Key Derivation	34
20.6.1.1 ODD/EVEN Key Generation Formulae	34
20.6.1.2 Initial Key Derivation Procedure	35
20.6.1.3 Subsequent Key Derivation Procedure	36
20.6.2 Content Encryption	38
20.6.3 Transport Scrambling Control	38
20.7 Key Application for the Data Channel	39
20.8 CP Parameter Passing	39
21 Copy Protection Commands	40
21.1 PutCP_parameter	40
21.2 GetCP_Parameter	41
21.3 GetCACP_Request	41
21.4 PutCP_Synchro_Reply (CP Synchronization)	41
Annex A Serial Processing One MPEG-2 Data Stream Using Two Cards (Informative)	42
Annex B Electrical Characteristics (Normative)	43
Annex C CA Device Physical Size Limitations (Normative)	44
Annex D A Particular Implementation (Informative)	45
Annex E An Operational Scenario (Informative)	46
Annex F CEA-679-C Part B Extensibility (Normative)	47
Annex G Stored CA Display Messages (Normative)	48
Annex H Homing Support in Hosts (Normative)	49
Annex I Chaining support in Hosts (Informative)	50
1.1 Description	50
1.2 Procedure	50
1.4 Menus	50

I.5 Requirements.....	50
-----------------------	----

FIGURES

Figure 1 Timing Diagram.....	4
Figure 2 Flow Control, Next Byte at Total Guard Time (TGT).....	5
Figure 3 Flow Control, Next Byte After end_flow_control + X	6
Figure 4 Procedure for Initial CP key derivation.....	35
Figure 5 Procedure for Subsequent CP Key Derivation	37
Figure 6 Copy Protection Key Derivation Timeline	38
Figure 7 Implementation Example—Two NRSS Part A Modules in One Host Device	42
Figure 8 NRSS Part A Form Factor	44
Figure 9 Sample System Block Diagram	45

TABLES

Table 1 Procedure for NRSS Part A Activation/Deactivation	3
Table 2 Additional Contacts Used in NRSS Part A.....	3
Table 3 Timing Diagram.....	4
Table 4 Definition of CA Module Status Bit n.....	9
Table 5 Definition of Host Status Bit n.....	10
Table 6 CP_system_ID_bitmask Bit Numbers.....	11
Table 7 Purchase Type	13
Table 8 CA Status Byte Field	15
Table 9 CP_system_ID.....	31
Table 10 CCI Types and Sizes	33
Table 11 Baseline CCI.....	33
Table 12 Analog Protection System	33
Table 13 CGMS Values	33
Table 14 Key Derivation Sequence Parameter/Key Sizes.....	34
Table 15 CP_transport_scrambling_control_field descriptor values.....	39
Table 16 Data Channel Message Encryption Parameter Sizes.....	39
Table 17 Copy Protection datatype_ID coding.....	40
Table 18 CP_system_ID.....	40
Table 19 NRSS Part A DC Electrical Characteristics.....	43
Table 20 NRSS Part A AC Electrical Characteristics.....	43

PART B CONTENTS

1 Introduction and Scope	53
2 Definitions	54
3 Normative References	54
3.1 Normative Reference List	54
3.2 Normative Reference Acquisition	55
4 Design Philosophy	55
4.1 Layering	55
4.2 Physical Implementation	56
4.3 Coding of Data	56
4.4 Extensibility	56
4.5 Incorporation of Existing Standards	56
5 Description and Architecture	56
5.1 Overview	56
5.2 Transport Stream Interface	56
5.3 Command Interface	57
5.4 Physical Requirements	57
5.4.1 Introduction	57
5.4.2 Data and Command Logical Connections	58
5.4.3 Connection and Disconnection Behavior	59
5.4.4 Multiple Modules	59
5.5 Operational Example	60
6 Transport stream interface (TSI)	60
6.1 TSI - physical, link layers	60
6.2 TSI - transport layer	60
6.3 TSI - upper layers	60
7 Command Interface - Transport & session layers	61
7.1 Generic Transport Layer	61
7.1.1 Introduction	61
7.1.2 Transport Protocol Objects	62
7.1.3 Transport protocol	62
7.2 Session Layer	67
7.2.1 Introduction	67
7.2.2 Session Protocol Objects	67
7.2.3 Session Protocol	67
7.2.4 SPDU Structure	69
7.2.5 Transportation of SPDU	70
7.2.6 Session Headers Description	70
7.2.6.1 Open Session Request	70
7.2.6.2 Open Session Response	71
7.2.6.3 Create Session	72
7.2.6.4 Create Session Response	72
7.2.6.5 Close Session Request	72
7.2.6.6 Close Session Response	73
7.2.6.7 Session Number	73
7.2.7 Coding of the Session Tags	73

8 Command Interface—Application Layer.....	74
8.1 Introduction.....	74
8.2 Resources	74
8.2.1 Introduction	74
8.2.2 Resource Identifier	75
8.2.3 Applications and Resource Providers.....	75
8.3 Application Protocol Data Units.....	75
8.3.1 Introduction	75
8.3.2 Chaining of APDU data fields	76
8.3.3 Transportation of APDU within SPDU	76
8.4 System Management Resources	76
8.4.1 Resource Manager.....	76
8.4.1.1 Resource Manager Protocol.....	77
8.4.1.2 Profile Inquiry	77
8.4.1.3 Profile Reply	77
8.4.1.4 Profile Changed.....	78
8.4.2 Application Information	78
8.4.2.1 Application Info Inquiry	78
8.4.2.2 Application Info	79
8.4.2.3 Enter Menu.....	79
8.4.3 Conditional Access Support.....	79
8.4.3.1 CA_Info Inquiry	80
8.4.3.2 CA_Info	80
8.4.3.3 Selection of services to be descrambled	80
8.4.3.4 CA_PMT	80
8.4.3.5 CA_PMT Reply.....	82
8.5 Host Control and Information Resources	84
8.5.1 NRSS Host Control	84
8.5.1.1 Tune	84
8.5.1.2 Replace	84
8.5.1.3 Clear Replace	85
8.5.1.4 Ask Release	85
8.5.2 System Time.....	86
8.5.2.1 System Time Inquiry	86
8.5.2.2 System Time	86
8.6 Man-Machine Interface Resource	86
8.6.1 Introduction	86
8.6.2 MMI Objects.....	86
8.6.2.1 close_MMI.....	86
8.6.2.2 Display Control.....	87
8.6.2.3 Display Reply.....	88
8.6.3 MMI Keypad objects	88
8.6.3.1 Keypad Control	88
8.6.3.2 Keypress	89
8.6.3.3 Table of Key Codes	89
8.6.4 MMI File Transfer objects.....	89
8.6.4.1 Server Query.....	90
8.6.4.2 Server Reply	90
8.7 Communications Resources	91
8.7.1 Low-Speed Communication Resource Class	91
8.7.1.1 Introduction	91
8.7.1.2 Requirements	91
8.7.1.3 Objects Supporting the Low-Speed Communications Resource	91
8.7.1.4 Comms Cmd	92
8.7.1.5 Comms Reply	93
8.7.1.6 Comms Send	94
8.7.1.7 Comms Rcv	94
8.8 Homing Resources	94
8.8.1 Host Homing Resource Class	94

8.8.1.1 Introduction	94
8.8.1.2 Requirements	95
8.8.1.3 Objects Supporting the Homing Resource	95
8.8.1.4 Homing Resource Coding	95
8.8.1.5 Open Homing.....	95
8.8.1.6 Homing_Cancelled.....	96
8.8.1.7 open_homing_Reply	96
8.8.1.8 Homing Active	97
8.8.1.9 Homing Done.....	97
8.9 Copy Protection Framework.....	97
8.9.1 Copy Protection Resource Class	97
8.9.2 Service Provider [Informative].....	97
8.9.2.1 Back Channel [Informative].....	98
8.9.2.2 Service Provider Processing [Informative].....	98
8.9.2.2.1 Checking Validity of Host Information	98
8.9.2.2.2 Host Revocation	98
8.9.3 CA Module/Host Binding.....	98
8.9.4 Discrete Logarithm DH.....	98
8.9.4.1 Elliptic Curve DH	98
8.9.4.2 Needham-Schroeder Key Exchange Protocol	99
8.9.4.3 Computation of K_{cpss} and $K_{validated_cpss}$	99
8.9.5 Copy Control Information Definition.....	99
8.9.6 Scrambling	101
8.9.6.1 Key Derivation Sequence	101
8.9.6.2 ODD/EVEN Key Generation	102
8.9.6.3 Key Application for In-Band Channel	103
8.9.6.4 Content (In-Band Channel) Encryption	103
8.9.6.5 Data Channel Encryption.....	103
8.9.7 Copy Protection resource coding	104
8.9.8 APDU	104
8.9.8.1 CP_open.....	104
8.9.8.1.1 CP_open_req()	104
8.9.8.1.2 CP_open_cnf()	104
8.9.8.2 CP_data.....	105
8.9.8.2.1 CP_data_req()	105
8.9.8.2.2 CP_system_id	105
8.9.8.2.3 program_number	106
8.9.8.2.4 datatype_ID	106
8.9.8.2.5 CP_data_cnf()	106
8.9.8.3 CP_sync.....	107
8.9.8.3.1 CP_sync_req()	107
8.9.8.3.2 CP_sync_cnf().....	107
8.9.8.3.3 status_field.....	107
8.9.9 transport_scrambling_control Field	108
8.10 Chaining	108
8.10.1 Chaining Resource Class	108
8.10.2 Chaining resource coding	108
8.10.3 Objects Supporting the Chaining Resource	108
8.10.4 chaining_request	109
8.10.5 chaining_Status	109
8.10.5.1 chaining_status_ID	109
8.10.5.2 chaining_address_ID	109
8.10.6 chaining_control	109
8.10.7 buffer_write	110
8.10.8 buffer_write_reply.....	110
8.10.9 buffer_write_status_ID	110
8.10.10 buffer_Read_Request.....	110
8.10.11 buffer_read	111

8.11 Resource Identifiers and Application Object Tags	111
8.11.1 Resource Identifiers	111
8.11.1.1 Low-Speed Communications Resource Types	111
8.11.2 Application Object Tags.....	113
8.12 Host Control	115
8.12.1 OOB_TX_tune_req() & OOB_TX_tune_cnf()	115
8.12.2 OOB_RX_tune_req() & OOB_RX_tune_cnf().....	116
8.12.3 Extended Channel Support.....	117
8.12.4 new_flow_req() & new_flow_cnf()	118
8.12.5 delete_flow_req() & delete_flow_cnf()	120
8.12.6 index_flow_req() & index_flow_cnf()	120
8.13 Generic IPPV Support	121
8.13.1 program_req() & Program_cnf()	121
purchase_req() & purchase_cnf()	124
8.13.2 cancel_req() & Cancel_cnf()	126
8.13.3 history_req() & history_cnf()	127
Annex A PC Card-Based Physical Layer (Normative).....	129
A.1 General Description.....	129
A.1.1 PC Card interface	129
A.1.2 Descrambler.....	129
A.1.3 Filter/Extract	130
A.1.4 CPU.....	130
A.1.5 ROM/EPROM and RAM/NVRAM.....	130
A.1.6 Security Processor	130
A.2 Electrical Interface	130
A.2.1 Transport Stream Interface	130
A.2.2 Command Interface.....	130
A.2.2.1 Hardware Interface Description.....	130
A.2.2.1.1 Initialization	131
A.2.2.1.2 Host to Module Transfers.....	132
A.2.2.1.3 Module to Host Transfers.....	132
A.2.2.2 Hardware Support in the Module	132
A.3 Link Layer	133
A.3.1 Transport Stream Interface	133
A.3.2 Command Interface.....	133
A.4 Implementation-specific Transport sublayer over PC Card Interface	133
A.4.1 Transport Protocol Objects.....	133
A.4.1.1 Command TPDU	134
A.4.1.2 Response TPDU.....	134
A.4.1.3 Chaining of Command or Response TPDU Data Fields	135
A.4.1.4 Create Transport Connection (create_t_c)	136
A.4.1.5 Create Transport Connection Reply (c_t_c_reply)	136
A.4.1.6 Delete Transport Connection (delete_t_c).....	137
A.4.1.7 Delete Transport Connection Reply (d_t_c_reply)	137
A.4.1.8 Request Transport Connection (request_t_c).....	137
A.4.1.9 New Transport Connection (new_t_c)	138
A.4.1.10 Transport Connection Error (t_c_error).....	138
A.4.1.11 List of c_tpdus and associated r_tpdus	139
A.4.1.11.1 Send Data command	139
A.4.1.11.2 Receive Data Command	140
A.4.1.12 Rules for the Polling Function.....	140
A.4.1.13 List of Transport Tags.....	140
A.5 PC Card Subset to be Used by Conformant Hosts and Modules	141
A.5.1 Objectives	141
A.5.2 Introduction	141
A.5.3 Terminology.....	141

A.5.4 Physical Specification	141
A.5.4.1 Card dimensions	141
A.5.4.2 Connector.....	141
A.5.4.3 PC Card Guidance	141
A.5.4.4 Grounding/EMI Clips	141
A.5.4.5 Connector Reliability.....	141
A.5.4.6 Connector Durability	142
A.5.4.7 PC Card Environmental.....	142
A.5.5 Electrical Specification.....	142
A.5.5.1 Card Type Detection.....	142
A.5.5.2 Pin Assignments	142
A.5.5.3 16-bit PC Card features	143
A.5.5.4 Signal Description	143
A.5.5.5 Memory Function.....	146
A.5.5.6 Timing Functions.....	146
A.5.5.7 Electrical interface.....	146
A.5.5.8 Card Detect	146
A.5.5.9 Battery Voltage Detect	146
A.5.5.10 Power Supply Management	146
A.5.5.11 I/O Function.....	147
A.5.5.12 Function Configuration	147
A.5.5.13 Card Configuration.....	147
A.5.6 Metaformat Specification	147
Annex B MMI Display Requirements (Normative)	149
B.1 HTML Requirements	149
B.2 Optional HTPV Extensions.....	150
Annex C Additional Objects (Informative).....	152
C.1 Authentication	152
C.1.1 Authentication Request and Authentication Response.....	152
C.1.2 Authentication Resource Coding	152
C.2 Mating.....	152
C.2.1 Mating Request and Mating Response	153
C.2.2 Mating Resource Coding.....	153
C.3 Smart Card Reader Resource Class	153
C.3.1 Objects	154
C.3.1.1 Smart Card Cmd	154
C.3.1.2 Smart Card Reply	155
C.3.1.3 Smart Card Send	156
C.3.1.4 Smart Card Rcv.....	156
C.3.2 Smart Card Reader Resource Coding.....	157
C.3.2.1 Resource Type Coding.....	157
Annex D Homing Support in Hosts (Informative)	158
Annex E Chaining of CA Modules (Informative)	159
Annex F ISO-7816 Interface Extensibility (Normative)	160

Figures

Figure 1 Example of Single Module in Connection with Host	53
Figure 2 Layering on the command interface	56
Figure 3 Transport Stream Interface Layers	57
Figure 4 Command Interface Layers	57
Figure 5 Transport Stream Interface Chaining between Modules	60
Figure 6 State Transition Diagram for the Host side of the Transport Protocol	64
Figure 7 State Transition Diagram for the Module side of the Transport Protocol	65
Figure 8 Object Transfer Sequence Example for the Transport Protocol	66
Figure 9 Session to a Host-provided Resource	68
Figure 10 Session to a Module-provided Resource	69
Figure 11 SPDU Structure	70
Figure 12 APDU structure	75
Figure 13 Illustration of the Chaining Mechanism	76
Figure 14 Baseline CCI Field	100
Figure 15 Copy Protection Key Derivation	102
Figure 16 Low-Speed Communications Resource Type Structure	112
Figure 17 Resource Type Structure for Modems	112
Figure 18 Primitive Tag Coding	113
Figure 19 RF_TX_frequency_value Coding	116
Figure 20 RF_TX_coding_value Coding	116
Figure 21 RF_RX_frequency_value Value	117
Figure 22 RF_RX_coding_value Value	117
Figure 23 Status_register Field	126
Figure 24 Typical CA Module Architecture showing the Position of the PC Card Interface	129
Figure 25 Map of Hardware Interface Registers	130
Figure 26 Status Register	131
Figure 27 Command Register	131
Figure 28 Layout of Link Protocol Data Unit	133
Figure 29 c_tpdu structure	134
Figure 30 r_tpdu structure	134
Figure 31 SB_value coding	135
Figure 32 Illustration of the Chaining Mechanism with c_tpduTPDU	135
Figure 33 Illustration of the Chaining Mechanism with r_tpdu	136
Figure 34 create_t_c structure	136
Figure 35 c_t_c_Reply Structure	136
Figure 36 delete_t_c Structure	137
Figure 37 d_t_c_reply Structure	137
Figure 38 request_t_c Structure	137
Figure 39 new_t_c Structure	138
Figure 40 t_c_error Structure	138
Figure 41 Send Data command/response pair	139
Figure 42 Receive Data command/response pair	140
Figure 43 Timing Relationships for Transport Stream Interface Signals	145
Figure 44 Smart Card Reader Resource Type Structure	157

TABLES

Table 1 Length field used by all Protocol Data Units at Transport, Session & Application Layers ..	61
Table 2 Objects Expected to be Received in the States of a Transport Connection on the Host	64
Table 3 Objects Expected to be Received in the States of a Transport Connection on the Module ..	65
Table 4 SPDU Coding	70

Table 5 Open Session Request Coding	71
Table 6 Open Session Response Coding	71
Table 7 Open Session Status Values	71
Table 8 Create Session Coding	72
Table 9 Create Session Response Coding	72
Table 10 Close Session Request Coding	73
Table 11 Close Session Response Coding	73
Table 12 Close Session Status Values	73
Table 13 Session Number Coding	73
Table 14 Session Tag Values	74
Table 15 resource_identifier coding	75
Table 16 APDU coding	76
Table 17 Profile Inquiry Object Coding	77
Table 18 Profile Reply Object Coding	78
Table 19 Profile Changed Object Coding	78
Table 20 Application Info Inquiry Object Coding	78
Table 21 Application Info object coding	79
Table 22 Enter Menu Object Coding	79
Table 23 CA_info Inquiry Object Coding	80
Table 24 CA_info Object Coding	80
Table 25 CA_PMT Object Coding	81
Table 26 CA_PMT_list_management Values	82
Table 27 CA_PMT_cmd_ID Values	82
Table 28 CA_PMT Reply Object Coding	83
Table 29 CA_enable Values	83
Table 30 Tune Object Coding	84
Table 31 Replace Object Coding	85
Table 32 Clear Replace Object Coding	85
Table 33 Ask Release Object Coding	85
Table 34 System Time Inquiry Object Coding	86
Table 35 System Time Object Coding	86
Table 36 Close MMI Object Coding	87
Table 37 close_MMI_cmd_ID Values	87
Table 38 Display Control Object Coding	87
Table 39 MMI_mode Values	88
Table 40 Display Reply Object Coding	88
Table 41 Keypad Control Object Coding	88
Table 42 Keypress Object Coding	89
Table 43 Key Code Values	89
Table 44 Server Query Object Coding	90
Table 45 Server Reply Object Coding	90
Table 46 file_status Values	90
Table 47 comms_cmd Object Coding	92
Table 48 Connection Descriptor Object Coding	92
Table 49 comms_reply Object Coding	93
Table 50 Comms Send Object Coding	94
Table 51 Comms Rcv Object Coding	94
Table 52 Objects Used in Homing Resource	95
Table 53 Homing Resource Coding	95
Table 54 open_homing Object Coding	95
Table 55 Comms Reply Object Coding	96
Table 56 homing_cancelled_id Values	96
Table 57 open_homing_reply Object Coding	96
Table 58 system_time Value in system_time()	96
Table 59 homing_active Object Coding	97
Table 60 homing_done Object Coding	97

Table 61 CCI Types and Sizes	100
Table 62 Baseline CCI.....	100
Table 63 CGMS Values	100
Table 64 Analog Protection System.....	100
Table 65 Key Derivation Sequence Parameter/Key Sizes.....	101
Table 66 Data Channel Message Encryption Parameter Sizes.....	103
Table 67 Copy Protection resource coding.....	104
Table 68 Objects Used in Copy Protection Resource	104
Table 69 CP_open_req() Object Coding	104
Table 70 CP_open_cnf() Object Coding.....	105
Table 71 CP_system_id_bitmask Values.....	105
Table 72 CP_data_req() Object Coding.....	105
Table 73 CP_system_id	106
Table 74 program_number	106
Table 75 datatype_ID values	106
Table 76 CP_data_cnf() Object Coding.....	107
Table 77 CP_sync_req () Coding	107
Table 78 CP_sync_cnf () Syntax.....	107
Table 79 status_field Value	108
Table 80 CP_transport_scrambling_control_field Descriptor Values	108
Table 81 Chaining Resource Coding	108
Table 82 Objects Used in Chaining Resource	109
Table 83 chaining_request Descriptor Object Coding	109
Table 84 chaining_status Descriptor Object Coding	109
Table 85 chaining_status_id Values	109
Table 86 chaining_control Descriptor Object Coding	110
Table 87 buffer_write Descriptor Object Coding	110
Table 88 buffer_write_reply Descriptor Object Coding.....	110
Table 89 buffer_write_status_ID Values	110
Table 90 buffer_read_request Descriptor Object Coding	111
Table 91 buffer_read Descriptor Object Coding	111
Table 92 Resource Identifier Values	111
Table 93 Device Field Values	112
Table 94 dataproc Field Values	112
Table 95 Modem Type Field Values.....	113
Table 96 Application Object Tag Values.....	114
Table 97 Host Control Resource	115
Table 98 Host Control Objects.....	115
Table 99 OOB_TX_tune_req() and OOB_TX_tune_cnf()	115
Table 100 OOB_TX_tune_cnf() & OOB_TX_tune_cnf_tag	116
Table 101 status_field Values	116
Table 102 OOB_RX_tune_req() & OOB_RX_tune_req_tag Coding.....	116
Table 103 OOB_RS_tune_cnf() & OOB_TX_tune_cnf_tag Coding	117
Table 104 status_field Values	117
Table 105 extended_channel Identifier	118
Table 106 extended_channel Objects	118
Table 107 new_flow_req() & new_flow_cnf() Coding	118
Table 108 service_type Value	118
Table 109 new_flow_cnf() & new_flow_cnf_tag Coding	119
Table 110 status_field Values	119
Table 111 delete_flow_req() & delete_flow_req_tag Coding	120
Table 112 delete_flow_cnf() & delete_flow_cnf_tag Coding.....	120
Table 113 status_field Values	120
Table 114 index_flow_req() & index_flow_req_tag Coding	120
Table 115 index_flow_cnf() & index_flow_cnf_tag Coding.....	121
Table 116 status_field Values	121

Table 117 Generic IPPV Support Identifier	121
Table 118 Generic IPPV Support Objects	121
Table 119 status_field Values	123
Table 120 purchase_type Value.....	123
Table 121 c_tpdu coding	134
Table 122 r_tpdu coding	134
Table 123 Coding of bit8 of SB_value.....	135
Table 124 create_t_c coding	136
Table 125 c_t_c Reply Coding	136
Table 126 delete_t_c coding	137
Table 127 d_t_c_reply coding	137
Table 128 request_t_c coding.....	138
Table 129 new_t_c coding.....	138
Table 130 t_c_error coding	138
Table 131 Error Code Values	139
Table 132 Coding of Send Data c_tpdu	139
Table 133 Coding of Send Data r_tpdu	139
Table 134 Coding of Receive Data C_TPDU	140
Table 135 Coding of Receive Data r_tpdu	140
Table 136 Transport Tag Values	141
Table 137 Pin assignments for this PC CARD variant	143
Table 138 Timing Relationship Limits.....	146
Table 139 NRSS Subset of HTML 3.2	149
Table 140 HTVP Functions	151

.....

(This page intentionally left blank.)

NATIONAL RENEWABLE SECURITY STANDARD (NRSS) PART A

1 Introduction

CEA-679-C Part A describes the physical and electrical parameters of the interface for conditional access (CA) devices for consumer electronics (CE). The purpose of this standard is to specify the interface to a CA device that performs the signal access and, optionally, descrambling and key generation. It is possible to imagine that several CA devices could be interfaced to the same CE device, simultaneously (see Annex A (informative)). This interface definition allows the entire CA system to be removable and hence replaceable. Very few constraints, if any, are placed on the CA system architecture.

2 Scope and Field of Application

CEA-679-C describes the interface from a physical, electrical, and functional view and the physical form factor for the CA device. There may be other specifications that apply generally to consumer household electronic devices, for example, environmental protection and radiation requirements. CEA-679-C is not inclusive of all specifications that may apply to the CA device.

3 Normative References

The following references contain provisions, which, through reference in this text, constitute normative provisions of this standard. At the time of publication, the edition indicated was valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standard indicated.

3.1 Normative Reference List

CEA-796-A, NRSS Copy Protection Systems, October 2005

ISO 7810 - Identification cards. Physical characteristics (Second Edition, 1995)

ISO 7816 - Identification cards. Integrated circuit(s) cards with contacts, Part 1: Physical characteristics (First Edition, 1987).

ISO 7816 - Identification cards. Integrated circuit(s) cards with contacts, Part 2: Dimensions and location of the contacts (Revision of first edition, 1988).

ISO 7816 - Identification cards. Integrated circuit(s) cards with contacts, Part 3: Electronic signals and transmission protocols (First edition, 1989 with Amendment 1, 1992)

ISO/IEC 13818-1(Draft International Standard, 1994) Information Technology - Generic Coding of Moving Pictures and Associated Audio

ATSC A/65—Program and System Information Protocol for Terrestrial Broadcast and Cable, 12/23/97

NIST FIPS PUB 46-3, Data Encryption Standard, October 25, 1999 (withdrawn May 19, 2005)

NIST FIPS PUB 81, DES Modes of Operation, December 2, 1980 (withdrawn May 19, 2005)

3.2 Normative Reference Acquisition

ATSC Standards:

- Advanced Television Systems Committee (ATSC), 1750 K Street, N.W., Suite 800, Washington, DC 20006, Phone 202-828-3130; Fax 202-828-3131; Internet <http://www.atsc.org>

CEA Standards:

- Global Engineering Documents, World Headquarters, 15 Inverness Way East, Englewood, CO USA 80112-5776; Phone 800-854-7179; Fax 303-397-2740; Internet <http://global.ihs.com>; Email global@ihs.com

IEC Standards:

- Global Engineering Documents, World Headquarters, 15 Inverness Way East, Englewood, CO USA 80112-5776; Phone 800-854-7179; Fax 303-397-2740; Internet <http://global.ihs.com> ; Email global@ihs.com
- IEC Central Office, 3, rue de Varembe, PO Box 131, CH-1211 Geneva 20, Switzerland; Phone +41 22 919 02 11; Fax +41 22 919 03 00; Internet <http://www.iec.ch> ; Email pubinfor@iec.ch

ISO Standards:

- Global Engineering Documents, World Headquarters, 15 Inverness Way East, Englewood, CO USA 80112-5776; Phone 800-854-7179; Fax 303-397-2740; Internet <http://global.ihs.com> ; Email global@ihs.com
- ISO Central Secretariat Office, 1, rue de Varembe, Case postale 56, CH-1211 Geneve 20, Switzerland; Phone +41 22 749 01 11; Fax +41 22 733 34 30; Internet <http://www.iso.ch> ; Email central@iso.ch

NIST FIPS PUBs

- National Institute of Standards and Technology (NIST), c/o National Technical Information Service (NTIS), 5285 Port Royal Road, Springfield, VA 22161; Phone 703 605 6000; Email orders@ntis.gov ; <http://www.csrc.nist.gov/publications/fips/index.html>

4 Definitions

Also see ISO 7816.

A **CA module (CAM)** is the conditional access device, such as, an ISO 7816 smart card or a conditional access card. This is the device that provides authorization and decryption functions for the consumer's system.

A **CE device** is a consumer electronics device, such as a television receiver, a set-top converter box, or a plug-in module that performs a signal conditioning function and houses the conditional access interface.

A **PID** is a packet identification number. This is used to identify a single data stream within a packetized multiplexed data stream. A television service may consist of several data streams, including but not limited to, one PID number for video, and one or several PID numbers for audio.

A **reportback channel** is a means of communication to a head-end or uplink system. Examples of such communication channels are telephone connections, reverse cable modem channels, and wireless communications.

NVM is non-volatile memory, that is, memory that is retained when operating power is removed, such as EEPROM, ferrite core memory, a disk drive, or battery-backed-up RAM.

Number Base: Unless otherwise specified, numbers in this document are decimal base. However, numbers that begin with 0x are hexadecimal base numbers.

5 General Electrical and Physical Characteristics

The CA module shall conform to the physical and electrical characteristics (Annex B (normative)) described in ISO 7816 with exceptions and additions noted in Section 6. Additionally, the CA module shall be capable of being inserted in and removed from a powered CE device without damage to either the CA module or CE device.

The conversation between the CA module and CE device shall start (end) with the activation (deactivation) of electrical contacts by the CE device. The consecutive operations needed for these two processes are shown in Table 1.

Activation	Deactivation
(1) RST state L	(1) RST state L
(2) VCC powered	(2) CLK state L
(3) I/O reception mode (in CA)	(3) VPP inactive (high speed clock)
(4) VPP inactive (high speed clock)	(4) DIN inactive or low
(5) CLK provided with a stable clock	(5) I/O state A
	(6) VCC inactive

Table 1 Procedure for NRSS Part A Activation/Deactivation**6 Exceptions and Additional Physical Characteristics**

The CA module shall employ the card height (nominally 53.98 mm), width (nominally 85.60 mm), and thickness (nominally 0.76 mm) dimensions of ID-1 type cards defined in ISO 7810, with the following exceptions. Unless otherwise specified, all measurements have a tolerance of 0.05 mm.

- a) Card Width: The card width may be any value between 85.60 mm and 111.00 mm.
- b) Card Thickness: The portion of the card that is inserted into the connector shall have a thickness of 0.76 mm, for a distance of at least 37.00 mm from that edge of the card which is inserted into the connector. At any point within 5.0 mm from any of the edges of the card, the card shall have a thickness of 0.76 mm. The remainder of the card may have a thickness not to exceed 5.00 mm.
- c) Flexibility: The flexibility requirements of ISO 7816 are not required in the CA module.

The intent of this set of physical requirements is to allow the use of a wide range of existing ISO 7816 connectors, while simultaneously accommodating a wide range of internal architectures for the CA module. The CA module may contain batteries and many ICs. The CA module form factor range also includes the flexible IC card which is superficially indistinguishable from the ISO 7816 card. Physical limitations of the CA module are depicted in Annex C (normative).

7 Exceptions and Additional Electrical Characteristics

This CA module uses the same 8 contact physical connector as ISO 7816, but three contacts are used differently. These contacts are C4, C6, and C8, and their uses are identified in Table 2.

contact	Use	Previous assignment
C4	High-speed data in	Reserved for future use
C6	High-speed clock	VPP (programming voltage)
C8	High-speed data out	Reserved for future use

Table 2 Additional Contacts Used in NRSS Part A

The contacts C1, C2, C3, C5, C7 remain unchanged. These are the complete set of ISO 7816 contacts that are used for most modern ISO 7816 integrated circuit cards containing microprocessors. An example architecture for a CA module is given in Annex D (informative). This architecture is flexible for future improvements; stronger scrambling algorithms, longer keys, and different formats for EMM and ECM filters can be used.

The timing diagram (see Figure 1) describes the relationship between the three high speed signals. Table 3 summarizes the timing requirements of this interface.

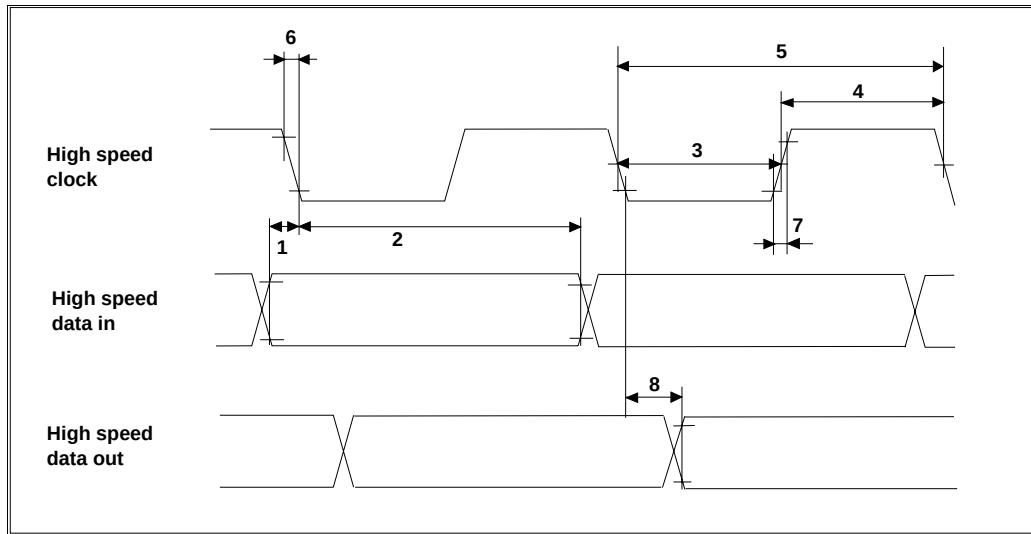


Figure 1 Timing Diagram

#	Symbol	Parameter	Conditions	Value		Unit
				Min.	Max.	
1	T_{sDIN}	High-speed Data In Set-up Time before DCLK↓		4.0		ns
2	T_{hDIN}	High-speed Data In Hold Time after DCLK↓		1.5		ns
3	T_{wLDCLK}	High-speed Digital Clock Low Period		9.0	11.0	ns
4	T_{whDCLK}	High-speed Digital Clock High Period		9.0	11.0	ns
5	T_{DCLK}	High-speed Digital Clock Period		20.0		ns
6	T_{rDCLK}	High-speed Digital Clock Rise Time			3.0	ns
7	T_{fDCLK}	High-speed Digital Clock Fall Time			3.0	ns
8	T_{dDOUT}	High-speed Data Out Valid Time after DCLK ↓	$C_{load}=50pF$	2.5	15.0	ns

Table 3 Timing Diagram

7.1 High-speed Data-in

This is the high-speed serial data in signal, and is a one bit input per each high-speed clock cycle. The data-in serial stream is an MPEG-2 system level transport packetized data multiplex. Between packets, zero stuffing is required.

7.2 High-speed Clock

This is the clock that determines the bit rate of both the high-speed input and high-speed output. Continuous operation is required.

7.3 High-speed Data-out

This is the high-speed data out signal that produces one bit of data output per each high-speed clock cycle. The data-out serial stream is an MPEG-2 system level transport packetized data multiplex. The card has a fixed bit delay between input and output.

7.4 Functional Operations on High-speed Data

On startup, the high-speed data out is a delayed copy of high-speed data-in with a fixed bit delay.

7.5 Communications Protocol and Flow Control

Only asynchronous half-duplex character transmission protocol, T=0 (see ISO 7816-3), is supported and required. (Asynchronous half-duplex block transmission protocol can optionally be implemented if the application needs it.) If an NRSS-A module will not be able to accept communications, it can assert the data line as indicated in Section 7.5.1 and Section 7.5.2 to provide flow control.

7.5.1 CA Module Implementation of Flow Control

The CA module may signal that it is busy in the following ways:

- Assert the data line (state A) beginning at 11.5 ± 0.2 ETUs after the start of the last character (during the time between characters). This will cause the host to delay the next character. The assertion shall be for more than 1.0 ETU and last until at least 12.5 ETU's from the current start bit.
- Assert the data line (state A) beginning more than 12.5 ETUs after the start of the last character (during the time between characters). This will cause the host to delay the next character. The assertion shall be for more than 12.0 ETU's.

The maximum time that the flow control shall be asserted is the duration of the work waiting time (see Section 8.1 of ISO 7816-3).

Flow control shall not be asserted on the last character before a direction change of the data.

Flow control shall not occur if the CA module has indicated a parity error condition on the previous word.

7.5.2 Host Implementation of Flow Control

The host shall implement Section 6.3 of ISO 7816-3 (error detection and repetition).

The host shall test the data line before any transmission to verify that the CA module is not exercising flow control. If the data line is asserted (state A), the host shall wait until at least 1.0 ETUs after the line returns to the high impedance state (Z) to start its transmission.

If the data line does not return to state Z during a full work waiting time, the host should reset the CA module.

The next byte after flow control when transmitting will not be sent before: max. (flow_control_end + X, TGT), where $X = 1 \pm 0.02$ ETU (see Figure 2 and Figure 3).

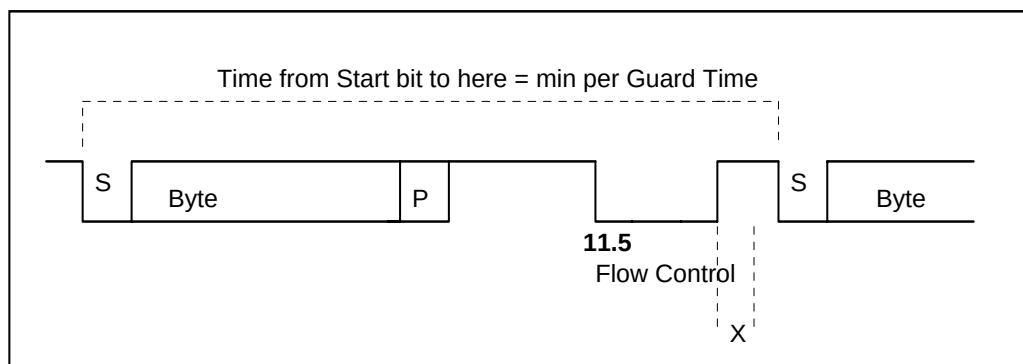


Figure 2 Flow Control, Next Byte at Total Guard Time (TGT)

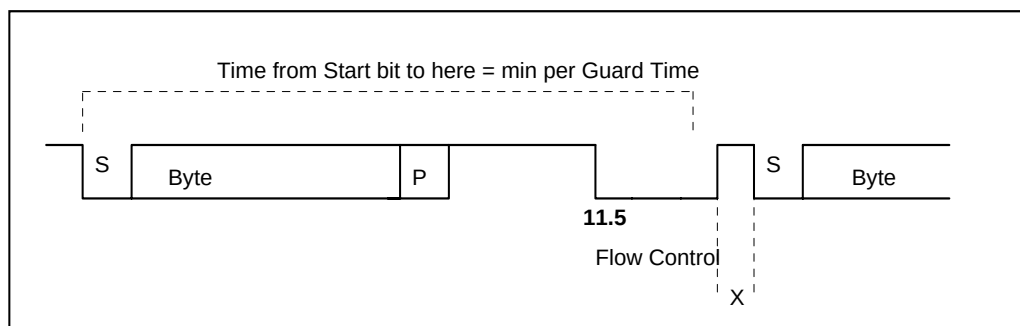


Figure 3 Flow Control, Next Byte After end_flow_control + X

8 Commands

Commands are sent between the CE device and the CA module via the ISO 7816 serial I/O line. Using the commands presented in Sections 9-21, complete conditional access applications can be created. An example of how this can be accomplished is included in Annex E (informative).

Only asynchronous half-duplex character transmission protocol, T=0 (see ISO 7816-3), is supported and required (Asynchronous half-duplex block transmission protocol can optionally be implemented if the application needs it). All commands are originated by the CE device. Any information that is sent from the CA module to the CE device must be requested by the CE device. Each command consists of a 5-byte command preamble and a message. The preamble is always sent from the CE device to the CA module. Depending on the contents of the preamble, the message either is sent from the CE device and is read by the CA module, or it is sent from the CA module and is read by the CE device.

For each command, the explanation indicates whether the command is required to be implemented in the CA module or not, and the default values of the data message, if any.

For all commands that are listed as “required”, the CA module must support the intended functionality in order to be considered an NRSS-A-compliant CA module.

The default values are to be used when the command must be supported to prevent a communication failure but the related functionality is not supported by the CA module.

The preamble of each command is 5 bytes consisting of an instruction class byte, CLA, followed by the instruction code byte, INS, followed by three parameter bytes, P1, P2, and P3. The parameter P3 is used to indicate the length of the following message. The CA module knows whether to receive a message or send a message as a function of the preamble bytes. Commands that begin with a “Get” will result in the CA module sending data to the CE device, while those that begin with a “Put” will send data from the CE device to the CA module.

The commands are of various types and purposes. The commands in this document are grouped together in common functions as follows:

- Section 9: General commands and answer-to-reset.
- Section 10: Pay-per-view - commands used to purchase current and future pay-per-view or subscription events.
- Section 11: Purchase history - commands to access and save pay-per-view history information (when supported by both devices).
- Section 12: Chaining – transferring information from an ‘old’ card to a ‘new’ card to continue a subscription when hardware is replaced.
- Section 13: Reportback – communications to send information from the subscriber system to a head-end system. This supports billing and other functions.
- Section 14: Locks and limits – commands to ‘lock’ the subscriber system and prevent unintended purchases/viewing.

- Section 15: Service changes – commands required to ‘tune’ channels and change services.
 Section 16: Display and menu – commands to allow user control and interface with the CA system.
 Section 17: Mail – communications from the head-end to subscribers via addressed messages.
 Section 18: Intercard messaging – commands to allow multiple NRSS CA modules to exchange information.
 Section 19: NVM – commands to allow NRSS CA modules to make use of CE host device non-volatile memory.
 Section 20: Copy protection – commands to support copy protection systems identified in CEA-796-A.

In the commands supporting purchase functionality, the term “slot” refers to a specific location in the CA module and/or CE device NVM dedicated to storage of purchase information. One slot contains the data related to one purchase action. The standard supports devices that store purchase information in either the CA module or in the CE device.

A number of commands pass on a 12-byte parameter in the data message that uniquely identifies the event in question and associates it with a particular data stream. ATSC has previously defined these bytes as indicated to the left of the “|”, and for POD uses, these bytes are defined as indicated to the right of the “|” as follows:

Bytes 0-1	Transport ID Network ID
Bytes 2-3	Major channel Transport ID
Bytes 4-5	Minor channel Service ID
Bytes 6-7	Event ID
Bytes 8-11	Start Time

In this document, this 96-bit descriptor will be called by the single name “Unique Event Descriptor” or UED.

9 General Commands

9.1 Answer-to-Reset

The answer-to-reset of the CA module shall conform with the asynchronous answer-to-reset as specified in ISO 7816-3, Section 6.1.

Coding of serial communications shall follow the ‘inverse’ convention described in ISO7813-3 Section 6.4.1.

9.2 Get CA Module Configuration

This command requests status and configuration data about the CA module itself.

Required: Yes

Default data message: None, command must be supported.

The data message returned is defined as follows:

Byte 0	CA Module Status Bit 0 =1 CA module is authorized Bit 1 =1 CA module has received chaining data Bit 2 =1 CA module has been disabled Bits 3-7 reserved
Bytes 1-8	Current CA module ID If the most significant bit of the CA module ID is 1, then the CA module is CEA-679-C Part B compatible. (See Annex F (Normative)).
Bytes 9-16	Replacement ID for chaining. This ID shall be stored by the CE device and checked against future device IDs. (See Section 12 and Annex I (Informative) for more details.)

- Byte 17 Functionality profile bitmap of CA system. The Host will decide which features it will support.
- Bit 0 =0 CA Device does not store IPPV history
 =1 CA Device stores IPPV history
 - Bit 1 =0 Host will provide ECM, FECM and EMM PID's.
 =1 CA module will process PSI stream. Host will provide primary service/Event ID.
 - Bit 2 =0 CA module does not provide any CA menus
 =1 CA module will manage CA related menus
 - Bit 3 =0 CA module does not support ANY PINs.
 =1 CA module will process and check PIN(s)
 (See BYTE 56 for number of PINs CA device supports.)
- Byte 18 Number of IPPV records in CA module
- Byte 19 Size of title field in CA Module for the IPPV history
- Byte 20 Time Zone of card – (binary) number of half-hours added to UTC for current time zone. Values from 24-47 subtract one day for correct date. Bit 7 =1 indicates daylight savings time is applicable in this region. (The host may choose to use PSIP time instead of this CA information.)
- Bytes 21-22 CA system ID (This shall match ISO-13818 CA descriptor field)
- Bytes 23-54 32-byte CA system name. This is an ASCII string with the CA system name for display by host of Card name. (NULL (0x0) terminated.)
- Byte 55 Number of PIDs the CA module can concurrently descramble.
- Byte 56 Number of users (PINs) that can be stored by CA module
- Byte 57 Maximum data block size for chaining, reportback, CA module poll messages, etc.
- Byte 58 Index of Selected Copy Protection System (See CEA-796-A) (0 = none)
- Bytes 59-65 *reserved for future use*

CLA	INS	P1	P2	P3
0x80	0x10	0	0	66

Message
CA Module status and configuration information

9.3 Get CA Module Status

This command returns the value Status from the CA module. The Status word is a 4 byte (32 bit) unsigned integer that is read by the CE device periodically to determine what type of operations are to be performed. If all of the bits of Status are 0, then there are no requests from the CA module. The bits of Status are numbered from 0 to 31, from rightmost to leftmost. The operation 2^n is defined to mean 2 raised to the power n. Bit n of Status is clear if $(\text{Status AND } 2^n)$ is zero. Bit n of Status is set if $(\text{Status AND } 2^n)$ is not zero. Table 4 defines the meaning of each bit in the Status word and the next operation that the CE device is to perform if bit n is set.

Status Bit n	I/O operation requested (listed by CEA-679-C Part A Section)	If Bit n is set then CA module:
0	16.6	has pending menu request
1	16.3	has pending alpha box request
2	16.1	has a pending display message request
3	13.1	has a pending open reportback request
4	13.3	has a pending reportback message to send
5	13.5	requests to close reportback channel
6	17.1	has an intercard message to send
7	18.1	is requesting the number of bytes of host NVM available
8	18.2	is requesting an allocation of host NVM
9	18.4	is requesting to read from NVM
10	18.4	is requesting to write to NVM
11	18.4	is requesting to delete an allocation of host NVM
12	9.4	is requesting to exchange identities
13	10.2	has purchase offer data available
14	None	CA module has completed homing, CE device may now turn off
15	17.1	has unread broadcaster mail for the user
16	15.6	is requesting that the CE device tune to a particular service
17	None	has received an EMM that cleared the password and the system is now unlocked
18	15.4	has a change in Program Status
19	11.4	has an update to make to the purchase history
20	9.2	configuration has been changed (Used to change replacement ID's, copy protection, or authorization bits)
21	None	Reserved
22	20.3.3	requesting a CP_parameter from the host
23	20.3.2	Has a CP_parameter available to be read
24	20.4.1	CA module is ready to change CP scrambling keys, and is waiting for the host
25—31		Reserved

Table 4 Definition of CA Module Status Bit n

CLA	INS	P1	P2	P3
0x80	0x12	00	00	4

Message
Status

9.4 Put CE device ID

This command writes an 8 byte identification code of the CE device to the CA module. If the most significant bit of the CE device ID is set to 1, then this command also instructs the CA module to switch to CEA-679-C Part B protocol, if so equipped. (See Annex F (normative).)

CLA	INS	P1	P2	P3
0x80	0x14	0	0	8

Message
CE device ID

Required: Yes

9.5 Get Suggested ID

This command gets an 8-byte code from the CA module. This code is a suggested ID for the CE device.

CLA	INS	P1	P2	P3
0x80	0x16	0	0	8

Message
Suggested ID

Required: No

9.6 Put Host Configuration/Status

Bytes 0-3 contain the host time. Calculated as (seconds since 12AM January 6, 1980 UTC) mod³².

Bytes 4-9 send the host status/configuration to the card. See Table 5.

Status Bit n	If Bit n is set then host:
32-63	Supports a given copy protection system (See Table 6 and CEA-796-A)
64-68	Host supports this number of rescrambled bitstreams for copy protection
69	has a reportback channel (modem or otherwise)
70	host device is now available for homing for this CA module
71-76	Reserved
77-80	for proprietary use

Table 5 Definition of Host Status Bit n

Bit numbers shall be assigned as indicated in Table 6.

CP_system_ID_bitmask	Bit Number =1
System 1	32
System 2	33
System 3	34
System 4	35
System 5	36
6-31 Reserved	Other values

Table 6 CP_system_ID_bitmask Bit Numbers

CLA	INS	P1	P2	P3
0x80	0x18	0	0	10

Message
Time, 6 bytes of status

9.7 Get Requested Homing Time

Prior to shutting down, the CE device sends this command to allow the CA module to request that the host turn on at a particular time for homing. 4 bytes of a time value are returned. The time calculation is the same as in Section 9.6. A return value of 0xFFFFFFFF means the host device should not shut off, but should start homing now. A return value of 0x00000000 means the host does not need to set a wake up time. See Annex H (Normative).

Required: Yes

Default: 0x00000000

CLA	INS	P1	P2	P3
0x80	0x1A	0	0	4

Message
Time

10 Purchase and Cancel IPPV Commands

10.1 Put Purchase Action

This command is used to request purchase information from the CA module and to provide the details for locating the program's purchase information. The command is also used to query the CA module if a future event is purchasable by asking the CA module to check the purchase status of an event using the associated FECM. The command Get Purchase Status returns the result of the query/purchase and must be called immediately after this command.

Required: yes

Default data message: None

The value of P1 indicates the type of purchase:

- = 0 from an ECM on the current channel
- = 1 from an FECM that is not included with this command
- = 2 from an FECM included with this command

The data message contains the information required to locate the purchase information.

If P1= 0, then the data is 2 bytes indicating the PID of the ECM that generated the purchase, as follows:

Byte 0, Bits 0-4 Most significant bits of PID number
 Byte 1 Least significant bits of PID number

If P1 = 1, then the data is 12 bytes identifying the event whose FECM is to be acquired:

Bytes 0-11 Unique Event Descriptor

If P1=2, the data is 12 bytes identifying the event, and N-12 bytes of an FECM for this event.

Bytes 0-11 Unique Event Descriptor

Bytes 12- The FECM for this event (parsed from the EIT)

If P1=2, then when the EIT contains purchase information for the desired program, it will be in the form of an ATSC_CA_descriptor that is associated with a given event. If an ATSC_CA_descriptor is available with a CA_ID that matches the CA_ID for the CA module, then the private data from this descriptor shall be included with the request for purchase information.

CLA	INS	P1	P2	P3
0x80	0x20	Purchase source	0	2, 12, or N

Message
see above

10.2 Get Number of Purchase Offers

After a Purchase Inquiry (Section 10.1) has been processed and a response is available, bit 13 of Get Status (Section 9.3) is set. This command responds to that flag and determines the number of available purchases for the requested event.

Required: Yes

Default message: All fields equal zero.

Byte 0 Number of purchase offers

Bytes 1-2 PID

Byte 3 Program Information Location. Values for byte three shall be defined as:

- 0- Title is available separately (see byte 16)
- 1- Current event on current service
- 2- Next event on current service
- 3- Event defined in bytes 4-15

Bytes 4-15 UED

Byte 16 Title Length to be read with command 10.4 (if indicated by byte 3, 0 otherwise)

CLA	INS	P1	P2	P3
0x80	0x22	0	0	17

Message
Number of purchase offers available

10.3 Get Purchase Offer Data

After the CA module has signaled the host (bit 13, Section 9.3) that it has purchase information available, the host will read back this information from the CA module. This command retrieves the information for one purchase offer generated in response to the last Put Purchase Inquiry (Section 10.1). A host will send a number of Get Purchase Offer Data messages based on the value read in Get Number of Purchase Offers (Section 10.2).

P1 contains N, the number of the purchase offer requested. (1 to K where K was the number returned in Section 10.2.)

Required: No

Default message: None

The purchase information is in the following format:

Byte 0	Type of purchase (see Table 7)
Bytes 1-2	Purchase price
Bytes 3-6	Expiration Date (MMDDYYYY, each nibble a BCD digit)
Bytes 7-8	Individual Free Preview Time
Byte 9	Rating limit
Byte 10	Purchase type length (0 if not used)

Value	Meaning
0x00	"Viewing Only"
0x01	"Tape Once"
0x02	"Tape Unlimited"
0xFF	Custom-byte 10 contains length of text string. See Section 10.4

Table 7 Purchase Type

CLA	INS	P1	P2	P3
0x80	0x24	N	0	30

Message
Purchase information

10.4 Get Purchase Offer Title

If the CA module returns a non-zero title length L in Section 10.2 or Section 10.3, the host may read the information immediately after the purchase offer data. When P1=00 to 0xFE, the purchase type for that number is requested. When P1=FF, then the program title is sent.

Required: No

Default message: None

CLA	INS	P1	P2	P3
0x80	0x26	0	0	L=Title Length

Message
Title

10.5 Put Purchase Action

This command is sent after the user has chosen a purchase offer from those presented using the previous commands. N, the first data byte, indicates which purchase offer has been selected. If P3=1, then the requested purchase is for the current channel and N is the only data. If P3=13, then the requested purchase is for a future event, and the Unique Event ID is sent

Required: Yes

Default message: None

CLA	INS	P1	P2	P3
0x80	0x28	0	0	1 or 13

Message
N, Unique Event Descriptor (optional)

10.6 Put Purchase Cancellation

This command may be used to cancel a previously purchased but not viewed program. There are two methods that can be used to cancel: (1) using the FECM data from the guide, or (2) from the purchase history using the slot number and program id.

Required: Yes

Default message: None

P1 indicates which method will be used for cancellation:

- = 0 Cancel using FECM
- = 1 Cancel using slot number and program event descriptor

If P1=1, then the slot number is in P2, otherwise P2 is meaningless.

The message data is the 12-byte Unique Event Descriptor used by the CA module to acquire the FECM. Alternatively, it is used for checking that the Unique Event Descriptor (UED) in the CE device matches the UED in the CA module.

This command should be followed by an immediate call to Get Purchase Status to see if it was successful.

CLA	INS	P1	P2	P3
0x80	0x2A	cancel_method	Slot Number	12

Message
Information for cancellation

10.7 Get Purchase Status

This command is used after a Section 10.5 (Put Purchase Action) or Section 10.6 (PutPurchase Cancellation) command to retrieve the CA module's response to the request or query.

Required: Only if number of IPPV records >0 (see Section 9.2)

Default message: None

The data message format is as follows:

- Byte 0 CA Status Byte (see Table 8)
- Byte 1 Slot used by CA module for previous purchase action/cancellation
- Byte 2 Program Status(see below)
- Byte 3-4 Price
- Byte 5 Rating Limit
- Byte 6-9 Expiration date: MMDDYYYY, each nibble a BCD digit
- Byte 10-11 Free Individual Preview Time remaining
- Byte 12 Special Reason Length (=0 unless Program Status bit 4=1)

CLA	INS	P1	P2	P3
0x80	0x2C	0	0	1913

Message
Purchase request/query response from CA module

10.7.1 CA Status Byte

If the Program Authorized or Viewing Purchased bit is set, the program is already authorized and can not be purchased again. See Table 8 for bit designations.

1) Bit	2) Bit 5 = 1	3) Bit 5 = 0
0	=1 Viewed	<i>reserved for future use</i>
1	=1 Cancelled	<i>reserved for future use</i>
2	=1 Reported	<i>reserved for future use</i>
3	=1 Free Individual Preview Used	<i>reserved for future use</i>
4	=1 Program Authorized	<i>reserved for future use</i>
5	=1 Viewing Purchased	=0
6		=1 Unauthorized CA module
7		=1 Purchase Available

Table 8 CA Status Byte Field

10.7.2 Program Status Byte

Each bit indicates a reason why viewing/taping is not permitted. (See Annex G (normative) for prestored OSD messages.)

- Bit 0 = 1 Blackout active. Use prestored OSD # 09.
- Bit 1 = 1 Credit limit exceeded. Use prestored OSD # 13.
- Bit 2 = 1 All slots full. Use prestored OSD # 12. .
- Bit 3 = 1 CA module is not paired to this CE device and Pairing is required. Use prestored OSD # 14.
- Bit 4 = 1 Spending Limit exceeded. PIN may be entered to override
- Bit 5 = 1 Rating exceeded. PIN May be entered to override
- Bit 6 = 1 Return Path required but not available. Use prestored OSD # 15.
- Bit 7 = 1 Special Reason. CA device will respond with a text message to override the default message or for another reason. (Message is sent in a separate command (Section 10.8).

10.7.3 Status Byte Interpretation

If the program was previously purchased or already authorized, the Authorized bit in CA status will be set.

If the spending-limit- or rating-limit-exceeded bit is set, then PIN entry to unlock or temporarily unlock the CA module is required for the requested action.

The CA module shall set only a single bit. The action to be taken should correspond to the failure that occurs. For instance, if there is a blackout, no user action will solve the problem whereas if the Credit limit is exceeded, the subscriber could be shown a message to call the Customer Service Center. The relevant prestored OSD is listed above.

10.8 Get Special Reason Message

If the CA module returned a non zero length for a purchase failure the reason will be read here. This is to allow adding additional reasons for a failure in the future.

Required: No

Default message: None

The value L in P3 is the length of the failure message. This length is defined by Byte 12 of the data transmitted in the Section 10.7.

CLA	INS	P1	P2	P3
0x80	0x2E	0	0	L

Message
Failure Reason Text String

11 Purchase History Commands

11.1 Get Purchase Record Length

Reads the length of a particular purchase record from the CA module. P1 specifies the slot requested.

Required: Yes

Default message: If a module does not support Purchase History, a length of 0 is returned.

CLA	INS	P1	P2	P3
0x80	0x30	N	0	1

Message
Length of record for slot N

11.2 Get Purchase Record

If the CA module is capable of storing the entire history then the host will use this command after any purchase. In addition, if the system previously stored the history in the host and is now transferring the data to a new module capable of storing history in the CA module (using chaining capability) the CA module will set a status bit and as a result the host will update the CA module's history.

Required: for Host devices that store IPPV history

Default message: None

P1 defines whether the data being stored in the purchase history record contains program data from the SI stream as well as CA data (P1=1), or only CA data (P1=0). P2 indicates the slot number for storage of the record. The value of the data length L in P3 depends on the value of P1:

If P1=0, P3=28

If P1=1, P3=35+{length of title text string}

The data message itself depends on the value of P1:

If P1=0:

Byte 0 CA Status
 Bytes 1-12 Unique Event Descriptor
 Bytes 13-14 Price
 Bytes 15-18 Expiration Date
 Bytes 19-23 reserved for future extensions

If P1=1:

Bytes 1-11 Unique Event Descriptor
 Byte 12 CA Status
 Byte 13 Purchase Type (See Table 7 in Section 10.3 for a list of values.)
 Byte 14-17 Purchase Time
 Bytes 18-21 Event time
 Bytes 22-25 Cancel Time
 Bytes 26-27 Price
 Bytes 28 Rating
 Bytes 29-32 Expiration Date
 Bytes 33-35 Reserved
 Bytes 36+ Title Text and Type description (when type=0xFF)

CLA	INS	P1	P2	P3
0x80	0x32	Types of data stored	slot number	L

Message
Purchase record data

11.3 Put Purchase Data

This command transfers the Program Information part of the purchase information for an IPPV program to the CA module so that the CA module may store the purchase history if the CA module has this capability. (See Get CA Module Configuration).

Required: Only if the number of IPPV's stored is >0 (see Section 9.2)

Default message: None

P2 contains the same slot number as the CA purchase record, so that the additional program data stored with this command remains mapped to the CA purchase record.

The data message contains the following information. Its length, P3, is 18 + {the length of the title text}.

Bytes 0-11 Unique Event Descriptor
 Bytes 12-16 reserved for future extensions
 Bytes 17- Title Text

CLA	INS	P1	P2	P3
0x80	0x34	0	Slot number	L

Message
Program data for purchased program

11.4 Get Updated Purchase Slot Info

This command returns the slot number where purchase data has been changed.

Required: Only if the number of IPPV's stored is >0 (see Section 9.2)

Default message: None

CLA	INS	P1	P2	P3
0x80	0x36	0	0	1

Message
Slot number

12 Chaining Commands

The CE host device shall store the CA module ID provided in Section 9.2. If the Host finds a future CA device that is both unauthorized (Byte 0 bit 0 of Section 9.2 is 0.) and an ID that matches this value, then the chaining process is invoked. (See Annex I (informative)).

12.1 Get Number of Chaining Blocks

This command is used to initiate the process of chaining: transferring CA data from the current CA module to a new replacement module.

The data message consists of a byte indicating the number of blocks to read and a second byte indicating block length.

Required: Yes

Default message: 0,0. If these values are given, then this CA module does not support the chaining feature.

CLA	INS	P1	P2	P3
0x80	0x40	0	0	2

Message
Number of blocks to read, Length of each block

12.2 Get Block

P1 contains an ID number of the block to read, while P2 contains the total number of blocks of chaining data to be read. The CE device receives the value of P2 as the first byte of Section 12.1. (Blocks are numbered from 0 to "total blocks-1".) P3, the data length, is the length of an individual block and is the second byte returned in Section 12.1.

Required: No

Default message: none

CLA	INS	P1	P2	P3
0x80	0x42	block_ID	Total blocks	Block length

Message
Data block from CA module

12.3 Put Block

P1 contains an ID number of the block to write, while P2 contains the total number of blocks of chaining data to be written. P3, the data length, is the length of an individual block returned in Section 12.1.

Required: No

Default message: None

CLA	INS	P1	P2	P3
0x80	0x44	block_ID	Total blocks	Block length

Message
Data block from CE device

13 Reportback Commands

After receiving a Section 9.3 status message with bit 1 set, the CE host device shall execute commands 13.1 and (if indicated) 13.2 to open a reportback channel.

After connecting, the host shall send a section 13.5 status command, and execute multiple Section 13.3 commands to send data from the CA device to the call center. The host shall maintain the connection until both of the following occur:

- 1) All data for return communications has been sent
- 2) Either all data for forward communications is received (NFWD x SFWD bytes—values are defined in Section 13.2), OR S seconds pass after the connection is made.

When the reportback connection is broken either due to a CA module request (bit 5 of Section 9.3), a dropped line/callback center hangup, or completion of the call, a Section 13.5 status command will be sent to the module.

13.1 Get Open Reportback Request

This command returns the value N from the CA module. If N=0, then there are no parameters and the reportback channel should not be opened. If N>0, then the Get Reportback Values command will return N bytes, and the reportback channel should be opened.

CLA	INS	P1	P2	P3
0x80	0x50	0	0	2

Message
N

Required

13.2 Get Reportback Values

This command returns the following data:

Byte 0	The number of blocks of data to read, NRET.
Byte 1	The number of bytes per block, SRET.
Byte 2	The number of blocks of data to receive from the callback center, NFWD.
Byte 3	The size of the blocks of data to receive from the callback center, SFWD.
Byte 4	The number of seconds to wait for callback center response, S.
Bytes 5-20	The primary number to call, Phonestring ₁ .
Bytes 21-36	The secondary number to call, Phonestring ₂ .

The parameters phonestring₁ and phonestring₂ are NULL terminated ASCII strings. All others are unsigned byte quantities.

Redialing can be handled either in the host, or in the CA device. The host first calls Phonestring₁. If the host fails to connect to this number then if phonestring₂ is not NULL, it is tried after 10 seconds. If both numbers fail then the host will retry at a later time. The redial algorithm is private to the host and not specified here. If Phonestring₂ is NULL, then the host does not redial, and returns a failure after one unsuccessful attempt to connect with Phonestring₁.

CLA	INS	P1	P2	P3
0x80	0x52	0	0	36

Message
see above

13.3 Get Reportback Message

This command returns a block of M data bytes from the CA module to send to the reportback channel. The block number in P1 will be a value from 0 to (K-1), where K, in P2, is the total number of blocks, defined in the data returned with Section 13.2.

CLA	INS	P1	P2	P3
0x80	0x54	block_number	K	M

Message
Block of M data bytes to send to reportback channel

13.4 Put Reportback Buffer

This command sends M data bytes from the reportback channel to the CA module.

CLA	INS	P1	P2	P3
0x80	0x56	0	0	M

Message
M data bytes to send to CA module from reportback channel

13.5 Put Reportback Status

This command sends a status byte for the reportback channel. The byte has one of the following values:

0x00 = Connection made –Speed not specified
 0x01 = Connection made –300 bps
 0x02 = Connection made –1200 bps
 0x03 = Connection made –2400 bps
 0x04 = Connection made –9600 bps
 0x05 = Connection made –14400 bps
 0x06 = Connection made –28800 bps
 0x07 = Connection made –56000 bps
 0x11 = Busy
 0x12 = No answer
 0x13 = No dial tone available
 0x14 = Carrier lost

CLA	INS	P1	P2	P3
0x80	0x58	0	0	1

Message
Status

14 Locks and Limits Commands

14.1 Get PIN

This command reads the PIN (password) from the CA module. It is only used in systems in which the CE device application software is doing the checking of the PIN, and not the CA module.

Required: No

Default message: None

CLA	INS	P1	P2	P3
0x80	0x70	0	0	4

Message
User PIN

14.2 Put PIN

When the CA module is locked by PIN, the parental rating and spending limits specified by the master-user are in effect: the CA module does not allow decryption of the event content if one of the limits is exceeded.

This command is used for five different purposes: (1) to send a new PIN (password) selected by the master user to the CA module and impose locking according to master-user-specified limits; (2) to temporarily override the limits by PIN, without changing the locked state of the system; (3) to unlock the CA module; (4) for any user (master or secondary) to change their user PIN; (5) for identification of a user to the CA module at the beginning of viewing.

P1 indicates the action being performed:

- = 0 Temporary override of the user limits (case 2 in above explanation)
- = 1 PIN change (locking the system) or removal (unlocking) (cases 1,3 and 4 above)
- = 2 Identification of current user to CA module (case 5 above)

P2 is the user number. 0 is the number of the master user, who sets the limits and lock state.

P3 is the length of the data message, 8 bytes.

Data are defined as:

Byte 0 - bit 7=1 for previously locked, =0 for previously unlocked
 bits 3-0= first digit of old PIN

Bytes 1-3 bits 3-0= second, third and fourth digits of old PIN
 Byte 4 bit 7=1 to set system lock, =0 to set system as unlocked
 bits 3-0= first digit of new PIN
 Bytes 5-7 bits 3-0= second, third, and fourth digits of new PIN

If P1 = 0, the meaning of the data message is the following:

Bytes 0-3 PIN of this user (who is the master user); if correct, the limits are overridden;
 Bytes 4-7 These bytes are ignored

If P1 = 1 and a new PIN has been entered for locking of the system, then the meaning of the data message is the following:

Bytes 0-3 Old PIN of this user (who is the master user); if correct, the CA module will change the PIN to the new value;
 Bytes 4-7 New PIN for the master user.

The most significant bit of the most significant byte is the new locked state, which in this case = 1 (locked)

A variation of this case is when the user wants to impose a locked state on the previously unlocked system; in this case the value of the old PIN is ignored, as is the locked state bit.

If P1 = 1, and the master user is unlocking the system, data is the following:

Bytes 0-3 Old PIN of this user (who is the master user); if correct, the CA module will change the lock state bit to zero, canceling the locked state
 Bytes 4-7 New PIN for the master user is ignored
 The most significant bit of the most significant byte is the new locked state, which in this case = 0 (unlocked)

In systems in which the CE device and not the CA module performs the PIN checking, the values of P1 and P2 are 0, the first four bytes of the data message contain the lock state bit followed by the PIN to be stored in the CA module, and the remaining four bytes are ignored.

Required: Yes, if number of PINs stored by the CA module (Section 9.2 byte 56) is greater than zero.

Default message: None

CLA	INS	P1	P2	P3
0x80	0x72	Action type	User number	8

Message
Old lock state bit, Old user PIN, New lock state bit, New user PIN

14.3 Get PIN Action Result

This command reads the result of the Put PIN command (Section 14.2) from the CA module. It should immediately follow that command.

The result of the PIN action defined in the data message has the following possible values:

= 0 Action accepted by CA module
 = 1 Action failed

Required: No

Default message: None

CLA	INS	P1	P2	P3
0x80	0x74	0	0	1

Message
Result of PIN action

14.4 Get Lock State

This command reads the current lock state from the CA module.

The current lock state defined in the data message has the following possible values:

- = 0 System unlocked
- = 1 System locked
- = 2 Temporary override of limits

Required: No

Default message: None

CLA	INS	P1	P2	P3
0x80	0x76	0	0	1

Message
Lock state

14.5 Get User Parameter

This command retrieves user parameter values from the CA module for display by the host. These parameters may be set using Sections 14.5 and 14.6, but CA devices shall also support setting them through menus in section 16 as well.

Required: No

Default message: None

P1 contains the user parameter type being retrieved:

- = 0 Spending limit
- = 1 Monthly spending limit. The data length is always 2 bytes. The spending limit is represented in cents.

CLA	INS	P1	P2	P3
0x80	0x78	User parameter type	User number	2

Message
User parameter

14.6 Put User Parameter

This command sends newly chosen user parameter values from the host to the CA module. The CA module will not accept the parameter sent by this command unless the module is in the unlocked state.

Required: No

Default message: None

P1 is the user parameter type being sent:

- = 0 Spending limit
- = 1 Monthly spending limit

See explanation of data length and meaning in the Get User Parameters command (Section 15.5).

CLA	INS	P1	P2	P3
0x80	0x7A	User parameter type	0	2

Message
User parameter

15 Service Changes and Related Commands

15.1 Put Add-Service Packet Identification (PID) Numbers

This command allows the CE device to send the service PID number and the ECM PID number information to the CA module. Service PIDs are the streams that the card is to descramble, and the ECM PID numbers are the identifiers of the stream where the ECMs for the service PID will be found. This is a complete list of PID pairs. It is not a list of additional service PIDs. If a PID is already being descrambled then the receipt of this message shall not disturb that process. The message is sent from the CE device to the CA module. The parameter N is the total number of bytes in the message. Each PID number is sent as a 13 bit integer represented by the 13 least significant bits of two consecutive bytes, with the most significant byte first. Also, each service PID is paired with and immediately followed by its ECM PID. If the ECM PID is zero then the module will parse the SI stream to find the ECM PID.

CLA	INS	P1	P2	P3
0x80	0x80	0	0	N

Message
pid ₁ ,... , pid _j

15.2 Put Remove-Service Packet Identification (PID) Numbers

This command allows the CE device to send the service PID number and the ECM PID number information to the CA module for services that are no longer valid. These are PIDs that the card is to stop descrambling, and the ECM PID numbers are the identifiers of the stream where the ECMs for the service PID were found. This is a complete list of PID pairs. If a PID is not included here then the receipt of this message shall not disturb that service. The message is sent from the CE device to the CA module. The parameter N is the total number of bytes in the message. Each PID number is sent as a 13 bit integer represented by the 13 least significant bits of two consecutive bytes, with the most significant byte first. Also, each service PID is paired with and immediately followed by its ECM PID. If the ECM PID is zero then the module will parse the SI stream to find the ECM PID. This command would be used when changing audio(or data) services without changing the complete service .

If P1 =1, then the CA module detunes all PID's (as in a channel change) otherwise only the listed PIDs are removed.

CLA	INS	P1	P2	P3
0x80	0x82	=1 Detune all services	0	N

Message
pid ₁ ,... , pid _j

15.3 Get Decode Status for PIDs

This command is used to obtain a response to the "Put Add-Service PID Numbers command (Section 15.1). The CA module will send the status of each service PID that it was requested to decode, and a display message for the user where appropriate. The data message is sent from the CA module to the CE device.

The message length (L) is 4 times the number of PIDs. L should be calculated from the maximum number of concurrent PIDs. This value is returned in byte 55 of Section 9.2.

Each PID number is sent as a 13 bit integer represented by the 13 least significant bits of two consecutive bytes, with the most significant byte first:

MS Byte: Bits 7-5 spare
 Bits 4-0 5 Most Significant Bits of PID
 LS Byte: Bits 7-0 8 Least Significant Bits of PID

PID Decode Status Byte: This byte is a bitmap defined as follows:

- Bit 0 =1 CW is valid
- Bit 1 =1 ECM not recognized by this device
- Bit 2 =1 ECM not valid Status not determined
- Bit 3 =1 The service on this PID is purchasable; will offer purchase
- Bit 4 =1 Parental rating exceeded
- Bit 5 =1 Spending limit exceeded
- Bit 6 reserved
- Bit 7 Meaning of text byte:
 - =1 Text byte is number of a stored display text associated with this PID
 - =0 Text byte is the size of a display message for this PID
(Use Section 16.2 to read the message.)

The text byte indicates a display text message for the user where appropriate. It can have the following values and meanings:

If Bit 7 of the status byte = 1, then this byte value is the number of a display message text stored in the CE device(see appendix H).

If the value of this number is 0, clear the previous message display.

If Bit 7 of the status byte = 0, this byte value is the length M of a display message for this PID that the host must request using Section 16.2. If the value of this number is 0, clear the display message associated with this PID.

CLA	INS	P1	P2	P3
0x80	0x84	0	0	L

Message
pid ₁ , status ₁ , text ₁ , ..., pid _i , status _i , text _i

15.4 Put CE Device PowerState

This command informs the CA module of the CE device's power on/off/standby status. This information may then be used for gathering viewing information. This command must be called on each CA module reset and when the Power state of the CE device changes.

Required: Yes

Default message: None

The data message contains a value representing the power state of the CE device (host):

- = 0 Host is in standby
- = 1 Host is powered on
- = 2 Host is on but user is navigating through menus and not viewing the current service (Optional for viewer statistics)

CLA	INS	P1	P2	P3
0x80	0x86	0	0	1

Message
Power state of CE device

15.5 Get Forced Service Change

This command is used to read the new service ID from the CA module after status bit 16 was set in Section 9.3 (Forced Tuning).

Hosts shall only respond to this request if one of two conditions are met:

- a) The CA device's CA system ID is included in the CA table for the current physical channel. (Forced service changes are only allowed by CA devices that are included in the current 'system' being viewed.)
- b) The host has informed the CA module that it is available for homing. (See Section 9.6)

Required: Yes

Default message: None

The data message is the Service ID to be tuned to.

CLA	INS	P1	P2	P3
0x80	0x88	0	0	12

Message
Unique Event Descriptor

15.6 Put EMM PID

This command is used by the CE device to send to the CA module the PID on which EMMs are to be received for this CA system. It shall be used whenever the CA module is reset.

Note: this assumes that the CE device is parsing the Conditional Access Table (CAT).

Required: Yes

Default message: None

CLA	INS	P1	P2	P3
0x80	0x8A	0	0	2

Message
PID for EMMs

16 Display and Menu Commands

16.1 Get Display Message Size

This command returns the number of bytes to read, M, the number of seconds to wait, S. M and S are unsigned byte quantities.

CLA	INS	P1	P2	P3
0x80	0xA0	0	0	2

Message
M, S

16.2 Get Display Message

This command returns M ASCII characters from the CA module to display as a message.

If the display message results from the processing of a Service (ECM), P1 and P2 contain the PID number of the Service ID (ECM). This number is returned from Section 15.3 (Get Decode Status for PIDs).

The PID number is an integer represented by the 13 least significant bits of the two consecutive bytes P1 and P2, where P1 is the most significant byte:

- P1: Bits 7-5 spare
Bits 4-0 5 Most Significant Bits of PID
- P2: Bits 7-0 8 Least Significant Bits of PID

The host gives a value of 0 for the PID when the display message is unrelated to a particular Service (ECM). In this case, the host is requesting the display message after status bit 2 was set in the returned data from Section 9.3 (Get Status).

CLA	INS	P1	P2	P3
0x80	0xA2	5 msb's of pid	8 lsb's of pid	M

Message
M characters to display

16.3 Get Alpha Box Request

This command returns the value N from the CA module. If N=0, then there is no request for user information. If N>0, then the Get Alpha BoxParameters command will return N bytes.

CLA	INS	P1	P2	P3
0x80	0xA4	0	0	1

Message
N

16.4 Get Alpha Box Parameters

This command returns the number of bytes to read, M, the number of seconds to wait, S, and the prompt to be displayed. The msb of M is a flag. If the msb is 1, the alpha box should accept alphanumeric entries, if the msb is 0, the alpha box should only accept digits. The remaining bits of M are the number of bytes (characters/numbers) the user should enter. The maximum number is 16. S is an unsigned byte. The parameter prompt is a NULL terminated ASCII string.

CLA	INS	P1	P2	P3
0x80	0xA6	0	0	N

Message
M, S, prompt

16.5 Put Alpha Box Input

This command in conjunction with "Get Alpha Box parameters"(sect 16.4) conveys the user response of M bytes to the CA module. The bytes are coded as an ASCII string, unentered bytes are NULLs.

CLA	INS	P1	P2	P3
0x80	0xA8	0	0	M

Message
M data bytes

16.6 Get Menu Request

This command returns the value N from the CA module. If N=0, then there is no menu to paint. If N>0, then there is a menu and N is the number of bytes to be returned in the Get Menu Parameters command. N is an unsigned byte and may be at most 255.

If P2=1, then the host is requesting the CA module to send its top level menu.

CLA	INS	P1	P2	P3
0x80	0xAA	0	0 or 1	1

Message
N

16.7 Get Menu Parameters

This command produces a menu from which the user may make a selection. The message of N bytes. The value of N is obtained from Section 16.6. A menu is generated by the CE device using the parameters in the message. The 8-bit number M is the number of entries in the menu. M may range from 1 to 10. The 1 byte parameter S is the number of seconds to display the menu before returning a no response reply, and erasing the menu. If S=0, then display the menu indefinitely until the user responds. The title is a NULL terminated ASCII string.

Each entry, text_i, is a NULL terminated ASCII string. To allow better management of the UI by host devices, the lines that cannot be selected are marked. A non 'selectable' line may have text on it, but the text is help or information, not a valid menu choice. If a particular entry is not a 'selectable' menu item, the MSB of the first byte is 1, otherwise it is 0.

CLA	INS	P1	P2	P3
0x80	0xAC	0	0	N

Message
M, S, title, text ₁ , ..., text _M

16.8 Put User Response

This command conveys the user's response to a menu display. This returns the byte value R = 0xff if there is no response before the menu times out. If the user cleared or exited the menu /dialog box R=0xFE is returned. If the user selects the i-th menu item then a value of R=i is returned. Values for 1-9 are 0x1-0x9 for the tenth menu item 0x00 is returned .

CLA	INS	P1	P2	P3
0x80	0xAE	0	0	1

Message
R

17 Mail Handling Commands

17.1 Get Mail Message Length

This command conveys the user's response to status bit #15 in the "GetCAModuleStatus" command (see Section 9.3). This command may be used to read the length of the mail message received by the CA device.

CLA	INS	P1	P2	P3
0x80	0xB0	Message number	0	1

Message
Length of pending Mail Message

17.2 Get Mail Message Text

This command is used to read the text of the message number specified in 'P1'. The Host uses this command to read the text of the message. Messages shall be null terminated (0x0) ASCII text.

CLA	INS	P1	P2	P3
0x80	0xB2	Message Number	0	N

Message
R

18 Inter-card Message Commands

18.1 Get Inter-card Message Sizes

This command returns the number of bytes to read, M, and the number, N, of the CA module to which to send the message. M, N, and K are unsigned byte quantities. K is the number of messages to get from this card and to put to card number N. A single get followed by a put should be performed at approximately 0.1 sec intervals until K messages have been received and sent. If K is 1 then the operation is performed immediately.

CLA	INS	P1	P2	P3
0x80	0xC0	0	0	3

Message
M, N, K

18.2 Get Inter-card Message

This command returns M characters from the CA module to send to another CA module.

CLA	INS	P1	P2	P3
0x80	0xC2	0	0	M

Message
M characters to send to another CA module

18.3 Put Inter-card Message

This command writes M characters of message to the target CA module.

CLA	INS	P1	P2	P3
0x80	0xC4	0	0	M

Message
M characters to send to another CA module

18.4 Put Card Numbers

This command sends the total number of CA modules, N, and the position of this CA module in the stack, P. The first CA module is numbered 1. N and P are unsigned byte quantities. The host will also send the CA descriptor ID's of all CA devices in the system to facilitate Inter-card communication.

CLA	INS	P1	P2	P3
0x80	0xC6	0	0	2+2*N(Number of slots)

Message
N,P,CA ID numbers for all cards

19 NVM Storage Commands

19.1 Put NVM Available

This command delivers a 4 byte unsigned integer msb first which is the total number of bytes of NVM available from the CE device to be allocated to the CA module.

CLA	INS	P1	P2	P3
0x80	0xD0	0	0	4

Message
Available NVM

19.2 Get Allocation Size

This command reads a 1 byte unsigned integer which is the number of bytes of NVM to be allocated for this request.

CLA	INS	P1	P2	P3
0x80	0xD2	0	0	1

Message
Number of bytes to allocate

19.3 Put Allocation Tag

This command delivers a 4 byte unsigned integer msb first which is the tag uniquely identifying this allocation. A tag value of 0xffffffff denotes allocation failure. The size is an unsigned one byte value.

CLA	INS	P1	P2	P3
0x80	0xD4	0	0	5

Message
Tag, size

19.4 Get Operation and Tag

This command gets a 4 byte unsigned integer msb first which is the tag uniquely identifying an allocation. A tag value of 0xffffffff denotes failure. The operation is a byte which is a 1 for write, 2 for read, and a 3 for delete. This command results in the CE device performing the operation in case of read or delete. In case of a write operation the CE device must get the block to write to NVM from the CA module. The tag size is specified here as a one byte unsigned value.

CLA	INS	P1	P2	P3
0x80	0xD6	0	0	6

Message
Tag, operation, Tag Size

19.5 Put Operation Results

This command returns a 4 byte unsigned integer msb first which is the tag uniquely identifying an allocation. A tag value of 0xffffffff denotes failure. The operation is a byte which is a 1 for write, 2 for read, and a 3 for delete. The result is a byte which is a 0 for failure and a 1 for success.

CLA	INS	P1	P2	P3
0x80	0xD8	0	0	6

Message
Tag, operation, results

19.6 Put Read Data

This command returns data read from NVM back to the CA module. The size is derived from the operation tag previously specified by the CA module. The sts is a 0 if the read was a failure, and a 1 if success.

CLA	INS	P1	P2	P3
0x80	0xDA	0	sts	Size

Message
Data

19.7 Get Write Data

This command reads data from the CA module. The size is derived from the operation tag previously specified by the CA module. The data then will be written to NVM and the success or failure of that operation will be returned to the CA module with a Put Operation Results command.

CLA	INS	P1	P2	P3
0x80	0xDC	0	0	size

Message
Data

20 Copy Protection Framework

The NRSS standard allows for protecting A/V data as it is returned from the CA module after processing. See CEA-796-A for further information concerning NRSS copy protection systems. The copy protection framework is used to protect (scramble) the content carried by the MPEG2 transport stream after it has been descrambled by the CA module. Diffie-Hellman(DH) or another equivalent key exchange mechanism may be used to establish a shared key and DES may be used as a local rescrambling function.

The framework is defined in a generic fashion so that different underlying copy protection technologies can be used in a common way. Resource types are defined within the framework to support communication using a common object set. While many of the common elements of the CEA-796-A copy protection systems are defined in this section, it should be realized that other elements including authentication may be necessary to meet content provider requirements.

All the copy protection systems use a common set of commands but the lengths may vary when passing parameters between the host and module depending on the CP system in use.

20.1 Service Provider [Informative]

Involvement of the service provider, i.e. headend or broadcast station, through the private CA system, may be used instead of, or, in addition to CA module processing and control. After obtaining the requisite host information, the service provider, through the CA system, may selectively de-authorize CA modules based on various criteria, for example, if CA modules are found to be releasing copy protected content to unprotected hosts.

20.2 Back Channel [Informative]

A back channel may be used to return host information to the service provider. The back channel may be automatic (i.e. using Out-of-Band modem, cable modem or Telephone), or may be manual, where the user recites information from the display to an operator on the phone. In the manual case, the user can be considered to be the "back channel". The same information may be conveyed to the Service Provider in either the manual case or in the automatic case. Existing NRSS commands may be employed by the CA module to cause the data to be displayed.

20.2.1 Service Provider Processing [Informative]

20.2.1.1 Checking Validity of Host Information [Informative]

The service provider may be used to check the validity of host information being delivered (i.e. host certificate) by using the appropriate method required for the copy protection technology used.

20.2.1.2 Host Revocation [Informative]

The service provider may be used to perform revocation of the host. Using proprietary CA mechanisms, revocation of the host may entail:

- Complete loss of service
- Loss of particular channels (e.g. "HBO" or "Showtime")
- Loss of individual programs on channels

The service provider may then use proprietary CA mechanisms to deliver a “pass/no pass” on a particular host device, or to deliver new copy protection parameters to the CA module. By involving the service provider in the process, a number of different underlying copy protection technologies may be accommodated.

20.3 CP System Negotiation

The Copy Protection system in NRSS shall first require a negotiation session between the Host and Module to determine which systems are implemented, if one shall be used and how many streams are supported. The chosen system shall be signified by the CP_system_ID (see Table 9). See CEA-796-A for additional information concerning these systems.

CP_system_ID	id value(binary)
System 1	xxx0 0001
System 2	xxx0 0010
System 3	xxx0 0011
System 4	xxx0 0100
System 5	xxx0 0101

Table 9 CP_system_ID

As part of the Module initialization by the host, bitfields in the **PutHostConfiguration** (Section 9.6) shall be used to signal the CA module which copy protection systems and how many unique rescrambling keys are supported by the Host.

The CA module shall signal the Host using a bit in the **GetCAModuleStatus** (Section 9.3) command in the event that a common CP system was chosen. Upon receiving this signal, the Host shall call **GetCAModuleConfiguration**. Byte 58 of **GetCAModuleConfiguration** shall indicate which CP system has been chosen.

20.4 CA Module/Host Binding

The CA module and the host may negotiate a long-term shared secret. This procedure shall be based on either a Discrete Logarithm DH, an Elliptic Curve DH or a Needham-Shroeder key exchange. Some of the CP systems described in CEA-796-A specify the use of one of these methods for key exchange and may specify a function to permute a value in the key derivation process in a licensable manner.

20.4.1 Discrete Logarithm DH

The Discrete Logarithm DH method is summarized as follows¹:

The CA module sends its public key $g^p \bmod n$ to the host.

The host sends its public key $g^h \bmod n$ to the CA module

The host takes $g^p \bmod n$ from the POD, and forms $(g^p \bmod n)^h \bmod n = g^{ph} \bmod n$.

The CA module takes $g^h \bmod n$ from the host, and forms $(g^h \bmod n)^p \bmod n = g^{ph} \bmod n$.

The CA module and host have now negotiated a shared secret $g^{ph} \bmod n$. The most significant 128 bits of this secret value shall be defined as K_s .

The constants (g and n) necessary for implementing this key exchange protocol shall be established by the licensor of the CP System. The maximum length of the prime modulus n is 1024 bits, export permitting. If desired, a smaller modulus may be used by padding the MSBs of bitfields with zeros.

20.4.2 Elliptic Curve DH

The Elliptic Curve DH method described is based on the ECSVDP-DH primitive defined in IEEE P1363/D3 (May 11, 1998). It is summarized as follows:

¹ A full description is available in Applied Cryptography Second Edition, Bruce Scheier, John Wiley & Sons Inc, p. 513.

The CA module sends its public key $G * p$ to the host.
 The host sends its public key $G * h$ to the CA module
 The host takes $G * p$ from the CA Module, and computes $(G * p) * h$.
 The CA module takes $G * h$ from the host, and computes $(G * h) * p$.

The CA Module and Host have now negotiated a shared secret $G * h * p$. Take the most significant 128 bits of the x coordinate of this secret as K_s .

The licensor of the CP system shall establish the basepoint G and other constants necessary for implementing this key exchange protocol. The maximum order of the Basepoint G and the field size is 160 bits, export permitting. If desired, a smaller order basepoint can be used.

20.4.3 Needham-Schroeder Key Exchange Protocol

The modified Needham-Schroeder key exchange protocol is summarized as follows²:

- The CA module generates a 64 bit random number, R_{CAModule} , and 128 bits of true random key material, $K_{\text{RandCAModule}}$, and encrypts both values using the public key of the host. This encrypted value is sent to the host.
- The host decrypts the value received from the CA module using the private key of the host thus retrieving R_{CAModule} and $K_{\text{RandCAModule}}$.
- The host generates a 64 bit random number, R_{Host} , and 128 bits of true random key material, K_{RandHost} , and encrypts both values and the random number R_{CAModule} using the public key of the CA module. This encrypted value is sent to the CA module.
- The CA module decrypts the value received from the host using the private key of the CA module thus retrieving R_{Host} , K_{RandHost} and R_{CAModule} .
- The CA module compares R_{CAModule} received and decrypted in step (4) with the random number sent to the host in step (1). If they are equal then the CA module sends the random value, R_{Host} back to the host. Otherwise the key exchange fails.
- The host compares R_{Host} sent to the host in step (5) with the random number sent to the CA module in step (3). If they are equal then the key exchange has completed successfully. Otherwise the key exchange fails.

After completing this protocol, the CA module and host have 256 bits of shared true random key material, $K_{\text{RandCAModule}}$ and K_{RandHost} , on which to base a common initial session key. The least significant 64 bits of $K_{\text{RandCAModule}}$ concatenated with the most significant 64 bits of K_{RandHost} is used as K_s .

20.4.4 Computation of K_{cpss} and $K_{\text{validated_cpss}}$

The CA module and host shall generate CP system specific 192-bit values $K_{\text{validated_cpss}}$ and K_{cpss} respectively. A licensable cryptographic function may be used to generate K_{cpss} and $K_{\text{validated_cpss}}$ from K_s . Additionally, some CP systems may require that the host device concatenate another value such as the host's ID to the permuted K_s to form K_{cpss} . CA modules may concatenate a host ID which has been validated by the head-end via the CA system (or at time of purchase) to the permuted K_s to form $K_{\text{validated_cpss}}$. Each CP system shall specify the cryptographic function used to permute K_s and procedure to generate K_{cpss} and $K_{\text{validated_cpss}}$.

This computation need occur only once, at the time CA module and host pair are bound. The resulting shared secret values K_{cpss} and $K_{\text{validated_cpss}}$ may be stored in non-volatile memory, and used to generate session keys later, in the key derivation phase.

20.5 Copy Control Information Definition

The format of copy control information (CCI) passed across the data channel consists of three components: baseline CCI, reserved CCI, CP system private CCI. The specific format is outlined in Table 10 through Table 13.

² A full description is available in Handbook of Applied Cryptography, A. Menezes, P van Oorschot, S Vanstone, CRC Press, 1997, p. 508 and note 12.39.