
**Information technology — Generic digital
audio-visual systems —**

**Part 7:
Basic security tools**

*Technologies de l'information — Systèmes audiovisuels numériques
génériques —*

Partie 7: Outils de sécurité de base

Reference number
ISO/IEC 16500-7:1999(E)



PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO/IEC 1999

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 734 10 79
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents	Page
Foreword	vii
Introduction.....	viii
1. Scope.....	1
2. Normative references	1
3. Definitions	2
4. Acronyms and abbreviations	3
5. Conventions	4
6. Overview of Security System	4
7. Security Tools	5
7.1 S1 Scrambling	5
7.1.1 Scrambled Elements	5
7.1.2 Control Word Synchronization	6
7.2 S2/S3 Authentication	7
7.2.1 Authentication Protocol.....	7
7.2.2 Syntax of Authentication Messages.....	8
7.2.3 Key Use.....	9
7.2.4 Certificates	9
7.2.5 Integration with Protocols for S2	9
7.3 S2/S3 Confidentiality and Integrity.....	9
7.3.1 Negotiation of Confidentiality and Integrity Algorithms.....	10
7.3.2 Definition of the SPI value	10
7.3.3 Signaling	10
7.3.4 Replay Protection on S2	10
7.4 S2 Digital Signatures	10
7.5 DSM-CC Commands for S1 Security Management	10
7.5.1 Security Association Configuration.....	10
7.5.2 Key Retrieval	12
7.6 Secure Download	13
7.6.1 Format of Security DownloadInfoRequest	13
7.6.2 Format of Security DownloadInfoResponse.....	14
7.6.3 Format of Security Module	14
7.6.4 Download Reasons in DownloadDataResponse.....	15
7.7 Parental Control	15
7.7.1 Introduction.....	15
7.7.2 Retrieval Services.....	15
7.7.3 Distribution Services	15
8. Flows and Protocol Stacks.....	16
8.1 Sample Flow: Scrambled Video on Demand (Informative).....	16

9.	Security Interfaces	19
10.	Security Interface CA0	20
10.1	Introduction	20
10.2	Additional DAVIC Requirements for CA0.....	20
10.2.1	Host to Security Device Authentication	20
10.2.2	Security Services	20
11.	Profiles/Contours.....	21
12.	Security Interface CA1	21
12.1	Introduction and scope	21
12.1.1	Introduction.....	21
12.1.2	CA1 reference model	21
12.2	Notation.....	23
12.3	Physical characteristics of the CA1 security device.....	23
12.4	Electronic signals and transmission protocols on the CA1 interface	23
12.4.1	Additions and restrictions to ISO/IEC 7816-3	23
12.4.2	Logical channels.....	24
12.5	CA message format and filter specification	24
12.5.1	CA message mechanism and format.....	24
12.5.2	Filtering of a CA message	24
12.5.3	Filter specification.....	25
12.5.4	Filter programming	25
12.5.5	Filter objects.....	27
12.5.6	Response objects from smart card.....	31
12.5.7	Filter conditions	32
12.5.8	Further filter requirements.....	32
12.6	Initialization of the smart card and application.....	33
12.6.1	Answer-to-Reset.....	33
12.6.2	Conditional PTS procedure	35
12.6.3	The ATR and/or DIR file	35
12.6.4	Card identification and initialization data	35
12.6.5	Application identification and selection	36
12.6.6	Application-independent card services.....	37
12.7	Smart card security functions.....	37
12.7.1	Access conditions.....	37
12.7.2	Commands and access conditions	38
12.7.3	Password management	39
12.7.4	Authentication.....	39
12.8	Data structures in the smart card.....	40
12.8.1	Introduction.....	40
12.8.2	File organization.....	40
12.8.3	Data structure headers	42
12.8.4	EF file content.....	45

12.8.5	Overview of DAVIC data objects	46
12.9	Basic data objects	47
12.9.1	Introduction	47
12.9.2	Application_related_data	47
12.9.3	System_related_data	48
12.9.4	Service_provider_related_data	49
12.9.5	Filtering	49
12.9.6	Parental_rating	50
12.9.7	Entitlement data	50
12.9.8	Response messages	51
12.10	Smart card commands	53
12.10.1	Introduction	53
12.10.2	Coding of the commands	53
12.10.3	Select_file	53
12.10.4	Read_binary	54
12.10.5	Read_record	55
12.10.6	Seek	56
12.10.7	Verify_password	57
12.10.8	Change_password	57
12.10.9	Disable_password	58
12.10.10	Enable_password	59
12.10.11	Write_binary	59
12.10.12	Write_record	60
12.10.13	Update_binary	61
12.10.14	Update_record	61
12.10.15	Get_response	62
12.10.16	Get_data	63
12.10.17	Put_data	64
12.10.18	Get_application_status	65
12.10.19	Perform_security_operation	66
12.10.20	Status Conditions returned by the smart card	66
12.11	Man machine interface	69
12.11.1	Introduction	69
12.11.2	MMI objects	70
12.12	Tag allocation	72
12.12.1	General	72
12.12.2	Tag allocation for DAVIC application	73
13.	Additional Resources for the DAVIC CA0 Interface	79
13.1	Host-provided Resources	79
13.1.1	TCP/IP Socket Resource	79
13.1.2	HTML MMI Display	86
13.2	PC card-provided Resources	87
13.2.1	Authentication Support	87
13.2.2	IP Security Support	89

14. Informative Methodology 91

14.1 Procedure 91

14.2 Logical Business Model 91

14.3 Systems Review 92

14.4 Threat Analysis 92

14.4.1 Threat Classes 92

14.4.2 Threats for Content Flow S1 93

14.4.3 Threats for Application Control Flow S2 94

14.5 Risk Assessment 94

14.5.1 S1 Flow 96

14.5.2 S2 Flow 97

14.6 Requirements Resulting from the Risk Analysis 97

14.6.1 S1 Flow 97

14.6.2 S2 Flow 98

14.6.3 Legislative And Regulatory Requirements 98

14.7 Security Services 99

14.7.1 General Security Services 99

14.8 Security Mechanisms 100

Bibliography 101

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 16500 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 16500-7 was prepared by DAVIC (Digital Audio-Visual Council) and was adopted, under the PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

ISO/IEC 16500 consists of the following parts, under the general title *Information technology — Generic digital audio-visual systems*:

- *Part 1: System reference models and scenarios*
- *Part 2: System dynamics, scenarios and protocol requirements*
- *Part 3: Contours: Technology domain*
- *Part 4: Lower-layer protocols and physical interfaces*
- *Part 5: High and mid-layer protocols*
- *Part 6: Information representation*
- *Part 7: Basic security tools*
- *Part 8: Management architecture and protocols*
- *Part 9: Usage information protocols*

Introduction

ISO/IEC 16500 defines the minimum tools and dynamic behavior required by digital audio-visual systems for end-to-end interoperability across countries, applications and services. To achieve this interoperability, it defines the technologies and information flows to be used within and between the major components of generic digital audio-visual systems. Interoperability between these components and between individual sub-systems is assured through specification of tools and specification of dynamic systems behavior at defined reference points. A reference point can comprise one or more logical (non-physical) information-transfer interfaces, and one or more physical signal-transfer interfaces. A logical interface is defined by a set of information flows and associated protocol stacks. A physical interface is an external interface and is fully defined by its physical and electrical characteristics. Accessible reference points are used to determine and demonstrate compliance of a digital audio-visual subsystem with this international standard.

A summary of each part follows.

ISO/IEC 16500-1 (DAVIC 1.3.1a Part 2) defines the normative digital audio-visual systems technical framework. It provides a vocabulary and a Systems Reference Model, which identifies specific functional blocks and information flows, interfaces and reference points.

ISO/IEC 16500-2 (DAVIC 1.3.1a Part 12) defines system dynamic behavior and physical scenarios. It details the locations of the control functional entities along with the normative protocols needed to support the systems behavior. It is structured as a set of protocol walk-throughs, or “*Application Notes*”, that rehearse both the steady state and dynamic operation of the system at relevant reference points using specified protocols. Detailed dynamics are given for the following scenarios: video on demand, switched video broadcast, interactive broadcast, and internet access.

ISO/IEC 16500-3 (DAVIC 1.3.1a Part 14) provides the normative definition of DAVIC Technology Contours. These are strict sets of Applications, Functionalities and Technologies which allow compliance and conformance criteria to be easily specified and assessed. This part of ISO/IEC 16500 contains the full details of two contours. These are the Enhanced Digital Broadcast (EDB) and Interactive Digital Broadcast (IDB). ISO/IEC 16500-3 specifies required technologies and is a mandatory compliance document for contour implementations.

ISO/IEC 16500-4 (DAVIC 1.3.1a Part 8) defines the toolbox of technologies used for lower layer protocols and physical interfaces. The tools specified are those required to digitize signals and information in the Core Network and in the Access Network. Each tool is applicable at one or more of the reference points specified within the Delivery System. In addition a detailed specification is provided of the physical interfaces between the Network Interface Unit and the Set Top Unit and of the physical interfaces used to connect Set Top Boxes to various peripheral devices (digital video recorder, PC, printer). The physical Delivery System mechanisms included are copper pairs, coaxial cable, fiber, HFC, MMDS, LMDS, satellite and terrestrial broadcasting.

ISO/IEC 16500-5 (DAVIC 1.3.1a Part 7) defines the technologies used for high and mid-layer protocols for ISO/IEC 16500 digital audio-visual systems. In particular, this part defines the specific protocol stacks and requirements on protocols at specific interfaces for the content, control and management information flows.

ISO/IEC 16500-6 (DAVIC 1.3.1a Part 9) defines what the user will eventually see and hear and with what quality. It specifies the way in which monomedia and multimedia information types are coded and exchanged. This includes the definition of a virtual machine and a set of APIs to support interoperable exchange of program code. Interoperability of applications is achieved, without specifying the internal design of a set top unit, by a normative Reference Decoder Model which defines specific memory and behavior constraints for content decoding. Separate profiles are defined for different sets of multimedia components.

ISO/IEC 16500-7 (DAVIC 1.3.1a Part 10) defines the interfaces and the security tools required for an ISO/IEC 16500 system implementing security profiles. These tools include security protocols which operate across one or both of the defined conditional access interfaces CA0 and CA1. The interface CA0 is to all security and conditional access functions, including the high speed descrambling functions. The interface CA1 is to a tamper resistant device used for low speed cryptographic processing. This cryptographic processing function is implemented in a smart card.

ISO/IEC 16500-8 (DAVIC 1.3.1a Part 6) specifies the information model used for managing ISO/IEC 16500 systems. In particular, this part defines the managed object classes and their associated characteristics for managing the access network and service-related data in the Delivery System. Where these definitions are taken from existing standards, full reference to the required standards is provided. Otherwise a full description is integrated in the text of this part. Usage-related information model is defined in ISO/IEC 16500-9.

ISO/IEC 16500-9 (DAVIC 1.3.1a Part 11) specifies the interface requirements and defines the formats for the collection of usage data used for billing, and other business-related operations such as customer profile maintenance. It also specifies the protocols for the transfer of Usage Information into and out of the ISO/IEC 16500 digital audio-visual system. In summary, flows of audio, video and audio-visual works are monitored at defined usage data collection elements (e.g. servers, elements of the Delivery System, set-top boxes). Information concerning these flows is then collected, processed and passed to external systems such as billing or a rights administration society via a standardised usage data transfer interface.

Additional Information

ISO/IEC TR 16501 is an accompanying Technical Report. Further architectural and conformance information is provided in other non-normative parts of DAVIC 1.3.1a (1999). A summary of these documents is included here for information.

ISO/IEC TR 16501 (DAVIC 1.3.1a Part 1) provides a detailed listing of the functionalities required by users and providers of digital audio-visual applications and systems. It introduces the concept of a contour and defines the IDB (Interactive Digital Broadcast) and EDB (Enhanced Digital Broadcast) functionality requirements which are used to define the normative contour technology toolsets provided in ISO/IEC 16500-3.

DAVIC 1.3.1a Parts 3, 4 and 5 are DAVIC technical reports. They provide additional architectural and other information for the server, the delivery-system, and the Service Consumer systems respectively. Part 3 defines how to load an application, once created, onto a server and gives information and guidance on the protocols transmitted from the set-top user to the server, and those used to control the set-up and execution of a selected application. Part 4 provides an overview of Delivery Systems and describes instances of specific DAVIC networked service architectures. These include physical and wireless networks. Non-networked delivery (e.g. local storage physical media like discs, tapes and CD-ROMs) are not specified. Part 5 provides a Service Consumer systems architecture and a description of the DAVIC Set Top reference points defined elsewhere in the normative parts of the specification.

DAVIC 1.3.1a Part 13 is a DAVIC technical report, which provides guidelines on how to validate the systems, technology tools and protocols through conformance and / or interoperability testing.

Information technology — Generic digital audio-visual systems — Part 7: Basic security tools

1. Scope

This part of ISO/IEC 16500 deals with security in DAVIC systems. A number of security tools are specified, including security protocols which operate across DAVIC interfaces, as well as two security related interfaces in the STU. These security tools must be implemented in ISO/IEC 16500 systems conforming to the security profiles.

2. Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 16500. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 16500 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau (TSB) maintains a list of currently valid ITU-T Recommendations.

2.1 ISO/IEC and ITU normative references

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

ISO/IEC 7816-2: 1988, *Identification cards - Integrated circuit(s) cards with contacts - Part 2: Dimensions and location of the contacts*.

ISO/IEC 7816-3: 1989, *Identification cards - Integrated circuit(s) cards with contacts - Part 3: Electronic signals and transmission protocols*.

ISO/IEC 7816-4: 1995, *Information technology - Identification cards - Integrated circuit(s) cards with contacts - Part 4: Inter-industry commands for interchange*.

ISO/IEC 7816-5: 1994, *Identification cards - Integrated circuit(s) cards with contacts - Part 5: Numbering system and registration procedure for application identifiers*.

ISO/IEC 7816-6: 1996, *Identification cards - Integrated circuit(s) cards with contacts - Part 6: Inter-industry data elements*.

ISO/IEC 7816-8, *Identification cards - Integrated circuit card(s) with contacts - Part 8: Security related interindustry commands*.

ISO/IEC 8824: 1990, *Information technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)*.

ISO/IEC 8825: 1990, *Information technology - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*.

ISO/IEC 13818-1: 1996, *Information technology - Generic coding of moving pictures and associated audio information: Systems*.

ISO/IEC 13818-6: 1998, *Information technology - Generic coding of moving pictures and associated audio information - Part 6: Extensions for DSM-CC*.

ITU Recommendation X.509, *Information technology - Open System Interconnection - The Directory: Authentication Framework, Amendment 2, 1996*.

ITU Recommendation X.511, *Information technology - Open System Interconnection - The Directory: Abstract Service Definition, Amendment 4, 1996*.

2.2 Other normative references

- CENELEC EN 50221 : 1997, *Common Interface for Conditional Access and other Digital Video Decoder Applications*.
- CENELEC R 206-001 : 1997, *Guidelines for the Implementation and Use of the Common Interface for DVB Decoder Applications*.
- ETSI ETR 162: 1995, *Digital Video Broadcasting (DVB): Allocation of Service Information (SI) codes for DVB Systems*.
- ETSI ETR 332: 1996-11, *Security requirements capture*.
- ETSI ETS 300 468: 1997-01, *Digital Video Broadcasting (DVB) - Specification for Service Information (SI) in DVB systems*.
- ETSI ETS 300 608: 1997-08, *Digital cellular telecommunication system (Phase 2) - Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) Interface (GSM11.11)*.
- ETSI prEN 726-3, *Terminal equipment (TE) - Requirements for IC cards and terminals for telecommunication use - Part 3: Application independent card requirements*.
- Personal Computer Memory Card International Association, *PC Card Standard - Volume 2: Electrical Specification*, February 1995.
- Personal Computer Memory Card International Association, *PC Card Standard - Volume 3: Physical Specification*, February 1995.
- Personal Computer Memory Card International Association, *PC Card Standard - Volume 4: Metaformat Specification*, February 1995.

3. Definitions

This clause defines new terms, and the intended meaning of certain common terms, used in this part of ISO/IEC 16500. Annex A of ISO/IEC 16500-1 defines additional terms and, in some cases, alternative interpretations that are appropriate in other contexts. For convenience, the normative definitions below are included in the annex.

- 3.1. access control:** Provides means to access services and protection against unauthorized use of resources, including protection against the use of resources in an unauthorized manner.
- 3.2. conditional access:** A means of allowing system users to access only those services that are authorized to them. In general, conditional access relates to environments in which data is broadcast.
- 3.3. confidentiality:** The protection of information from unauthorized disclosure.
- 3.4. Control Word (CW):** The secret key used for a scrambling algorithm.
- 3.5. data integrity:** The detection or protection of unauthorized modification of data.
- 3.6. data origin authentication:** Corroboration that the identity of the source of data is as claimed.
- 3.7. Entitlement Control Message (ECM):** Conditional access messages carrying an encrypted form of the control words or a means to recover the control words, together with access parameters, ie., an identification of the service and of the conditions required for accessing this service.
- 3.8. Entitlement Management Message (EMM):** Conditional access messages used to convey entitlements or keys to users, or to invalidate or delete entitlements or keys.
- 3.9. host:** A piece of equipment where one or more devices can be connected (eg. an IRD, a VCR, a PC or a STU).
- 3.10. key management:** The generation, storage, distribution, archiving, deletion, revocation, registration, and deregistration of cryptographic keys.
- 3.11. device:** A small piece of hardware, not working by itself, designed to run specialized tasks in association with a host, or to provide resources required by an application but not provided directly by the host. Examples include a conditional access sub system or an electronic program guide application device.

- 3.12. non-repudiation:** The proof of the origin and reception of a message. This means that the sender cannot deny the sending of the message and the receiver cannot deny the reception of the message.
- 3.13. PC card:** Physical packaging standard conforming to PCMCIA Type 1 or Type 2 implementations.
- 3.14. peer-entity authentication:** Corroboration that a communicating entity has the claimed identity.
- 3.15. resource:** A unit of functionality provided by the host for use by a device. A resource defines a set of objects exchanged between device and host by which the device uses the resource.
- 3.16. scrambling:** The process of making a signal unintelligible at the transmission point in order that it can only be received if an appropriate descrambling system is in place at the point of reception. Scrambling can be applied to audio, video or data signals
- 3.17. service:** A set of elementary streams offered to the user as a program. They are related by a common synchronization. They may be made of different data, eg. video, audio, subtitles, other data.
- 3.18. smart card:** Physical packaging standard conforming to ISO/IEC 7816 Parts 1 and 2.
- 3.19. Transport Stream:** MPEG-2 Transport Stream.

4. Acronyms and abbreviations

This clause defines the acronyms and abbreviations used in this part of ISO/IEC 16500. Annex B of ISO/IEC 16500-1 defines acronyms and abbreviations used within ISO/IEC 16500.

AC	Access Conditions
AID	Application Identifier
APDU	Application Protocol Data Unit
API	Applications Programming Interface
ASN.1	Abstract Syntax Notation One
ATM	Asynchronous Transfer Mode
ATR	Answer-to-Reset
BER	Basic Encoding Rules
bslbf	Bit String - Left Bit First
BSS	Business Support System
CA	Conditional Access
CAS	Conditional Access (Sub)System
CBC	Cipher Block Chaining
CD	Committee Draft
CP	Content Provider
CPE	Customer Premises Equipment
CW	Control Word
DE	Data Element
DES	Data Encryption Standard
DF	Dedicated File
DIR	Directory File
DO	Data Object
DSM-CC	Digital Storage Media Command and Control
DVB	Digital Video Broadcast
DVB SI	Digital Video Broadcast Service Information
ECM	Entitlement Control Message
EF	Elementary File
ES	Elementary Stream
ESP	Encapsulated Security Protocol
EMM	Entitlement Management Message
EPG	Electronic Program Guide
ETU	Elementary Time Unit
FP	Filter Program
FS	Filter Set
HFC	Hybrid Fibber Co-ax
HTML	Hypertext Markup Language

IDL	Interface Definition Language
IDO	Inter-industry Data Object
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPPV	Impulse Pay Per View
IPR	Intellectual Property Rights
Ipv4	Internet Protocol Version 4
lsb	Least Significant Bit
lsB	Least Significant Byte
MF	Master File
MJD	Modified Julian Date
MMI	Man Machine Interface
MPEG	Moving Pictures Expert Group
msb	Most Significant Bit
msB	Most Significant Byte
NIU	Network Interface Unit
NP	Network Provider
OMG	Object Management Group
PCR	Program Clock Reference
PDU	Protocol Data Unit
PID	Packet Identifier
PIN	Personal Identification Number
PIX	Proprietary Application Identifier Extension
PMT	Program Map Table
PPV	Pay Per View
PTS	Protocol Type Selection
RFU	Reserved for Future Use
RID	Registered Application Provider Identifier
RP	Record Pointer
RPC	Remote Procedure Call
SP	Service Provider
SPI	Security Parameter Index
STU	Set-Top Unit
SVB	Switched Video Broadcast
TCP	Transmission Control Protocol
TLV	Tag Length Value
TS	Transport Stream
UDP	User Datagram Protocol
URL	Uniform Resource Locator
UTC	Universal Coordinated Time
VoD	Video on Demand

5. Conventions

The style of this specification follows the *Guide for ITU-T and ISO/IEC JTC 1 cooperation. Appendix II: Rules for presentation of ITU-T | ISO/IEC common text (March 1993)*.

6. Overview of Security System

The definition of a single, common security system for DAVIC is a very difficult task. Security requirements vary from application to application, and will also depend on the characteristics of the access network, the nature and value of the content, and the business models which govern the system. Furthermore, implementations of security related hardware and software are subject to strict Government regulation, which controls the deployment and movement of such equipment. A final consideration is the very different failure modes of security systems compared to other systems: it is almost impossible to formally verify the strength of individual cryptographic algorithms, and the decreasing cost of computational power makes any given algorithm more vulnerable to attack over time.

For these reasons it is necessary to provide some scope for flexibility in the choice of security mechanisms. In particular, it is desirable to allow the coexistence of multiple cryptographic algorithms, and to provide suitable mechanisms for negotiating their use. This will allow algorithms to be replaced if they are compromised. Note that this will have an implication for interoperability, since a service which makes use of particular security algorithms will not be available unless parties involved in the service delivery have implemented compatible algorithms.

This part of ISO/IEC 16500 defines a number of security tools which must be provided within DAVIC systems implementing the security profiles. These tools include security protocols which operate across DAVIC interfaces. Where possible, these protocols have been designed to be independent of particular cryptographic algorithms. Within certain limits, there exists the flexibility to negotiate the types of algorithms which are used. This is particularly true in the case of algorithms for session confidentiality and integrity.

In the case of conditional access for broadcast services, even more scope for flexibility is required. Due to the nature of the threats in such a system, it is desirable for the design of the system to be kept secret, thereby increasing the cost of an attack on the system. For this reason, it is inevitable that conditional access systems will remain proprietary in nature.

To enhance interoperability, two security related interfaces in the STU are specified. These interfaces allow various components of the security system to be replaced, either in response to attacks, to facilitate customer mobility, or to promote interoperability. By replacing the security devices connected to these interfaces, the STU may be used with different (proprietary) conditional access systems.

The next clause defines a number of security tools. These consist of protocols for

- scrambling the S1 flow;
- scrambling algorithm negotiation and key distribution for the S1 flow;
- authentication and key exchange between the end user and the service provider over the S2 flow;
- confidentiality and integrity of the S2 and S3 flows;
- secure download of data and/or software images;
- parental control.

Clause 9 describes the STU architecture with respect to security, introducing the two conditional access interfaces CA0 and CA1. The CA0 interface is based on the Common Interface (EN 50221), with extensions defined in clause 12 to support interactive services. The CA1 interface is specified in clause 12.

7. Security Tools

This clause defines a number of security tools for DAVIC. These tools are generic in nature, and may be used by a wide variety of applications in different types of systems, to provide interoperable security. Their specification and use does not preclude the implementation of additional, application specific, security services at the application layer.

7.1 S1 Scrambling

A scrambling system is a tool which alters the digital representation of the content (video, audio, coded data) in such a way as to make it unintelligible to unauthorized users of the system, but which may be recovered by authorized users.

The requirement of the use of a scrambling system tool will typically be specified by Content Providers and/or Service Providers (including intermediate service providers), since these parties have a vested interest in the ability to limit the distribution of the content information to selected users on the DAVIC system. The scrambling system tool may be implemented by Content Providers, Service Providers, and/or Network Providers; however, content at any point in the DAVIC system can only be scrambled once.

7.1.1 Scrambled Elements

Scrambling can occur at any location within the system, but it shall be done at the MPEG 2 Transport Stream packet layer. Descrambling shall therefore also be done at the MPEG 2 Transport Stream packet layer.

Scrambling shall be applied only to payload, according to an algorithm determined by a combination of the algorithms available to the scrambling equipment and those available in the STU. The prefix fields must be left in the clear. Since services may be independently supplied to the same MPEG 2 level data multiplex then these services must be independently scrambled within the data multiplex.

The ISO/IEC 13818-1 specification specifically prohibits the encryption of transport packet headers and adaptation fields. This allows transport control and de-multiplexing/re-multiplexing without requiring decryption. This allows the Program Clock Reference (PCR), for example, to be carried and modified by multiplexers without scrambling.

TS packets shall be scrambled independently of each other. This prevents erroneous de-scrambling of information in the event of packet loss.

7.1.2 Control Word Synchronization

Control words (CW) are cryptographic keys necessary to de-scramble the scrambled payload of the MPEG-2 Transport Stream packets. The control words may be distributed or derived in a number of ways, depending on the nature of the security application in use.

For security reasons, the CWs may have a very short life (eg. in the order of seconds). In order to facilitate a real time change of the control words, it may be necessary to store two separate control words. These are known as even and odd control words.

The MPEG-2 Transport Packet header contains the transport scrambling control field to indicate the status and the parity (even or odd) of the CW used to scramble the packet payload. This field shall be used to indicate the use of the scrambling tool. The 00 state is defined in ETR 289 as signifying that the packet is unscrambled. For DAVIC compliant systems, the following additional definitions shall apply:

value	Description
00	Not scrambled
01	Reserved for future use
10	Scrambled with EVEN CW
11	Scrambled with ODD CW

7.1.2.1 Scrambling Algorithm Identification

Considering the potential legal and commercial factors which affect the implementation of scrambling algorithms in various countries, a common scrambling algorithm is not specified for DAVIC systems. As different scrambling algorithms may be used in DAVIC systems, the algorithm to be invoked must be identified.

In the case of retrieval applications, the scrambling algorithm may be negotiated on the S2 flow as described in Subclause 7.5 prior to the delivery of the service. Algorithms are identified using an ASN.1 Algorithm Identifier as defined in ISO/IEC 8824 and ISO/IEC 8825. Scrambling algorithms used in DAVIC compliant systems must be registered with an appropriate authority to enable identification.

In the case of distribution applications, the algorithm used to scramble a signal is indicated in a descriptor of the Program Map Table (PMT) during the delivery of the MPEG-2 scrambled service. This allows the STU to determine the correct descrambling algorithm, or to flag appropriate error conditions. Encoding of this descriptor is defined below. The STU may signal the scrambling algorithm and conditional access systems available in the STU as part of the STU profile.

7.1.2.2 The Scrambling Descriptor

The following descriptor may be used in either the global loop, or the local loop of the PMT to indicate the algorithm used to scramble the signal.

Syntax	No. of Bits	Encoding
scrambling_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
scrambling_code	8	uimsbf
}		

descriptor_tag: This value indicates the type of descriptor. *This value is still to be determined. A request will be made of the DVB to define this descriptor as part of the DVB specification.*

descriptor_length: This is the number of byte immediately following this field to the end of the descriptor. For this descriptor, this value is always 1.

scrambling_code: This field indicates the type of scrambling in use, coded as described below:

Code	Description
0	Reserved
1	DVB Common Scrambling
2-3F	Reserved for DVB
40-7F	Reserved for DAVIC
80-FE	User Private
FF	Reserved

7.1.2.3 Scrambling Algorithm constraints

Because channel change time must be minimized, the decoder must be able to start decryption without significant delay. Therefore it is advisable to originate the scrambler on packet boundaries since this provides for the best access point granularity. That is, such an approach provides the absolute minimum wait time for a stream entry point. Some block cipher scrambling algorithms require the packet payload size to be a multiple of the block length (8 bytes). For these algorithms, some constraints are put on the size of the adaptation field to ensure that the payload is a multiple of a block length. This constraint does not apply to combination block/stream ciphers or to stream ciphers.

7.2 S2/S3 Authentication

A three way authentication mechanism as described in ITU X.509 shall be used to provide mutual authentication between communicating entities in an S2 and S3 flow. Three way authentication eliminates the need for synchronized clocks, and allows for the establishment of session keys and negotiation of session security mechanisms as part of the authentication process. Nevertheless, timestamps appear in the protocol in order to maintain syntactic compatibility with ITU X.511. Their value should be ignored.

7.2.1 Authentication Protocol

The semantics of the authentication exchange are described below:

1. A→B: $A, \{B, T_A, N_A, IntAlgs, ConfAlgs\}^{S_A}, C_A$

The prover (A) initiates the exchange by sending a time stamp T_A and nonce N_A bound to the verifier's identity B with the prover's signature. Also included is an optional list of integrity and confidentiality algorithms ($IntAlgs$ and $ConfAlgs$) supported by the prover. The prover's identity A and certificate C_A are included in the message. The nonce N_A may be either a random number or a sequence number.

2. B→A: $B, \{A, T_B, N_B, N_A, IntAlg, \{K_{AB}^I\}^{P_A}, ConfAlg, \{K_{AB}^C\}^{P_A}\}^{S_B}, C_B$

The verifier responds with its own time stamp T_B and nonce N_B which is bound to the prover's identity A , and the prover's initial challenge N_A with the verifier's signature. Also included is an optional session key and algorithm identifier for encryption and integrity protection of the subsequent session. The algorithms must be chosen from the list provided by the prover in the previous message. These session keys are encrypted with the public key of the prover. This message authenticates the verifier to the prover.

3. $A \rightarrow B: A, \{B, T_A, N_B\}^{S_A}$

The final message is a signed response from the prover including the verifier's nonce and identity. This message completes the mutual authentication by verifying the prover's identity to the verifier.

7.2.2 Syntax of Authentication Messages

The following ASN.1 structures are used to transfer authentication messages between the two authenticating parties in an S2 or S3 information flow. Except for **AuthRequestParameter** all structures are taken from ITU X.511. The way in which these data structures are carried in the S2 and S3 information flows is defined below. The **SPIValue** of **AuthRequestParameter** is used to identify the key(s) which may be exchanged during the authentication to the IP security tools defined in Subclause 7.3.

```

AuthRequestParameter ::= SET {
    argument                [0]    BindArgument,
    SPIValue                 [1]    BIT STRING OPTIONAL}

BindArgument ::= SET {
    credentials              [0]    Credentials OPTIONAL,
    versions                 [1]    Versions DEFAULT {v3} }

Credentials ::= CHOICE {
    simple                   [0]    SimpleCredentials,
    strong                   [1]    StrongCredentials,
    externalProcedure        [2]    EXTERNAL,
    spkm                     [3]    SpkmCredentials }

SimpleCredentials ::= SEQUENCE {
    name                     [0]    DistinguishedName,
    validity                 [1]    SET {
    time1                    [0]    UTCTime OPTIONAL,
    time2                    [1]    UTCTime OPTIONAL,
    random1                  [2]    BIT STRING OPTIONAL,
    random2                  [3]    BIT STRING OPTIONAL } OPTIONAL,
    password                 [2]    CHOICE {
        unprotected          OCTET STRING,
        protected            SIGNATURE { OCTET STRING } OPTIONAL }

StrongCredentials ::= SET {
    certification-path       [0]    CertificationPath OPTIONAL,
    bind-token               [1]    Token,
    name                     [2]    DistinguishedName OPTIONAL }

Token ::= SIGNED { SEQUENCE {
    algorithm                [0]    AlgorithmIdentifier
    name                     [1]    DistinguishedName,
    time                     [2]    UTCTime,
    random                   [3]    BIT STRING,
    response                 [4]    BIT STRING OPTIONAL,
    bindIntAlgorithm         [5]    SEQUENCE OF AlgorithmIdentifier OPTIONAL,
    bindIntKeyInfo           [6]    BindKeyInfo OPTIONAL,
    bindConfAlgorithm        [7]    SEQUENCE OF AlgorithmIdentifier OPTIONAL,
    bindConfKeyInfo          [8]    BindKeyInfo OPTIONAL,
    dirqop                   [9]    OBJECT IDENTIFIER OPTIONAL,
    attributeCertificate      [10]   AttributeCertificate OPTIONAL,
    clearanceCertPath        [11]   ClearanceCertPath OPTIONAL } }

Versions ::= BIT STRING { v1(0), v2(1), v3(2) }

```

BindKeyInfo ::= ENCRYPTED { BIT STRING }

The above syntax does not preclude the use of additional private conventions consistent with the syntax which may extend the capabilities of the security services specified herein. In particular, it is to be noted that:

- Use of Distinguished Names may be supplemented, or even supplanted, with use of the Alternative Name attribute in the X.509 certificate. By this means, for example, Internet domain names may be supplied. Also such names are covered by the same certificate as the Distinguished Name. If a Distinguished Name is not supplied, but an Alternative Name is supplied, it is the responsibility of the Directory to establish the mapping between the Alternative Name and the Distinguished Name.
- Clone detection may be based on the use of a monotonically increasing sequence number. Such a sequence number may be incorporated into the Random field provided by the prover. The existence of such a structure is a private arrangement between the prover and the authentication server. Participating servers have the responsibility to implement the detection of clones based on violations of the monotonic increase of the sequence.

7.2.3 Key Use

A single key pair is used for both signing authentication tokens and for encrypting keys in the authentication protocol. This requires only a single certificate chain to be transferred in either direction. Note that long term digital signatures (eg. as used for non-repudiation) may well use a different key pair. This separation of keys for digital signatures on one hand, and authentication and key exchange on the other, is encouraged.

7.2.4 Certificates

Certificates according to ITU-T X.509 Version 3 shall be used.

Certificate revocation is application specific. Certification revocation lists may be obtained by a Directory lookup using a directory access protocol, eg. as described in ITU-T X.500.

7.2.5 Integration with Protocols for S2

Within the S2 flow, the end user will always adopt the role of prover in the authentication exchange.

Authentication exchanges are initiated at the discretion of the user, although the server may refuse to interact with the user without such an authentication. This is signaled to the user by issuing a **NO_AUTH** exception.

The authentication messages described above are carried in the **AuthRequest_T** structure in DSM-CC user to user messages. In particular, the first message will be included in the **authInfo** parameter in the **ServiceContextList** of an attempted operation invocation on the server object. The response from the server is carried in the **authData** parameter of a **NO_AUTH** exception. The final response from the client is carried in the **ServiceContextList** of a repeated attempt. Any operation which returns the **NO_AUTH** exception can be used to perform the authentication exchange. For example, the authentication exchange may occur when the client attempts a **directory.open** operation on the server.

The **AuthRequest_T** type is defined as opaque in DSM-CC. DAVIC compliant systems shall make use of the authentication tokens defined in X.511. A BER encoding of the **AuthRequestParameter** ASN.1 structure shall be used for the **authInfo** and **authData** parameters.

7.3 S2/S3 Confidentiality and Integrity

Confidentiality and integrity functions must be performed below DSM-CC because preservation of common bit representations cannot be guaranteed through the RPC operations performed by DSM-CC. Confidentiality shall be provided by encrypting data at the IP level. The IP Encapsulating Security Payload (ESP) [26] shall be used in Transport mode. The IP Authentication Header [25] shall be used to provide integrity and data origin authentication, if required.

7.3.1 Negotiation of Confidentiality and Integrity Algorithms

Confidentiality and integrity algorithms on S2 are negotiated during the authentication exchange described above. This results in the choice of an encryption algorithm and an integrity algorithm, together with session keys for each. In RFC 1825, support of DES CBC mode encryption is recommended. At present, DAVIC does not specify default algorithms.

7.3.2 Definition of the SPI value

The encryption algorithm and integrity algorithm, together with the source and destination addresses, define two uni-directional security associations between the two end-points. These are referenced using a security parameter index (SPI) as described in [24]. The value used for the SPI for a channel is given by **SPIValue** in the authentication exchange. The **SPIValue** shall be defined by the destination party as recommend in [24]. It is a local matter which values are used. The **SPIValue** in the **AuthRequestParameter** of the second message of the authentication exchange shall be used to refer to the security association from the STU to the server. The **SPIValue** in the **AuthRequestParameter** of the final message of the authentication exchange shall be used to refer to the security association from the server to the STU. The **SPIValue** of the **AuthRequestParameter** of the first message may be ignored.

7.3.3 Signaling

The application program shall signal to the IP layer protocol entity if data has to be confidentiality or integrity protected. This signaling is a local matter of the application which is not defined here. The use of confidentiality and/or integrity services is signaled to the receiving party in the IP security protocol headers.

7.3.4 Replay Protection on S2

The integrity mechanisms of IP security protect the PDU which means that an attacker cannot alter the content of TCP layer information since it is protected by an integrity check value. Therefore an already used TCP sequence number will be repeated. This will be detected by the TCP layer.

Note: Two problems are associated with this pragmatic solution:

- *TCP will not report a detection of repeated sequence number as a security error.*
- *TCP sequence numbers may wrap around. An attacker could then insert a packet with a previously used sequence number. This cannot be detected by TCP.*

Proposals for addressing replay protection within IP are being discussed within the IETF. Any solutions developed will be considered by DAVIC in a future revision.

7.4 S2 Digital Signatures

In some applications it is necessary to sign data which is transferred across the S2 flow. This may be for data origin authentication and/or non-repudiation. In general this must be done at the application layer, since the unit of data being signed has significance only in the context of the application protocol being used. Consequently DAVIC does not specify any mechanism to carry out this function.

7.5 DSM-CC Commands for S1 Security Management

Within the retrieval profile, the security services provided on the S1 flow may be negotiated and managed over the S2 flow. This subclause defines DSM-CC User to User commands to configure a security association between the STU and the Server for the S1 flow. This includes the negotiation of scrambling algorithms and the establishment and updating of cryptographic keys. These commands are defined within a separate interface called **S1Security**.

7.5.1 Security Association Configuration

The **assocConfig** operation is used to establish the security association for the S1 flow. This includes negotiation of the scrambling algorithm. Also included is either the transfer of the cryptographic keys needed to descramble

the S1 flow, or an indication of the CA system which will be used to deliver the keys. The **assocConfig** operation forms part of the **S1Security** interface.

7.5.1.1 IDL Syntax

The following syntax shall be used for the **assocConfig** operation:

```
module DAVIC_Security {
    const AccessRole assocConfig_ACR = READER;
    typedef sequence<octet> AlgorithmID;
    typedef sequence<AlgorithmID> AlgorithmIDList;
    typedef sequence<octet> Key;
    typedef sequence<octet> CASystemId;
    exception ALGORITHM_UNAVAILABLE;
    exception UNKNOWN_SYSTEM;
    interface S1Security {
        void assocConfig (
            in AlgorithmIDList supportedAlg,
            in any systemCharacteristics,
            out AlgorithmId s1Alg,
            out Key oddKey,
            out Key evenKey,
            out u_long keyChangeInterval,
            out CASystemId caSystem)
            raises (NO_AUTH, ALGORITHM_UNAVAILABLE, UNKNOWN_SYSTEM);
    };
};
```

7.5.1.2 Semantics

The **S1Security assocConfig()** operation establishes a security association between the STU and the server over the S1 flow. This includes the negotiation of a scrambling algorithm, and the transfer of a cryptographic key from the server to the STU. A lifetime for the key is also defined. In some environments it may be desirable to deliver the cryptographic keys via the CA system. In such a case the server includes the CA system identifier in the return parameters.

The STU lists the scrambling algorithms which it supports via the **supportedAlg** parameter. Algorithms are identified by ISO object identifiers as defined in ITU X.509. The server chooses a single algorithm **s1Alg** which will be used to protect the S1 flow, or raises an **ALGORITHM_UNAVAILABLE** exception. The server returns two keys (**oddKey** and **evenKey**) for use with the scrambling algorithm. The keys are used as the odd and even control words in the scrambling system. The use of two keys facilitates the synchronization of key changes, which are signaled in the MPEG stream.

The lifetime of the keys is returned in the **keyChangeInterval** parameter. If the key change interval is non zero, the STU shall perform a **S1Security getNewKey()** operation within **keyChangeInterval** milliseconds of a key change signaled in the MPEG stream. The key change interval is determined by the server with reference to the **systemCharacteristics** provided by the STU. The **systemCharacteristics** are of the **OMG any** type. The coding of the system characteristics is not defined, but can be used to signal return path delay for example. Unless a pre-determined common encoding exists, the value **null** should be provided by the STU.

Note: While very short key change intervals are supported by this specification, careful consideration must be given to potential network latencies, as well as processing speeds, when determining a suitable value. In general, a minimum key change interval will be of the order of a few seconds, if not minutes or hours. Note that in broadcast conditional access systems the key changes may occur much more frequently.

In some environments it may be necessary to deliver the keys via the CA system. In such a case the server returns a two octet CA system identifier, coded as described in ETR 289. Otherwise a null sequence should be returned.

Note: These messages shall be protected by confidentiality services on the S2 flow. This will be indicated by the ALL SECURE flag in DSM-CC.

7.5.1.3 Privileges Required

The invoking object requires **READER** privileges.

7.5.1.4 Parameters

The following table summarizes the parameters of the **assocConfig** operation:

Type/Variable	Direction	Description
AlgorithmIDList supportedAlg	input	A list of identifiers of scrambling algorithms supported by the STU, ordered by preference (eg. due to speed).
any systemCharacteristics	input	A description of some system characteristics eg. delays in access network or return path. Coding of this parameter is not defined.
AlgorithmID s1Alg	output	The identifier of the scrambling algorithm which will be used to protect the S1 flow.
Key oddKey	output	Odd parity key used for descrambling S1 flow.
Key evenKey	output	Even parity key used for descrambling S1 flow.
u_long keyChangeInterval	output	The minimum time in milliseconds between key changes.
CASystemId caSystem	output	The CA System identifier of the CA system used to deliver cryptographic keys (or null if the keys are to be transferred directly over S2 in the output parameters).

7.5.2 Key Retrieval

If more than one scrambling key is to be delivered over the S2 flow, the **getNewKey** operation shall be used to retrieve subsequent keys.

7.5.2.1 IDL Syntax

The following syntax shall be used for the **getNewKey** operation:

```

module DAVIC_Security {
    const AccessRole assocConfig_ACR = READER;
    typedef sequence<octet> Key;
    interface S1Security {
        void getNewKey (
            out Key key,
            out boolean keyParity,
            out u_long keyChangeInterval)
            raises (NO_AUTH);
    };
};

```

7.5.2.2 Semantics

The **S1Security getNewKey()** operation is used by the STU to retrieve new keys for descrambling the S1 flow. New keys are needed at intervals indicated by the **keyChangeInterval** returned by the previous **getNewKey()** or **assocConfig()** operations. The new key is returned in the **key** parameter, while the parity of this key (odd or even) is indicated by **keyParity**.

Note: These messages must be protected by confidentiality services on the S2 flow.

7.5.2.3 Privileges Required

The invoking object requires **READER** privileges.

7.5.2.4 Parameters

The following table summarizes the parameters of the **getNewKey** operation:

Type/Variable	Direction	Description
Key key	output	New key used for descrambling S1 flow.
boolean keyParity	output	Parity of the above key 0=even; 1=odd.
u_long keyChangeInterval	output	The minimum time in milliseconds until the next key change.

7.6 Secure Download

This subclause defines a mechanism for protecting software and data downloads to the STU. The integrity, source and freshness of the data are protected by this mechanism; however, confidentiality of the data is not provided.

Secure download is implemented by including a security module in the downloaded data image. The **moduleID** of the security module shall be 0xFFFE. No other type of module in a DAVIC compliant system may use the 0xFFFE **moduleID**. The **moduleVersion** for security modules conforming to this specification should be 0x01.

7.6.1 Format of Security DownloadInfoRequest

The STU signals its requirement for a secure download in the **privateDataBytes** in the **DownloadInfoRequest** message. DAVIC systems implementing the security mechanisms defined in this part of ISO/IEC 16500 shall make use of the following format for the **privateDataBytes** (Syntax according to ISO/IEC 13818-6):

Syntax	No. of Bits	Encoding
davicPrivateSecurityData() {		
securityModuleVersion	8	uimbsf
securityParametersLength	8	uimbsf
if (securityModuleVersion==1){		
challenge	32	uimbsf
hashingMechanism	8	uimbsf
signatureMechanism	8	uimbsf
certificationPublicKeyHash	32	uimbsf
}		
for (i=0;i<N;i++) {		
privateDataBytes		
}		
}		

The **securityModuleVersion** indicates which version of the security module definitions the STU supports. A value of 0 indicates it does not support or does not require security. The security module defined in this specification should be indicated by the value 0x01.

The **securityParametersLength** indicates the number of bytes devoted to security parameters in this message. For version 1 defined in this specification the value should be 10 (0x0A).

The **challenge** is a four byte random challenge used to ensure the freshness of the authentication mechanism.

The **hashingMechanism** indicates the hashing mechanism supported by the STU or a security device. Encoding of this field is proprietary, and will be defined by the STU or security device manufacturer.

The **signatureMechanism** indicates the signature mechanism supported by the STU or security device. Encoding of this field is proprietary.

Note: The STU shall send information for identifying the smart card and/or security device manufacturer within the compatibilities or **privateData**. This will allow the server to link the mechanism identifier to a manufacturer and thereby choose the correct mechanism.

The **certificationPublicKeyHash** is the four least significant bytes of the hash of the public key the STU will use to check the signature, or, if a certificate chain is provided, to check the first certificate in the chain. Hashing is according to the hash algorithm indicated by the **signatureMechanism**. It is included as a check that secure download is possible. If not, the download may be aborted, since the signature will fail.

The remainder of the **PrivateData** field is proprietary. The number of proprietary bytes may be computed from the total length of the **PrivateData** field, given in the **DownloadInfoRequest** message.

7.6.2 Format of Security DownloadInfoResponse

The format of DownloadInfoResponse messages in DAVIC is generally unchanged from the definition in DSM-CC; however, where a security module is present, the **ModuleInfo** field corresponding to the security module shall conform to the following format:

Syntax	No. of Bits	Encoding
ModuleInfo() { timeStampLength for(i=0;i<timeStampLength;i++) { UTCTimeBytes } for (i=0;i<N;i++) { privateDataBytes } }	8	uimbsf

The **UTCTimeBytes** represent the current time at which the **DownloadInfoResponse** message was generated. It is coded according to the definition of UTC time in ISO/IEC 8824.

The remainder of the **ModuleInfo** field is proprietary. The number of proprietary bytes may be computed from the total length of the **ModuleInfo** field, given in the **DownloadInfoResponse** message.

7.6.3 Format of Security Module

The security module shall conform to the following format:

Syntax	No. of Bits	Encoding
DownloadSecurityModule() { signatureLength for(i=0;i<signatureLength;i++) { signatureBytes } certificateChainLength for (i=0;i<certificateChainLength;i++) { certificateChainBytes } }	16 16	uimbsf uimbsf

The **signatureBytes** are obtained by calculating a signature on the concatenation of bit strings used to represent the **DownloadInfoRequest** message, the **DownloadInfoResponse** message and the entire data image excluding the security module. This data image is formed by concatenating the **blockDataBytes** of **DownloadDataBlock** messages, to form individual modules, and ordering these modules as specified in the **DownloadInfoResponse** message. It is recommended that the padding mechanism for the signature generation should be according to ISO 9976-2 (currently CD); however, the actual definition of the padding mechanism will form part of the definition of the signature mechanism used. Note that the headers of **DownloadDataBlock** messages, and the contents of **DownloadDataResponse** messages are not included in the calculation of the signature. In a pure broadcast environment the signature will be calculated over the **DownloadInfoResponse** message and the downloaded data only.

The **certificateChain** is optional. If present it contains a chain of certificates which can be used by the STU to verify the public key of the server. This key would then be used to verify the signature. The certificates are X.509 version 3. The first certificate in the chain must be issued by the certification authority whose key is available at the STU (the hash of which is sent in the **DownloadInfoRequest** message). In this version of the download protocol, it is assumed that there will be at most a single certificate in the chain. If there is no certificate chain, the certification authority public key is used to verify the signature directly.

Note: The certification authority used to verify download images may be different to the general purpose certification authority used for authentication etc, and may take an active role in verifying the correctness/suitability of the code.

7.6.4 Download Reasons in DownloadDataResponse

The **DownloadDataResponse** to the **DownloadDataBlock** of the last block of the secure download, ie. the last block of the security module shall indicate the result of the verification of the signature. The **downloadReason** concerning this are defined in the following table.

downloadReason	Definition
0xF0	Signature OK
0xF1	Signature verification failed
0xF2	Failure due to expiration of certificate
0xF3	Other failure

7.7 Parental Control

7.7.1 Introduction

Parental control enables the user to block certain contents. Parental Control is not a mandatory feature for DAVIC. Supporting parental control may depend on legal issues or the decisions of the provider/manufacturers. However, if parental control is supported the necessary information must be transmitted in an interoperable manner.

DAVIC specifies a rating based system. It is assumed that a specific rating (eg. suitable for children of age 10 years or older) is associated with all content (eg. a movie). However, DAVIC is not responsible for finding the appropriate rating for specific content.

In the context of Parental Control, DAVIC distinguishes between retrieval services, such as VoD, and distribution services such as Pay-per-channel or Switched Video Broadcasting (SVB). In both cases it is necessary that the information about the user on which the access control is based (like this user is allowed to watch all content up to rating xyz) is input to some CPE device (like the STU). This information is called *user access rights* in the following. DAVIC does not specify the user interface for entering the user access rights to the CPE device. DAVIC does not mandate the use of cryptographic methods for enforcing parental control, ie. parental control may or may not be performed by the security subsystem.

7.7.2 Retrieval Services

For retrieval services like VoD the most efficient way for parental control is to block the content at the source, ie. to enforce parental control by the Service Provider. The user access rights must be available to the Service Provider which then can decide if the user is allowed to order a certain content. This information may be stored by the Service Provider in the user profile.

7.7.3 Distribution Services

For distribution services it is assumed that the rating is not necessarily the same for all contents of the channel, ie. the rating shall be content based and not channel based.

For distribution services the network provider usually does not know both pieces of information necessary for parental control, the content rating and the user access rights. Even in the SVB scenario the broadcast control unit and the replication unit in the access network do not know the content rating because the rating is transmitted with the content in S1 and the access network is only a pipe for S1. The broadcast provider usually does not have the ability to block content for certain users.

Therefore, for distribution services (like Pay-per-channel or SVB) DAVIC specifies that if parental control is supported it is supported by the STU. If the content rating is sent it shall be part of the Service Information (SI) data. The content rating shall be based on ETS 300 468, which specifies the transmission of content rating in the SI data of the MPEG transport stream. However, it is recognized that the DVB SI specification does not meet the requirements of content rating schemes of some non-European countries. Therefore, some extensions will be necessary.

The STU reads the rating of the content and compares it with the user access rights. If the user is not allowed to receive the content, the content is blocked. The implementation is left to the STU manufacturer.

In a situation where billing depends on the user behavior (like Pay-per-view or SVB) it must be guaranteed that blocked content is not billed. It is the responsibility of the STU to either not request for a channel with inappropriate rating or to switch (eg. to a default channel) if the program is already received by the STU.

8. Flows and Protocol Stacks

This clause defines the information flows and protocol stacks associated with the security tools specified in clause 7.

8.1 Sample Flow: Scrambled Video on Demand (Informative)

This subclause shows the information flows between different entities associated with the delivery of a scrambled video on demand service. The authentication exchange can take place on the S2 flow at any point during the navigation process. It will be triggered by a no authentication exception raised by the server. The message flows are summarised in Figure 1.

Figure 1

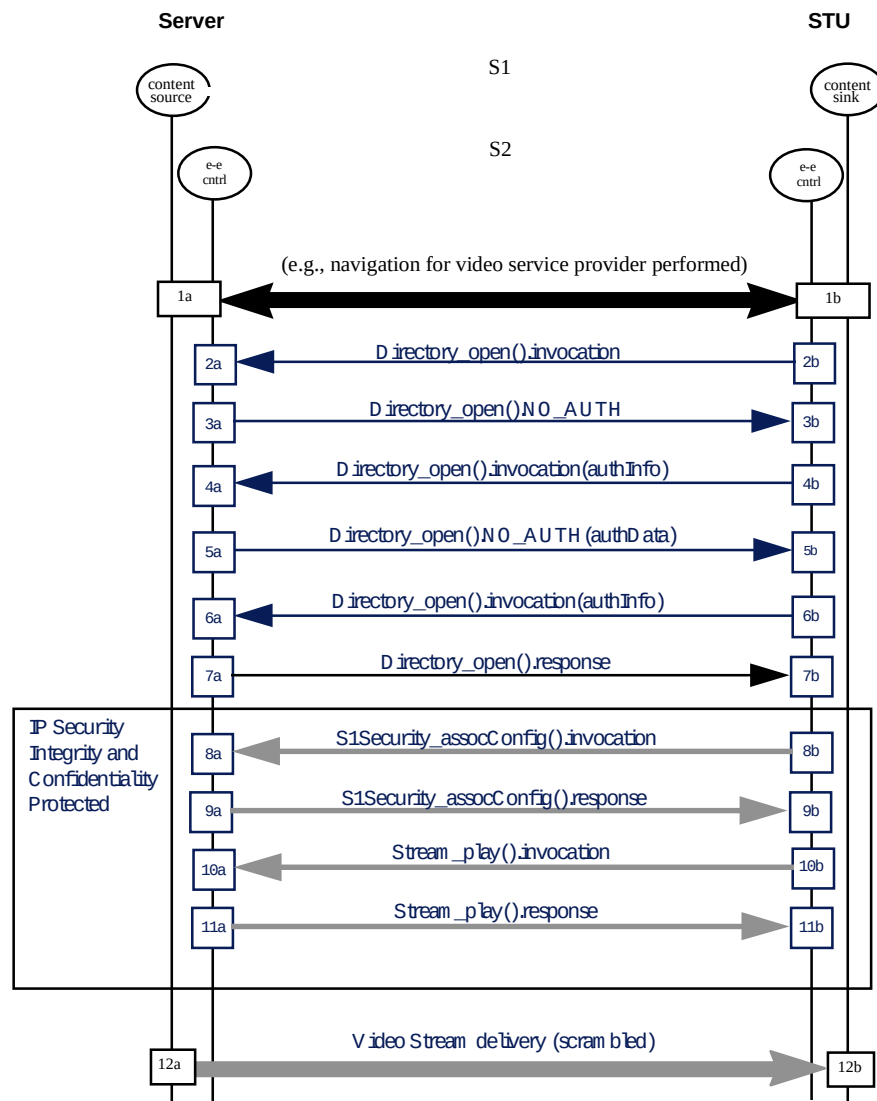


Figure 1. — Information flows for scrambled VoD scenario

1a-1b	Navigation through server directory using DSM-CC - no security services have been invoked at this stage.
2a-2b	Directory open operation invoked by STU, referring to object requiring authentication.
3a-3b	Server returns NO_AUTH exception with empty authData parameter
4a-4b	STU repeats the Directory open operation, this time carrying security authenticate information in the service context list of the operation. AuthInfo parameter is BER encoding of following: <pre> AuthRequestParameter ::= SET { argument [0] BindArgument, SPIValue [1] BIT STRING } BindArgument ::= SET { credentials [0] Credentials, versions [1] Versions} Credentials ::= CHOICE { strong [1] StrongCredentials} StrongCredentials ::= SET { certification-path [0] CertificationPath, bind-token [1] Token } Token ::= SIGNED { SEQUENCE { algorithm [0] AlgorithmIdentifier name [1] DistinguishedName, time [2] UTCTime, random [3] BIT STRING, bindIntAlgorithm [5] SEQUENCE OF AlgorithmIdentifier, bindConfAlgorithm [7] SEQUENCE OF AlgorithmIdentifier }} </pre>
5a-5b	Server responds with NO_AUTH exception. The authData parameter of this exception contains the following ASN.1 object encoded in BER: <pre> AuthRequestParameter ::= SET { argument [0] BindArgument, SPIValue [1] BIT STRING} BindArgument ::= SET { credentials [0] Credentials, versions [1] Versions } Credentials ::= CHOICE { strong [1] StrongCredentials } StrongCredentials ::= SET { certification-path [0] CertificationPath, bind-token [1] Token } Token ::= SIGNED { SEQUENCE { algorithm [0] AlgorithmIdentifier name [1] DistinguishedName, time [2] UTCTime, random [3] BIT STRING, response [4] BIT STRING, bindIntAlgorithm [5] SEQUENCE OF AlgorithmIdentifier, bindIntKeyInfo [6] BindKeyInfo, bindConfAlgorithm [7] SEQUENCE OF AlgorithmIdentifier, bindConfKeyInfo [8] BindKeyInfo } } </pre>

6a-6b	The STU re-issues the Directory open invocation, again including the security authenticate parameter authInfo in the service context list. The following ASN.1 type is BER encoded to form authInfo: <pre> AuthRequestParameter ::= SET { argument [0] BindArgument, SPIValue [1] BIT STRING} BindArgument ::= SET { credentials [0] Credentials, versions [1] Versions } Credentials ::= CHOICE { strong [1] StrongCredentials } StrongCredentials ::= SET { certification-path [0] CertificationPath, bind-token [1] Token } Token ::= SIGNED { SEQUENCE { algorithm [0] AlgorithmIdentifier name [1] DistinguishedName, time [2] UTCTime, response [4] BIT STRING } } </pre>
7a-7b	The Directory open operation succeeds. At this stage both ends of the connection have complete information for the S2 security association, and can invoke the security services negotiated for S2. In general this will happen on the first operation to the new object (in this case a stream object providing a movie on demand).
8a-8b	STU invokes assocConfig() operation to negotiate the security services on S1. This operation informs the server of the scrambling algorithms supported by the STU.
9a-9b	The server responds indicating the scrambling algorithm it will use to protect the content, and informing the STU of the keys it will use to scramble the content. Note that this message must be protected by S2 confidentiality services.
10a-10b	The STU invokes a Stream play() operation to begin playing the movie.
11a-11b	Response from play operation.
12a-12b	The scrambled movie is sent on the S1 flow.

9. Security Interfaces

There are many advantages in providing detachable security device(s) on the STU. Such an approach allows security functionality to be replaced in response to successful attacks, or changing functional requirements. While it is possible to implement changes in the STU functionality which is not related to security by means of software downloads, this approach is not acceptable for security related functions. These will require some form of trusted (tamper-resistant) hardware.

An additional advantage of a detachable device is that it allows the STU to be generic, in that it contains no customer specific data. This allows customer mobility (in that a customer may make use of other STUs - eg. when travelling, or visiting friends), easy replacement of the STU, and reuse of an STU (eg. if a customer moves or ceases to subscribe to a service). This solution permits open competition while maximizing the commonality of the STU.

Two interfaces are defined to different parts of the security system. These are CA0 and CA1 as shown in Figure 2. A DAVIC STB or STU that is compliant with the security tools must offer at least one of these interfaces. The first interface CA0 is to all security and conditional access functions, including the high speed descrambling functions, while CA1 is an interface to a low cost, tamper resistant device used for low speed cryptographic processing. This cryptographic processing function is implemented in a smart card. Where a CA1 interface is implemented, the filtering function described in this specification must also be implemented.

However, if the DAVIC STB utilizes the DAVIC Internal A0 tool, it may support the physical CA1 smart card reader component on the NIU rather than in the STU. In this case all other security elements reside in the STU.

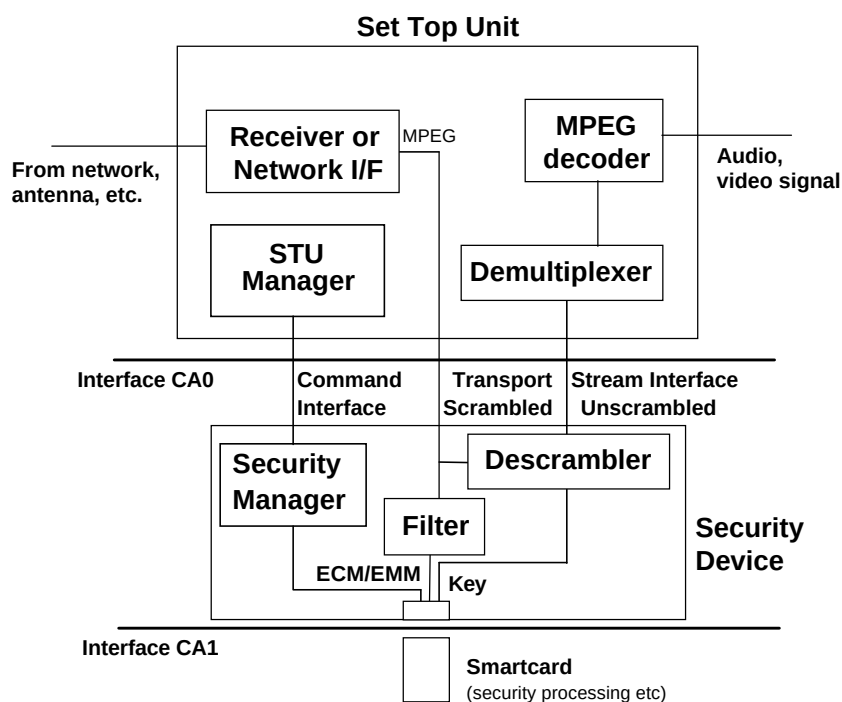


Figure 2. — Security interfaces in the STU

10. Security Interface CA0

10.1 Introduction

The CA0 interface is between a PC card based security device and a host where stream decryption, conditional access, security and more generally defined proprietary functions may be implemented in the PC card. This solution also allows service providers to use security devices containing solutions from different suppliers in the same network thus increasing their choice and anti-piracy options. The interface to be used is based on the Common Interface standard CENELEC EN 50221 and CENELEC R 206-001.

The specification defines a set of functions, partitioned into resources. Not all of the functions required are defined in CENELEC EN 50 221. Necessary additional resources to support DAVIC systems are defined in clause 12. DAVIC has identified these to DVB together with requirements for other additions for incorporation in future revisions of the standard.

Annex B.3 of CENELEC EN 50 221 specifies an optional smart card reader resource class dedicated to electronic commerce applications. The resource is intended to support a smart card reader in a host or on a device used for short term sessions, such as a Bank Card, teleshopping or pay-per-view transactions. This resource is intended to offer users alternative payment options which use smart cards issued by financial services organizations. This Annex of CENELEC EN 50 221 is also adopted, as an optional resource.

10.2 Additional DAVIC Requirements for CA0

10.2.1 Host to Security Device Authentication

This has been identified as needed to support clone detection, and also to associate a PC card with a particular host. These requirements will be refined in a future version of the specification.

10.2.2 Security Services

S2 security is supported by resources defined in clause 12. However applications running on the STU will also require security services, for example to support payment protocols. These services will also be provided by the

PC card, but the exact nature of the Application Programming Interface to be provided will be defined in a future version of the specification.

11. Profiles/Contours

DAVIC 1.3.1 defines two security profiles:

1. a secure distribution services profile which supports
 - the interfaces CA0/CA1,
 - the secure download
 - the STU authentication of the smart card,
2. a secure retrieval services profile which supports
 - the interfaces CA0/CA1,
 - the secure download
 - the S1, S2, S3 security tools,
 - the S1 negotiation.

Both profiles are optional.

12. Security Interface CA1

12.1 Introduction and scope

12.1.1 Introduction

The following text specifies the CA1 interface. The approach taken emphasizes optimization of the implementation cost and user-friendliness. The governing principle in the design is to provide sufficient detail to promote interoperability of components without imposing unnecessary limitations on implementations. The CA1 interface is cost effective and has benefits for both the replacement/change of conditional access systems and for providing the means to allow the STU to be user independent. The CA1 interface is based on proven technology. Using existing standards promotes openness and takes advantage of economies of scale due to previous or concurrent use in other industries.

The specification is compliant to the ISO/IEC 7816 standard for smart cards (parts 1-6). Elements of ISO/IEC 7816 parts 1-6 are used, but the usage of especially parts 3 and 4 are carefully limited to selected, minimum functions so as not to burden the implementation with excessive overhead. The solution is network-independent and is able to co-exist with other security interface definitions. By emphasizing the use of existing standards, the openness of the CA1 interface is enhanced.

12.1.2 CA1 reference model

The architecture of the CA1 interface is shown in Figure 3. In this figure, the incoming transport stream (TSin) is parsed and the appropriate streams are sent to the **descrambling unit** and the **message filter**. The specification assumes that data are descrambled in MPEG-2 Transport Stream [ISO/IEC 13818-1] level. The message filter further parses the input stream to extract eg. CA messages. These are then sent for processing by the smart card. The message filter is initialized by the smart card for filtering. This is shown by the (logical) relation between the smart card and the message filter in Figure 3.

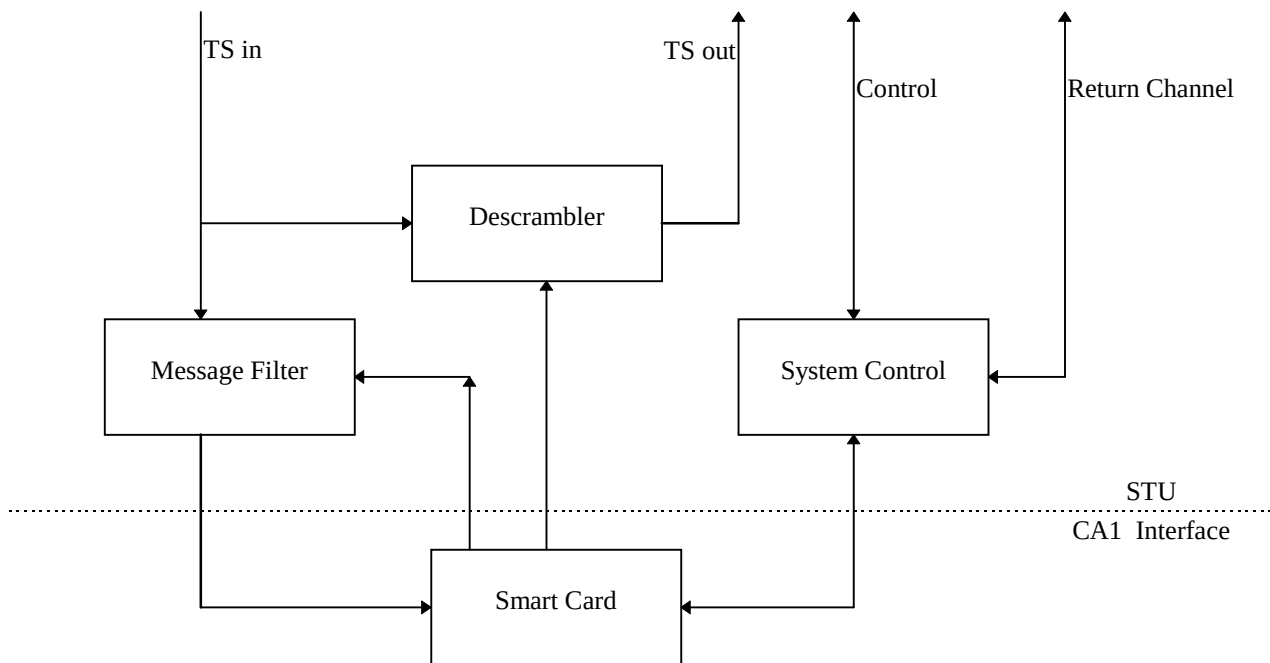


Figure 3. — CA1 Reference Model

The **smart card** processes CA messages that belong to its CA system and uses the results of ECM processing to initialize the **descrambler unit**. This is shown in the figure by the (logical) relation between the smart card and the descrambler. Data exchange may include CA system identification. The **smart card** also supports interaction with external entities by interfacing with the **system control** unit. This interaction can be both with a local user (eg. password entry) and/or with a network or servers (eg. service ordering, authentication) over a **return channel**. This specification is not restricted to a particular implementation of the return channel. For example, the return channel may flow over a different physical path than the forward channel data.

After being initialized by the **smart card**, the **descrambler** uses the CWs to process the incoming MPEG-2 stream into the clear. The stream is then passed as TSout for further processing by the STU or other units.

In this architecture, there is a physical separation of security functions. The **smart card** is distinguished from the **descrambler/message filter** at all layers.

The architecture for the CA1 interface includes a physical layer (based on ISO/IEC 7816-1,2) and also contains electrical, protocol and data structure specifications (based on ISO/IEC 7816-3,4). ISO/IEC 7816-5 and ISO/IEC 7816-6 are also used. Based on these standards this CA1 specification provides specific usage methods/structures and sets limits and constraints that promote cost-effective implementation without significant reduction of functionality.

This specification is based on the following assumptions about DAVIC conditional access implementations:

- Each CA system is identified by its CA_system_id (as defined in ETR 289)
- One STU may function with one or more CA system(s)
- One or more service providers may use the same security device.
- The CA1 interface is delivery network independent

The different elements in the CA1 reference model have the following functionality:

TS in: This is the full transport stream input to the STU.

TS out: This is the full transport stream output to the back-end of the STU. This stream may include descrambled services.

Control: This represents the user interface, and any control the user may apply to the STU, such as channel changes and setting maturity ratings.

System Control: This represents the STU central control microprocessor. This controller can communicate with the smart card for user interfaces, and setting services.

Return Channel: This is the DAVIC return channel which may be used for authentication, or digital interactive services.

CA1 Interface: This is an ISO 7816 compliant interface used to communicate with the smart card.

Descrambler: this is the common descrambler unit used to descramble services using control words from the smart card.

Message Filter: this is a combination of hardware and/or software filters used to reduce the ECM or EMM data rate from the high speed transport stream to a rate suitable for the smart card interface. The filter program (state machine) is read from the smart card and loaded into the message filter.

Smart Card: This is the actual integrated circuit card defined by ISO. This intelligent card contains the conditional access system used to control the services available to the STU. This card also handles DAVIC authentication, digital signatures, and may contain some high-level man-machine interfaces.

12.2 Notation

In all representations of data objects the left most bit represents the most significant bit (msb) and the right most bit represents the least significant bit (lsb). The same holds for bytes: the left most or first byte represents the most significant byte (msB), while the right most or last byte represents the least significant byte (lsB). The Figure 4 illustrates this.

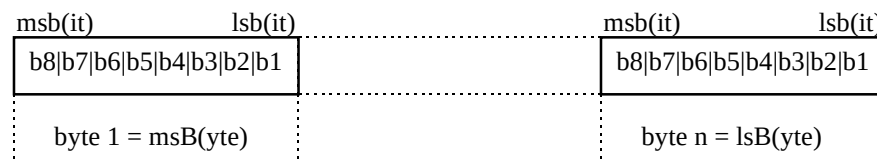


Figure 4. — Representation of data objects.

12.3 Physical characteristics of the CA1 security device

The physical characteristics of the CA1 security device shall satisfy the specifications ISO/IEC 7816-1 and ISO/IEC 7816-2.

12.4 Electronic signals and transmission protocols on the CA1 interface

12.4.1 Additions and restrictions to ISO/IEC 7816-3

The electronic signals and transmission protocols on the CA1 interface shall satisfy ISO/IEC 7816-3, but with the following additions and restrictions:

1. The STU shall be able to support raw data rates of communication of up to 107.563 kilobit per second, between the security device and the STU.
2. The smart card shall respond with the asynchronous answer-to-reset (ATR) as described in ISO/IEC 7816-3.
3. The ATR shall be transmitted in half-duplex mode.
4. The character frame for the ATR shall be as in clause 6.1.2 of ISO/IEC 7816-3. That is, a character shall consist of 12 bits, which are: a start bit, 8 bits of information, a parity bit and 2 stop bits. This is only a repetition of ISO/IEC 7816-3. It does not give any restriction to the standard.
5. The ATR shall specify the T=0 protocol. The STU shall support T=0.
6. There is no requirement for the STU nor the smart card to support Vpp.
7. The STU must support the PTS protocol, but it is not mandatory for the smart card to support it.
8. During ATR the clock frequency shall be in the range defined by ISO.

- 9. During normal operation, the clock frequency shall be at least 5Mhz.
- 10. The STU shall execute an external reset before using the card.
- 11. Support of the retry mode is not mandatory in the STU.

12.4.2 Logical channels

It is mandatory for the STU to support at least two logical channels as described in ISO/IEC 7816-4.

12.5 CA message format and filter specification

12.5.1 CA message mechanism and format

The CA message mechanism and format shall be according to ETR 289. Figure 5 shows the CA message syntax from this specification.

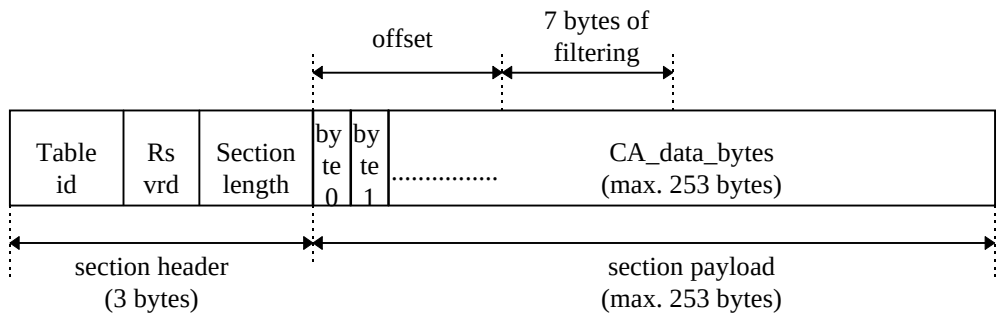


Figure 5. — CA message syntax according to DVB.

Table_id: this is an 8-bit field for message type identification.

Rsvrd: this is a 4-bit field which consist of a **section_syntax_indicator** (1 bit) field, a **DVB_reserved** (1 bit) field, and an **ISO_reserved** (2 bits) field.

Section_length: this is a 12-bit field, that specifies the number of bytes that follow the section_length field up to the end of the section.

CA_data_byte: this is an 8-bit field which carries private CA information.

12.5.2 Filtering of a CA message

Filters are used by a STU to extract required messages for an incoming MPEG-TS stream. In the case of a conditional access system these messages are normally EMMs and ECMs. For the STU to extract these messages it needs to be told the characteristics of these messages. In the case of this specification these characteristics are retrieved from the smart card. The STU uses these characteristics together with the CA_descriptors found in the CAT or PMT to program its filters dedicated to conditional access and assign them to the correct PID. This is not a trivial task as:

- Filters may be time variable.
- Multiple CA messages may exist for one service which requires the set-up of multiple filters.
- Different CA messages for multiple services may be contained in one PID. This requires different filter programs on the same PID to acquire the correct message. The different services should come from the same service provider.
- Support of remultiplexing requires extra association information.

For filtering the table_id byte together with up to 7 filter bytes are compared within the filterable area of a CA message. This area is the first 38 bytes of the payload of the CA message. The offset sets the first location for