EUROPEAN TELECOMMUNICATION STANDARD

ETS 300 331

November 1995

Source: ETSI TC-RES

Reference: DE/RES-03013

ICS: 33.060.50

Key words: DECT, DAM

Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT); DECT Authentication Module (DAM)

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE
Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE
X.400: c=fr, a=atlas, p=etsi, s=secretariat - **Internet:** secretariat@etsi.fr

Tel.: +33 92 94 42 00 - Fax: +33 93 65 47 16

Contents

Foreword

This European Telecommunication Standard (ETS) has been prepared by the Radio Equipment and Systems (RES) Technical Committee of the European Telecommunications Standards Institute (ETSI).

Proposed transposition dates	
Date of adoption of this ETS:	10 November 1995
Date of latest announcement of this ETS (doa):	28 February 1996
Date of latest publication of new National Standard or endorsement of this ETS (dop/e):	31 August 1996
Date of withdrawal of any conflicting National Standard (dow):	31 August 1996

Blank page

1 Scope

This European Telecommunication Standard (ETS) specifies the interface between the DECT Authentication Module (DAM) and the Portable Equipment (PE) for use in the Digital European Cordless Telecommunications (DECT) system as well as those aspects of the internal organisation of the DAM which are related to this use. This is to ensure interoperability between a DAM and a PE independently of the respective manufacturers and operators. The concept of a split of the Portable Part (PP) into a DAM, a type of Integrated Circuit (IC) card, and a PE is described in ETS 300 175-7 [4]. Where equivalent functions are provided in both the PE and the DAM, the DAM functions take precedence over the functions implemented in the PE.

This ETS specifies:

- the requirements for the physical characteristics of the DAM, the electronic signals and the transmission protocols;

- the model which shall be used as a basis for the design of the logical structure of the DAM;

- the security services and facilities;

- the DAM functions which may be requested by the PE over the interface;

- the commands;

- the contents of the files required for the DECT application;

- the application protocol.

This ETS does not specify any aspects related to the administrative management phase. Any internal technical realization of either the DAM or the PE are only specified where these reflect over the interface. This ETS does not specify any of the security algorithms which may be used.

2 Normative references

This ETS incorporates, by dated or undated reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this ETS only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

[1] ETS 300 175-1 (1992): "Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT); Common Interface Part 1: Overview".

[2] ETS 300 175-5 (1992): "Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT); Common Interface Part 5: Network layer".

[3] ETS 300 175-6 (1992): "Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT); Common Interface Part 6: Identities and addressing".

[4] ETS 300 175-7 (1992): "Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT); Common Interface Part 7: Security features".

[5] CCITT Recommendation E.118 (1988): "Automated international telephone credit card system".

[6] ISO Publication 639 (1988): "Codes for the representation of names of languages".

[7] ISO Publication 3166 (1988): "Codes for the representation of names of countries".

[8] ISO Publication 7811-1 (1985): "Identification cards - Recording technique - Part 1: Embossing".

[9] ISO Publication 7811-3 (1985): "Identification cards - Recording technique - Part 3: Location of embossed characters on ID-1 cards".

[10] ISO Publication 7816-1 (1987): "Identification cards - Integrated circuit(s) cards with contacts - Part 1: Physical characteristics".

[11] ISO Publication 7816-2 (1988): "Identification cards - Integrated circuit(s) cards with contacts - Part 2: Dimensions and location of the contacts".

[12] ISO/IEC Publication 7816-3 (1989): "Identification cards - Integrated circuit(s) cards with contacts - Part 3: Electronic signals and transmission protocols".

[13] ISO/IEC Publication 7816-5 (1993): "Identification cards - Integrated circuit(s) cards with contacts - Part 5: Numbering system and registration procedure for application identifiers".

[14] ISO Publication 8859-1 (1987): "Information processing - 8-bit single-byte coded graphic character sets, Part 1: Latin alphabet No. 1".

[15] EN 726-3 (1994): "Terminal Equipment (TE); Requirements for IC cards and terminals for telecommunication use - Part 3: Application independent card requirements".

[16] ETS 300 608: "European digital cellular telecommunications system (Phase 2); Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface" (GSM 11.11, version 4.13.1).

3 Definitions symbols and abbreviations

3.1 Definitions

For the purposes of this ETS, the following definitions apply:

access conditions: A set of security attributes associated with a file.

application: An application consists of a set of security mechanisms, files, data, protocols (excluding transmission protocols).

application protocol: The set of procedures required by the application.

Application Specific Command (ASC) set: To a Dedicated File (DF) can be associated, optionally, an ASC-set (an application specific command set and/or an application specific program). This means that when selecting this application, the general command set of the card is extended or modified by this specific command set. The ASC is valid for the whole subtree of this application unless there are other ASCs defined at the lower levels of this application.

card session: A link between the card and the external world starting with the Answer To Reset (ATR) and ending with a subsequent reset or a deactivation of the card.

current directory: The latest Master File (MF) or DF selected.

current EF: The latest Elementary File (EF) selected (if an EF is selected).

DECT session: That part of the card session dedicated to the DECT operation.

Dedicated File (DF): A file containing access conditions and, optionally, Elementary Files (EFs) or other Dedicated Files (DFs).

directory: General term for MF and DF.

Elementary File (EF): A file containing access conditions and data and no other files.

file: A directory or an organised set of bytes or records in the DAM.

file identifier: The 2 bytes which address a file in the DAM.

Fixed radio Termination (FT): As defined in ETS 300 175-1 [1].

ID-1 Card: The DAM having the format of an ID-1 card (see ISO 7816-1 [10]).

International Portable User Identity (IPUI): The IPUI is an identity that uniquely defines one user within the domain defined by his access rights as related to his IPUI. The IPUI consists of a PUT and a PUN. The IPUI may be locally unique or globally unique depending on the type of the PUT.

Master File (MF): The unique mandatory file containing access conditions and optionally DFs and/or EFs.

offset: Gives the number of bytes from the beginning of a record to the point where the action of the command starts.

padding: One or more bits appended to a message in order to cause the message to contain the required number of bits or bytes.

path: The path is the sequence of file IDs beginning from the current directory, which is either the MF or a DF, to the respective file.

Plug-in card: A second format of DAM (specified in clause 4).

Portable Equipment (DECT Portable Equipment) (PE): The PP without the DAM.

Portable Part (DECT Portable Part)(PP): As defined in ETS 300 175-1 [1].

Portable radio Termination (DECT Portable Termination) (PT): As defined in ETS 300 175-1 [1].

record: A string of bytes within a linear fixed or cyclic EF handled as an entity (see clause 6).

record number: The number which identifies a record within a linear fixed or cyclic EF.

record pointer: The pointer which addresses one record in a linear fixed or cyclic EF.

3.2 Symbols

For the purposes of this ETS the following symbols apply:

Vcc	Supply voltage
Vpp	Programming voltage
"0" to "9" and "A" to "F"	The sixteen hexadecimal digits

3.3 Abbreviations

For the purposes of this ETS the following abbreviations apply:

AC	Authentication Code
ADM	access condition to an EF which is under the control of the authority which creates this file
ALW	ALWays
APDU	Application Protocol Data Unit
ARC	Access Rights Class
ARD	Access Rights Details
ARI	Access Rights Identity; (the ARI consists of an ARC and ARD)
ASC	Application Specific Command
ATR	Answer To Reset
AUT	Authenticated
BCD	Binary Coded Decimal
CHV	Card Holder Verification information; access condition used by the DAM for the verification of the identity of the user
CLA	CLAss
CMOS	Complimentary Metal Oxide Semiconductor
DAM	DECT Authentication Module
DECT	Digital European Cordless Telecommunications
DCK	Derived Cipher Key
DF	Dedicated File
DSAA	DECT Standard Authentication Algorithm
EF	Elementary File
etu	elementary time unit
FP	Fixed Part
FT	Fixed radio Termination
GSM	Global System for Mobile communications
IC	Integrated Circuit
ICC	Integrated Circuit(s) Card
ID	IDentifier
INC	INCrementation bit contained in the Auth type (see ETS 300 175-5, [2])
IPDI	International Portable DAM Identity
IPUI	International Portable User Identity
K	authentication Key
KS	PT authentication session key
KS'	FT authentication session key
KSG	Key Stream Generator
LANG	LANGuage
lgth	the (specific) length of a data unit
LCSR	Last Chosen Subscription Registration
MF	Master File
MMI	Man Machine Interface
NEV	NEVer
PARK	Portable Access Rights Key; consists of ARC and ARD and states the access rights for a PP
PARK{y}	PARK with value y for its PLI
PE	Portable Equipment
PIN	Personal Identification Number; the type of CHV information used for the verification of the identity of the user in this standard
PLI	PARK Length Indicator
PP	Portable Part
PT	Portable radio Termination
PTS	Protocol Type Select
PUN	Portable User Number (see ETS 300 175-6, [3])
PUK	PIN Unblocking Key; the type of UNBLOCK CHV used in the UNBLOCK CHV command in this standard
PUT	Portable User Type (see ETS 300 175-6, [3])

RAND_F	a RANDom challenge issued by an FT
RAND_P	a RANDom challenge (calculated by a DAM and) issued by a PT
RES1	a RESponse calculated by a DAM
RES2	a RESponse calculated by an FT
RFU	Reserved for Future Use
RS	a value used to establish authentication session keys (a Random number used for one Session)
SIM	Subscriber Identity Module
SW1/SW2	Status Word 1/Status Word 2
TPUI	Temporary Portable User Identity
UAK	User Authentication Key
UPI	Universal Personal Identification
XRES1	an eXpected RESponse calculated by an FT
XRES2	an eXpected RESponse calculated by a DAM

4 Physical characteristics

Two physical types of DAM are specified. These are the "ID-1 Card" and the "Plug-in Card".

The physical characteristics of both types of DAM shall be in accordance with ISO 7816-1 and -2 [10, 11] unless otherwise specified. The following additional requirements shall be applied to ensure proper operation in the DECT environment.

4.1 Format and layout

The identification number as defined in EF_{ID} (see clause 10) shall be present on the exterior of the ID-1 Card. The information on the exterior of the Plug-in Card shall include at least the individual account identifier and the check digit.

4.1.1 ID-1 Card

Format and layout of the ID-1 Card shall be in accordance with ISO 7816-1,2 [10, 11]. The card shall have a polarisation mark which indicates how the user should insert the card into the PE.

The PE shall accept embossed ID-1 Cards. The embossing shall be in accordance with ISO 7811-1 [8] and ISO 7811-3 [9]. The contacts of the ID-1 Card may be located on either the front (embossed face) or the back of the card.

4.1.2 Plug-in Card

The Plug-in Card has a width of 25 mm, a height of 15 mm, a thickness the same as an ID-1 Card and a feature for orientation. See annex A for details of the dimensions of the card and the dimensions and location of the contacts.

NOTE: The Plug-in Card is identical to that specified in ETS 300 608 (GSM 11.11) [16] under the name Plug-in SIM.

Annexes A.1 and A.2 of ISO 7816-1 [10] do not apply to the Plug-in Card.

Annex A of ISO 7816-2 [11] applies with the location of the reference points adapted to the smaller size. The three reference points P1, P2 and P3 measure 7,5 mm, 3,3 mm and 20,8 mm, respectively, from 0. The values in table A.1 of ISO 7816-2 [11] are replaced by the corresponding values of figure A.1.

4.2 Temperature range for card operation

The temperature range for full operational use shall be between - 25°C and + 70°C with occasional peaks of up to + 85°C. "Occasional" means not more than 4 hours each time and not over 100 times during the life time of the card.

4.3 Contacts

4.3.1 Provision of contacts

PE: There shall not be any contacting elements in positions C4 and C8. Contact C6 need not be provided for Plug-in Cards.

DAM: Contacts C4 and C8 need not be provided by the DAM. Contact C6 shall not be bonded in the DAM for any function other than supplying Vpp.

4.3.2 Activation and deactivation

The PE shall connect, activate and deactivate the DAM in accordance with the operating procedures specified in ISO/IEC 7816-3 [12].

For any voltage level, monitored during the activation sequence, or during the deactivation sequence following "normal" power-down, the order of the contact activation/deactivation shall be respected.

NOTE: It is recommended that whenever possible the deactivation sequence defined in ISO/IEC 7816-3 [12] should be followed by the PE on all occasions when the PE is powered down.

If the DAM clock is already stopped and is not restarted, the PE is allowed to deactivate all the contacts in any order, provided that all signals reach low level before Vcc leaves high level. If the DAM clock is already stopped and is restarted before the deactivation sequence, then the deactivation sequence specified in ISO/IEC 7816-3, subclause 5.4 [12] shall be followed.

When Vpp is connected to Vcc, as allowed by DECT (see clause 5), then Vpp will be activated and deactivated with Vcc, at the time of the Vcc activation/deactivation, as given in the sequences of ISO/IEC 7816-3 [12], subclauses 5.1 and 5.4.

The voltage level of Vcc, used by DECT, differs from that specified in ISO/IEC 7816-3 [12]. Vcc is powered when it has a value between 4,5 V and 5,5 V.

4.3.3 Inactive contacts

The voltages on contacts C1, C2, C3, C6 and C7 of the PE shall be between 0 and ± 0,4 Volts referenced to ground (C5) when the PE is switched off with the power source connected to the PE. The measurement equipment shall have a resistance of 50 kohms when measuring the voltage on C2, C3, C6 and C7. The resistance shall be 10 kohms when measuring the voltage on C1.

4.3.4 Contact pressure

The contact pressure shall be large enough to ensure reliable and continuous contact (e.g. to overcome oxidisation and to prevent interruption caused by vibration). The radius of any curvature of the contacting elements shall be greater than or equal to 0,8 mm in the contact area.

Under no circumstances may a contact force be greater than 0,5 N per contact.

4.4 Precedence

For Portable Equipment, which accepts both an ID-1 Card and a Plug-in Card, the ID-1 Card shall take precedence over the Plug-in Card.

4.5 Static Protection

Considering that the DAM is a Complimentary Metal Oxide Semiconductor (CMOS) device, the PE manufacturer shall take adequate precautions (in addition to the protection diodes inherent in the DAM) to safeguard the PE, DAM and DAM/PE interface from static discharges at all times, and particularly during DAM insertion into the PE.

5 Electronic signals and transmission protocols

Electronic signals and transmission protocols shall be in accordance with ISO/IEC 7816-3 [12] unless specified otherwise. The following additional requirements shall be applied to ensure proper operation in the DECT environment.

The choice of the transmission protocol(s), to be used to communicate between the PE and the DAM, shall at least include that specified and denoted by T=0 in ISO/IEC 7816-3 [12].

The values given in the tables hereafter are derived from ISO/IEC 7816-3 [12], subclause 4.2 with the following considerations:

- V_{OH} and V_{OL} always refer to the device (PE or DAM) which is driving the interface. V_{IH} and V_{IL} always refer to the device (PE or DAM) which is operating as a receiver on the interface.
- This convention is different to the one used in ISO/IEC 7816-3 [12], which specifically defines an Integrated Circuit Card (ICC) for which its current conventions apply. The following clauses define the specific core requirements for the DAM. For each state (V_{OH}, V_{IH}, V_{IL} and V_{OL}) a positive current is defined as flowing out of the entity (PE or DAM) in that state.
- The high current options of ISO/IEC 7816-3 [12] for V_{IH} and V_{OH} are not specified for the DAM as they apply to N-channel Metal Oxide Semiconductor (NMOS) technology requirements. No realisation of the DAM using NMOS is foreseen.

5.1 Supply voltage Vcc (contact C1)

The DAM shall be operated within the following limits:

Table 1: Electrical characteristics of Vcc under normal operating conditions

Symbol	Minimum	Maximum	Unit
V_{CC}	4,5	5,5	V
I_{CC}		10	mA

NOTE 1: The use of 3 Volt technology is currently under study in several ETSI groups. 3 Volt technology will be incorporated into this standard as an amendment.

The current consumption of the DAM shall not exceed the value given in table 1 at any frequency accepted by the DAM.

When the DAM is in idle state (see below) the current consumption of the card shall not exceed 200 µA at 1 MHz and 25 °C; it shall not exceed 1 mA at any frequency accepted by the DAM.

The PE shall source the maximum current requirements defined above. It shall also be able to counteract spikes in the current consumption of the card up to a maximum charge of 40 nAs with no more than 400 ns duration and an amplitude of at most 200 mA, ensuring that the supply voltage stays in the specified range.

NOTE 2: A possible solution would be to place a capacitor (e.g. 100 nF, ceramic) as close as possible to the contacting elements.

5.2 Reset RST (contact C2)

The PE shall operate the DAM within the following limits:

Table 2: Electrical characteristics of RST under normal operating conditions

Symbol	Conditions	Minimum	Maximum
V_{OH}	I_{OHmax} = + 20 µA	Vcc - 0,7	Vcc (note)
V_{OL}	I_{OLmax} = - 200 µa	0V (note)	0,6 V
t_R t_F	C_{out} = C_{in} = 30 pF		400 µS

NOTE: To allow for overshoot the voltage on RST shall remain between - 0,3 V and Vcc + 0,3 V during dynamic operation.

5.3 Programming voltage Vpp (contact C6)

DAMs shall not require any programming voltage on Vpp. The PE need not provide contact C6. If the PE provides contact C6, then, in the case of ID-1 Cards the same voltage shall be supplied on Vpp as on Vcc, while in the case of Plug-in Cards the PE need not provide any voltage on C6. Contact C6 may be connected to Vcc in any PE but shall not be connected to ground.

5.4 Clock CLK (contact C3)

The DAM shall support 1 to 5 MHz. The clock shall be supplied by the PE. No "internal clock" DAMs shall be used.

The duty cycle shall be between 40 % and 60 % of the period during stable operation.

The PE shall operate the DAM within the following limits:

Table 3: Electrical characteristics of CLK under normal operating conditions

Symbol	Conditions	Minimum	Maximum
V_{OH}	I_{OHmax} = + 20 µA	0,7 x Vcc	Vcc (note)
V_{OL}	I_{OLmax} = - 200 µa	0 V (note)	0,5 V
t_R t_F	C_{out} = C_{in} = 30 pF		9 % of period with a maximum of 0,5 µs

NOTE: To allow for overshoot the voltage on CLK shall remain between - 0,3 V and Vcc + 0,3 V during dynamic operation.

5.5 I/O (contact C7)

Table 4 defines the electrical characteristics of the I/O (contact C7). The values given in the table have the effect of defining the values of the pull-up resistor in the PE and the impedances of the drivers and receivers in the PE and DAM.

Table 4: Electrical characteristics of I/O under normal operating conditions

Symbol	Conditions	Minimum	Maximum
V_{IH}	I_{IHmax} = ± 20 µA (note 2)	0,7 x Vcc	Vcc + 0,3 V
V_{IL}	I_{ILmax} = + 1 mA	- 0,3 V	0,8 V
V_{OH} (note 1)	I_{OHmax} = +20µA	3,8 V	Vcc (note 3)
V_{OL}	I_{OLmax} = -1ma	0 V[3)]	0,4 V
t_R t_F	C_{out} = C_{in} = 30pF		1µS

NOTE 1: It is assumed that a pull-up resistor is used in the interface device (recommended value: 20 kohms).

NOTE 2: During static conditions (idle state) only the positive value can apply. Under dynamic operating conditions (transmission) short term voltage spikes on the I/O line may cause a current reversal.

NOTE 3: To allow for overshoot the voltage on the I/O shall remain between - 0,3 V and Vcc + 0,3 V during dynamic operation.

5.6 States

There are two states for the DAM while the power supply is on:

- the DAM is in operating state when it executes a command. This state also includes transmission from and to the PE;

- the DAM is in idle state at any other time. It shall retain all pertinent data during this state.

The DAM may support a clock stop mode. The clock shall only be switched off subject to the conditions specified in EF_{ICC} (see clause 10). A PE shall wait at least five (5) elementary time units after having received the last bit of the response before it switches off the clock (if it is allowed to do so). It shall wait at least two (2) elementary time units before it sends the first command after having started the clock.

5.7 Baudrate

The baudrate for all communications shall be (clock frequency)/372.

5.8 Answer To Reset (ATR)

The ATR is information presented by the DAM to the PE at the beginning of the card session and gives operational requirements.

5.8.1 Structure and contents

The following table gives an explanation of the characters specified in ISO/IEC 7816-3 [12] and the requirements for their use in DECT. The ATR consists of at most 33 characters. The PE shall be able to receive interface characters for transmission protocols other than T = 0, historical characters and a check byte, even if only T = 0 is used by the PE.

Table 5: ATR

Character	**Contents**	**sent by the card**	**a) evaluation by the PE b) reaction by the PE**
1) Initial character TS	coding convention for all characters (direct or inverse convention)	always	a) always b) using subsequent appropriate convention
2) Format character T0	subsequent interface characters, number of historical characters	always	a) always b) identifying the subsequent characters accordingly
3) Interface character (global) TA1	parameters to calculate the work etu	optional	a) always if present b) if TA1 is not "11", PTS procedure shall be used (see below)
4) Interface character (global) TB1	parameters to calculate the programming voltage and current	optional	a) always if present b) if PI1 is not 0, then reject the DAM (in accordance with clause 5.10)
5) Interface character (global) TC1	parameters to calculate the extra guardtime requested by the card; no extra guardtime is used to send characters from the card to the PE	optional	a) always if present b) if TC1 is not 0 or not 255, then reject the DAM (in accordance with clause 5.10); see the note after the table
Character	**Contents**	**sent by the card**	**a) evaluation by the PE b) reaction by the PE**
6) Interface character TD1	protocol type; indicator for the presence of interface characters, specifying rules to be used for transmissions with the given protocol type	optional	a) always if present b) identifying the subsequent characters accordingly
7) Interface character (specific) TA2	not used for protocol T=0	optional	a) optional b) --------
8) Interface character (global) TB2	parameter to calculate the programming voltage	never	the allowed value of TB1 above defines that an external programming voltage is not applicable

(continued)

Table 5: ATR (concluded)

9)	Interface character (specific) TC2	parameters to calculate the work waiting time	optional	a) always if present b) using the work waiting time accordingly
10)	Interface character TDi (i>1)	protocol type; indicator for the presence of interface characters, specifying rules to be used for transmissions with the given protocol type	optional	a) always if present b) identifying the subsequent characters accordingly
11)	Interface character TAi, TBi TCi (i>2)	characters which contain interface characters for other transmission	optional	a) optional b) -------- protocols
12)	Historical characters T1,...,TK	contents not specified in ISO/IEC	optional	a) optional b) --------
13)	Check character TCK	check byte (exclusive -ORing)	not sent if only T=0 is indicated in in cases shall sent	a) optional b) -------- the ATR; all other TCK be

NOTE: According to ISO/IEC 7816-3:1989/AM2 (see annex C [3]) N=255 indicates that the minimum delay is 12 etu for the asynchronous half-duplex character transmission protocol.

5.8.2 Protocol Type Select (PTS) procedure

Figure 1 gives an example of using PTS according to ISO/IEC 7816-3 [12].

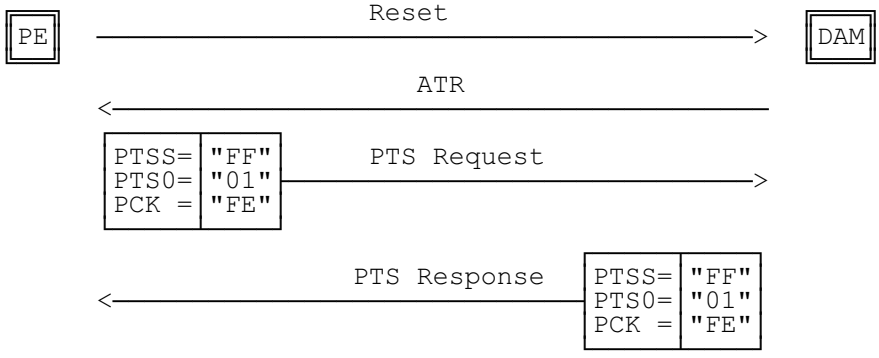


Figure 1: PTS procedure

After the completion of this procedure the transport protocol has changed from T=0 to T=1 (with the default values for the transport protocol).

5.9 Bit/character duration and sampling time

The bit/character duration and sampling time specified in ISO/IEC 7816-3 [12], subclauses 6.1.1 and 6.1.2, are valid for all communications.

5.10 Error handling

Following receipt of an ATR, which is not in accordance with this specification, e.g. because of forbidden ATR characters or too few bytes being transmitted, the PE shall perform a Reset. The PE shall not reject the DAM until at least three consecutive wrong ATRs are received.

During the transmission of the ATR and the protocol type selection, the error detection and character repetition procedure specified in ISO/IEC 7816-3 [12], subclause 6.1.3, is optional for the PE. For the subsequent transmission on the basis of T=0 this procedure is mandatory for the PE.

For the DAM the error detection and character repetition procedure is mandatory for all communications.

5.11 Presence of the DAM

In addition to mechanical detection that the DAM is still present in the PE an electrical check shall be performed. This is done by a hardware check on the contacts measuring the impedance or the power consumption, or by a software check according to the procedure specified in clause 11.

6 Logical model

This clause describes the logical structure for a DAM, the code associated with it, and the structure of files used.

6.1 General description

Figure 2 shows the general structural relationships which may exist between files. The files are organised in a hierarchical structure and are of one of three types as defined below. These files may be either administrative or application specific. The operating system handles the access to the data stored in different files.

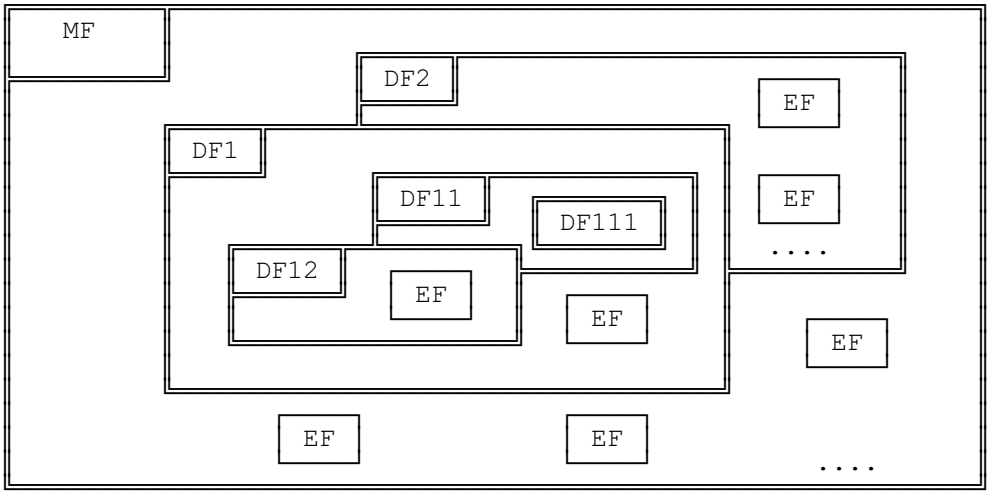


Figure 2: Organisation of memory

Files are composed of a header, which is internally managed by the DAM, and optionally a body part. The information of the header is related to the structure and attributes of the file and may be obtained by using

the commands GET RESPONSE or STATUS. This information is fixed during the administrative phase. The body part contains the data of the file.

6.2 File identifier

A file IDentifier (ID) is used to address or identify each specific file. The file ID consists of two bytes and shall be coded in hexadecimal notation. They are specified in clause 10.

The first byte identifies the type of file, its level in the hierarchy and for DECT is:

- "3F": Master File (MF) (coded "3F00");
- "7F": Dedicated File (DF);
- "00", "01" and "2F": Elementary File (EF) under the MF;
- "6F": EF under a DF.

File IDs shall be subject to the following conditions:

- the file ID shall be assigned at the time of the creation of the file concerned;
- no two files under the same parent shall have the same ID;
- a child and any parent, either immediate or remote in the direct hierarchy, e.g. grandparent, shall never have the same file ID.

In this way each file is uniquely identified.

6.3 Dedicated Files (DF)

A DF is a functional grouping of files consisting of itself and all those files which contain this DF in their parental hierarchy (that is to say it consists of the DF and its complete "subtree"). A DF consists of only a header part.

The DECT application is contained in DF_{DECT} which can be placed at any level. The identifier for and the path to DF_{DECT} are given in the Elementary File EF_{DIR}. Though any ID may be used it is recommended to use "7F 50" as a default value for the identifier of DF_{DECT}. The DFs for registrations shall be placed as immediate children under DF_{DECT}. Each registration shall be placed in its own DF.

NOTE: Further directories for specific access profiles or applications may be added as part of later amendments to this standard. This may, for instance, be a telecom directory based on those specified in prEN 726-6 (see annex C) or in ETS 300 608 (GSM 11.11) [16].

6.4 Elementary Files (EF)

An EF is composed of a header and a body part. The following three structures of an EF are used by DECT.

6.4.1 Transparent EF

An EF with a transparent structure consists of a sequence of bytes. When reading or updating, the sequence of bytes to be acted upon is referenced by a relative address (offset), which indicates the start position (in bytes), and the number of bytes to be read or updated. The first byte of a transparent EF has the relative address "00". The total data length of the body of the EF is indicated in the header of the EF.

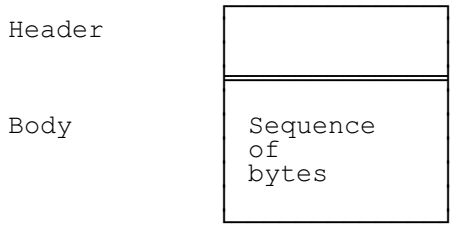


Figure 3: Structure of a transparent EF

6.4.2 Linear fixed EF

An EF with linear fixed structure consists of a sequence of (n) records all having the same (fixed) length. The first record is record number 1. The length of a record as well as this value multiplied by the number of records are indicated in the header of the EF.

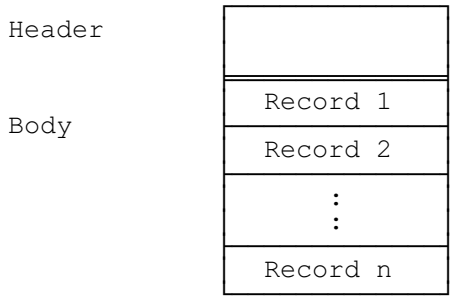


Figure 4: Structure of a linear fixed EF

There are several methods to access records within an EF of this type:

- absolutely using the record number;

- when the record pointer is not set it shall be possible to perform an action on the first or the last record;

- when the record pointer is set it shall be possible to perform an action on this record, the next record (unless the record pointer is set to the last record) or the previous record (unless the record pointer is set to the first record);

- by identifying a record using pattern seek starting:

  - forwards from the beginning of the file;
  - forwards from the record following the one at which the record pointer is set (unless the record pointer is set to the last record);
  - backwards from the end of the file;
  - backwards from the record preceding the one at which the record pointer is set (unless the record pointer is set to the first record).

If an action following selection of a record is aborted, then the record pointer shall remain set at the record at which it was set prior to the action.

NOTE: It is not possible, at present, to have more than 255 records in a file of this type, and each record cannot be greater than 255 bytes.

6.4.3 Cyclic EF

Cyclic files are used for storing records in chronological order. When all records have been used for storage, then the next storage of data shall overwrite the oldest information.

An EF with a cyclic structure consists of a fixed number of records with the same (fixed) length. In this file structure there is a link between the last record (n) and the first record. When the record pointer is set to the last record n, then the next record is record 1. Similarly, when the record pointer is set to record 1, then the previous record is record n. The last updated record containing the newest data is record number 1, and the oldest data is held in record number n.

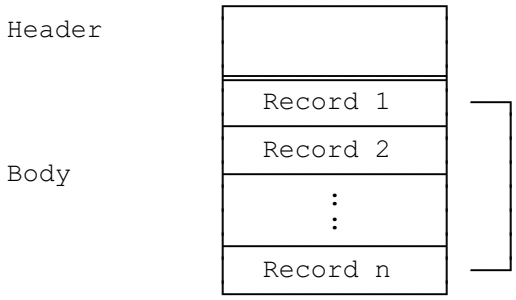


Figure 5: Structure of a cyclic file

For update operations only PREVIOUS record shall be used. For reading operations, the methods of addressing are First, Last, Next, Previous, Current and Record Number.

After selection of a cyclic file (for either operation), the record pointer shall be set at the record updated or increased last. If an action following selection of a record is aborted, then the record pointer shall remain set at the record at which it was set prior to the action.

NOTE: It is not possible, at present, to have more than 255 records in a file of this type, and each record cannot be greater than 255 bytes.

6.5 Methods for selecting the DECT application

After the ATR the MF is implicitly selected and becomes the current directory. The DECT application may then be selected by using the application identifier for DECT (direct selection) or according to the methods specified in subclause 6.6 by using the file identifiers stored in EF_{DIR} (see clause 10).

The PE shall support both selection methods while the card may only support one of the two methods. If the card does not contain EF_{DIR} or if it contains EF_{DIR} but no path to DF_{DECT}, then the PE shall try to directly select the DECT application before it may reject the card for the reason "application not found".

6.6 Methods for selecting a file

After the MF has (implicitly) been selected each file may be selected by using the SELECT function in accordance with the following rules.

Selecting a DF or the MF sets the current directory. After such a selection there is no current EF. Selecting an EF sets the current EF and the current directory remains the DF or MF which is the parent of this EF. The current EF is always a child of the current directory.

Any application specific command shall only be operable if it is specific to the current directory.

The following files may be selected from the last selected file:

- any file which is an immediate child of the current directory;
- any DF which is an immediate child of the parent of the current DF;
- the parent of the current directory;
- the current DF;
- the MF.

This means in particular that a DF shall be selected prior to the selection of any of its EFs. All selections are made using the file ID.

The following figure gives the logical structure for the DECT application.

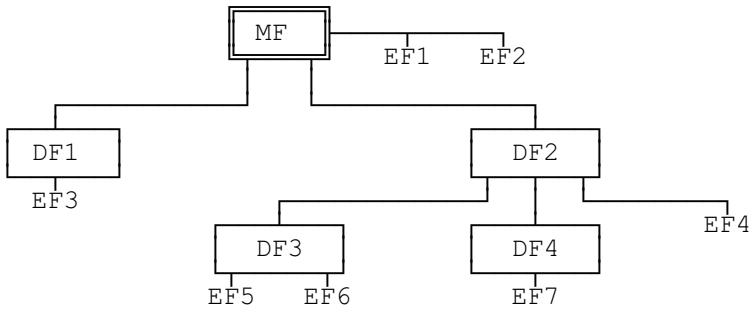


Figure 6: Logical structure

The following table gives the valid selections for DECT for the logical structure in figure 6. Reselection of the last selected file is also allowed but not shown.

Table 6: File selection

Last selected file	Valid selections
MF	DF1, DF2, EF1, EF2
DF1	MF, DF2, EF3
DF2	MF, DF1, DF3, DF4, EF4
DF3	MF, DF2, DF4, EF5, EF6
DF4	MF, DF2, DF3, EF7
EF1	MF, DF1, DF2, EF2
EF2	MF, DF1, DF2, EF1
EF3	MF, DF1, DF2
EF4	MF, DF1, DF2, DF3, DF4
EF5	MF, DF2, DF3, DF4, EF6
EF6	MF, DF2, DF3, DF4, EF5
EF7	MF, DF2, DF3, DF4

6.7 Reservation of file IDs

The following file IDs are reserved for future use by DECT.

1) DFs:

   a) administrative use: "7F 60";
   b) registrations: "7F 7X";
   c) operational use: "7F 50" (DECT application), "7F 51", "7F 52";

2) EFs:

a) administrative use:

- "2F DX" in the MF "3F 00";
- "6F DX" in the DFs "7F 50", "7F 51", "7F 52", "7F 7X";

b) operational use:

- "2F 1X" in the MF "3F 00";
- "6F XX" (except "6F DX") in the DFs "7F 50", "7F 51", "7F 52" "7F 7X".

In all the above X ranges, unless otherwise stated, from "0" to "F". The coding "FF FF" shall not be used.

7 Security services and facilities

The security aspects of DECT are described in ETS 300 175-7 [4]. This clause considers those aspects which are related to the DAM.

7.1 Overview

The PE interacts with the DAM, the Fixed radio Termination (FT) and the user when performing the following security services and facilities:

- authentication of a PT (subclause 7.2.1);
- authentication of an FT (subclause 7.2.2);
- mutual authentication of PT and FT (subclause 7.2.3);
- authentication of a user to an FT (subclause 7.2.4);
- allocation of initial User Authentication Key (UAK) (subclause 7.3);
- data confidentiality over the air interface (subclause 7.4);
- access rights to the DECT system (subclause 7.5);
- access control of files in the DAM (subclause 7.6);
- entering a new subscription registration (subclause 7.8.1);
- updating an existing subscription registration (subclause 7.8.2);
- termination of an existing subscription registration (subclause 7.8.3).

7.1.1 Authentication keys

An authentication key K is used in the authentication mechanisms defined in subclauses 7.2.1 to 7.2.4. The authentication key K has length 128 bits.

The authentication key is derived from other keys. These keys are identified below. The processes for deriving K from these keys are given in subclause 7.1.3.

UAK: User Authentication Key (maximum 128 bits), stored in the DAM.
AC: Authentication Code (16-32 bits), stored in the DAM.
UPI: User Personal Identity (16-32 bits), entered manually into the PE.

7.1.2 Cipher key

The Key Stream Generator (KSG) which is used to provide data confidentiality over the air interface, is part of the PE. The DAM generates and, if required, stores the following cipher key:

DCK: Derived Cipher Key (64 bits).

7.1.3 Algorithms and processes

The DECT Standard Authentication Algorithm (DSAA) shall be available in the DAM. The use of other authentication algorithms in addition to the DSAA is optional.

The following processes defined in ETS 300 175-7 [4] are performed by the DAM:

1) Process B1:

   a) Purpose:

      to expand the input to 128 bits (if the input is less than 128 bits);

   b) Input:

      AC or UAK;

   c) Output:

      Authentication Key K;

2) Process B2:

   a) Purpose:

      - to generate a key for User Authentication;

   b) Input:

      - UPI and UAK;

   c) Output:

      - Authentication Key K;

3) Process A11:

   a) Purpose:

      - to generate an Authentication Session Key as input to the A12 process;

   b) Input:

      - K and RS;

   c) Output:

      - Session Key KS;

4) Process A12:

   a) Purpose:

      - to calculate RES1 for PT or User Authentication and a DCK;

   b) Input:

      - KS and RAND_F;

   c) Output:

      - RES1 and DCK;

5) Process A21:

   a) Purpose:

      - to generate an Authentication Session Key as input to the A22 process;

   b) Input:

      - K and RS;

   c) Output:

   d) Session Key KS';

6) Process A22:

   a) Purpose:

      - to calculate RES2 for FT authentication;

   b) Input:

      - KS' and RAND_P;

   c) Output:

      - RES2.

7.2 Authentication

This subclause describes the different authentication mechanisms which are specified in ETS 300 175-7 [4]. For the specification of the corresponding procedures see clause 11.

7.2.1 Authentication of a Portable radio Termination (PT)

This mechanism is employed by the FT to authenticate the DAM when placed in the PE with respect to a selected subscription registration.

The FT obtains two random numbers RS and RAND_F, which are sent to the DAM via the PE over the air interface in an AUTHentication-REQUEST message. The DAM replies to the PE with RES1, DCK and, if stored in the DAM, the ZAP. RES1, DCK (if requested) and ZAP (if available) are sent to the FT over the air interface in the AUTHentication-REPLY message. If requested in the AUTHentication-REQUEST message, the DCK is stored in the DAM for later use by the KSG in the PE. The FT verifies RES1, by comparing it with the value XRES1 calculated by itself, and the ZAP by comparing it with the one it expects. If XRES1 and RES1 are not equal, then the FT drops the call (see ETS 300 175-7 [4]).

The processes used are B1, A11 and A12. The key used for process B1 is found in EF_{AC} or EF_{UAK}.

7.2.2 Authentication of a Fixed Termination (FT)

This mechanism is employed by the DAM, when placed in the PE, to authenticate the FT with respect to a selected subscription registration.

The DAM generates a random number RAND_P which is sent via the PE and the air interface to the FT in an AUTHentication-REQUEST message.

The FT replies by sending the random number RS and the result RES2 in an AUTHentication-REPLY message or by sending an AUTHentication-REJECT message.

If the PE receives an AUTHentication-REJECT from the FT, then it performs the actions as specified in subclause 6.5.2.2 of ETS 300 175-7 [4].

If the PE receives an AUTHentication-REPLY it forwards RS and RES2 to the DAM. The DAM verifies RES2 by comparing it with the value XRES2 calculated by itself. If XRES2 and RES2 are not equal, then the PT drops the call (see ETS 300 175-7 [4]).

If the verification is positive, the FT-Authenticated (AUT) condition is fulfilled. FT-AUT can be used as an access condition to certain files at the discretion of the registration provider. Hence the access condition FT-AUT is only valid for the subscription registration involved.

The processes used are B1, A21 and A22. The algorithm ID is found in EF_{SR}, while the key is contained in EF_{AC} or EF_{UAK}.

7.2.3 User authentication

This mechanism is employed by the FT to authenticate the user of the DAM when placed in the PE with respect to a selected subscription registration.

User authentication is achieved in the same way as an authentication of a PT, the differences being that the B2 process is used instead of the B1 process, and the user is requested to give his UPI to the DAM via the PE. The input to the DAM consists of UPI, RS and RAND_F, the latter two are provided by the FT. RES1 and, if requested, the DCK are sent to the FT over the air interface in an AUTHentication-REPLY message. If requested in the AUTHentication-REQUEST message, the DCK is stored in the DAM for later use by the KSG in the PE. The FT verifies RES1 by comparing it with the value XRES1 calculated by itself.

7.2.4 Mutual authentication

Mutual authentication between the DAM or the user with respect to a selected subscription registration and the FT is achieved in one of the following two ways:

Direct method: an authentication of a PT or a user authentication is followed by an authentication of an FT.

Indirect method: this is a combination of an authentication of a PT or a user authentication and the data confidentiality service. The sequence of events to achieve mutual authentication is as follows: Authentication of a PT is performed and the DCK is stored in the DAM (if required). The use of the data confidentiality service is then enforced between the FT and the PT using the DCK. By enforcing data confidentiality, an FT authentication is indirectly achieved.

7.3 UAK allocation

This mechanism is performed only for the initial allocation of a UAK by means of an AC stored in the EF_{AC} for a selected subscription registration. The FT sends a KEY-ALLOCATE message containing an allocation type element, RAND_F and RS to the PE over the air interface and initiates hereby an authentication of a PT. For the specification of the corresponding procedure see clause 11.

The PE examines the allocation type element. If it is unacceptable the PE returns an AUTHentication-REJECT message to the FT.

If the allocation type element is acceptable, the random number RAND_F and RS are sent to the DAM which replies to the PE with RES1, DCK and RAND_P. RES1 and RAND_P are sent to the FT over the air interface in an AUTHentication-REQUEST message, this logically initiates an authentication of an FT.

If the verification by the FT of an authentication of a PT is unsuccessful, the FT shall drop the call (see ETS 300 175-7 [4], subclause 6.5.6.2).

After a successful verification by the FT of the authentication of a PT, it replies with an AUTHentication-REPLY message containing RES2 and RS (RS is identical to the one contained in the KEY-ALLOCATION message). The PE forwards RS, RES2 and the UAK number, which is contained in the allocation type element to the DAM. The DAM verifies RES2 by comparing it with the value XRES2 calculated by itself.

If the verification is successful, it stores the UAK obtained during the computation of XRES2 under the appropriate number.

If the verification fails, the DAM shall inform the PE which shall then drop the call.

7.4 Data confidentiality

In order to provide data confidentiality as requested in a CIPHER-REQUEST or CIPHER-SUGGEST message sent over the air interface, the FT and the PT are required to share a cipher key. This key is used to generate, in conjunction with a key stream generator in the PE, a key stream which is employed for the enciphering of data over the air interface (see ETS 300 175-6 [3], subclause 6.4). It is obtained as the DCK during user authentication or authentication of a PT.

If the DCK is required to be stored, it shall be stored in the DAM in the corresponding EF_{DCK}.

7.5 Access rights to the DECT system

Four categories of identities, used for identification and addressing in a general DECT environment, are specified in ETS 300 175-6 [3]. This subclause specifies the PP identities related to the DAM.

Every subscription registration in a DAM contains at least one Portable Access Rights Key (PARK), used to identify its domain of use, and a unique International Portable User Identity (IPUI), to identify the PP in that domain. The (IPUI,PARK) pair uniquely identifies the DAM and its access rights in a domain. A PE is allowed to access only those radio FTs which can be identified by any of the PARKs stored in the DAM. To access a system the PARK needs to be contained in the subscription registration which has been selected by the user. One and the same PARK can be associated to different subscription registrations (IPUIs).

The common base for the DECT identity structure is the Access Rights Class (ARC) and the Access Rights Details (ARD). These must be known by both the FT and the DAM. In the FT the ARC and ARD are called Access Rights Identity (ARI) while they are called PARK in the DAM. The distinction between PARK and ARI is that each PARK can have a group of ARDs allocated to it, PARK{y}. Here "y" denotes the value of the PARK Length Indicator (PLI) given in the PT subscription process. One PARK{y} can relate to several ARIs of several FTs by a suitable choice of the value {y} which indicates the "don't care bits" in the received ARIs (see ETS 300 175-6 [3], clause 4). This permits the DAM to have extended access rights using a low number of PARK{y}s.

A PT is identified by its pairs of PARK{y} and IPUI. A PT is only allowed to access an FT if one of its PARKs contained in the subscription registration chosen by the user includes one of the ARIs of the FT. A subscription registration in the DAM is fully identified by the chosen ARI and IPUI in the FT.

The Portable User Type (PUT) and Portable User Number (PUN) form the IPUI. This identity can either be globally unique or locally unique. At the present there are 7 types of IPUIs. IPUIs (except class N IPUI) have a variable length. The portable identity information element of the network layer contains a field to indicate the selected length of the IPUI.

An IPUI can be replaced by a temporary and shorter identity for paging. That is an individual Temporary Portable User Identity (TPUI), see ETS 300 175-6 [3].

Four access rights classes A-D and a number of international portable user identities have been defined to meet the need for a differentiation in the identity structures. An overview table is contained in ETS 300 175-6 [3].

7.6 File access control

Every file has its own specific access condition for each command. The relevant access condition of the last selected file shall be fulfilled before the requested action can take place. No file access conditions are currently assigned by DECT to the MF and the DFs.

For each file:

- the access conditions for the commands READ and SEEK are identical;

- the access conditions for the commands SELECT and STATUS are ALWays.

The following access conditions are possible:

ALWAYS: the action can be performed without any restriction.

CHV1 (Card Holder Verification 1): the action shall only be possible if one of the following three conditions is fulfilled:

- a correct CHV1 value has already been presented to the DAM during the current session;

- the CHV1 enabled/disabled indicator is set to "disabled";

- UNBLOCK CHV1 has been successfully performed during the current session.

The CHV1 value of the relevant EF_{CHV1} shall be used. This means that EF_{CHV1} is a child of the current Directory. If there is no such EF, then the relevant CHV of the parent directory shall be used.

CHV2 (Card Holder Verification 2): the action shall only be possible if one of the following three conditions is fulfilled:

- a correct CHV2 value has already been presented to the DAM during the current session;

- the CHV2 enabled/disabled indicator is set to "disabled";

- UNBLOCK CHV2 has been successfully performed during the current session.

The CHV2 value of the relevant EF_{CHV2} shall be used. This means that EF_{CHV2} is a child of the current Directory. If there is no such EF, then the relevant CHV of the parent directory shall be used.

FT-AUT: the action shall only be possible after a successful Authentication of an FT by the DAM.

ADM: allocation of these levels and the respective requirements for their fulfilment are the responsibility of the appropriate administrative authority.

NEVER: the action can not be performed over the DAM/PE interface. The DAM may perform the action internally.

Access conditions are not hierarchical. For instance, the correct presentation of CHV2 does not allow actions to be performed which require correct presentation of CHV1. An access condition which has been satisfied remains valid until the end of the DECT session as long as the corresponding secret code remains unblocked, i.e. only after three consecutive wrong attempts, not necessarily in the same card session, the access rights previously granted by this secret code are lost immediately.

The PE shall determine whether CHV2 is available. If CHV2 is not available, then commands on an EF requiring CHV2, e.g. VERIFY CHV2, shall not be executable.

7.7 Identification, keying and algorithm information

This subclause gives an overview of the purpose and use of the security related information which is stored in the DAM. See clause 10 for more information on the contents of the files and their coding.

7.7.1 Subscription registration information

The DAM contains a mandatory file EF_{LSR} listing all subscription registrations together with the file identifiers of the respective DFs.

For every subscription registration there is a mandatory file EF_{SR}. It shall contain the following information:

- DECT phase;
- key types (AC, UAK, DCK);
- available key numbers per key type (AC,UAK);
- authentication algorithm identifiers.

7.7.2 IPUI

For every subscription registration there is a mandatory file EF_{IPUI} which contains the unique IPUI associated to it.

7.7.3 PARK

For every subscription registration there is a mandatory file EF_{PARK} which contains the different PARK{y}s associated to the IPUI for this subscription registration and their respective PLIs.

7.7.4 TPUI

For every subscription registration there is an optional file EF_{TPUI} which contains the unique individual TPUI which is associated to the IPUI for this subscription registration.

7.7.5 ZAP

The purpose of the ZAP information is to suspend or terminate a subscription registration. A particular value of the ZAP information is expected in the AUTHentication-REPLY message in an Authentication of a PT. If the ZAP information is not as expected, then the PT will not be successfully authenticated. Implementation and use of the ZAP mechanism are at the discretion of the service provider.

For every subscription registration there is an optional file EF_{ZAP} which contains the ZAP value. The only action of the DAM with respect to the contents of this EF is to increment the ZAP value (see subclause 11.4.1 for details).

7.7.6 User Authentication Key(s) (UAK)

For the authentication mechanism a UAK may be used. For every subscription registration there is a mandatory file EF_{UAK} containing at least one (1) and at most eight (8) UAKs. The number of keys stored is up to the service provider. For every authentication mechanism, where a UAK is used, the EF_{UAK} under the appropriate subscription registration shall be used.

The UAKs are stored in the DAM at personalisation or by means of the UAK allocation mechanism.

7.7.7 Authentication Code(s) (AC)

For the authentication mechanisms an AC may be used. For every subscription registration there is a mandatory file EF_{AC} containing at least one (1) and at most eight (8) ACs. The number of keys stored is up to the service provider. For every authentication, where an AC is used, the EF_{AC} under the appropriate subscription registration shall be used.

The ACs are entered into the DAM by the user via the PE.

7.7.8 Derived Cipher Key (DCK)

For data confidentiality over the air interface a DCK may be used by the PE. For every subscription registration there is a mandatory file EF_{DCK} which contains at most one DCK.

The DCK is generated in the DAM during a user authentication or an authentication of a PT. If a DCK needs to be stored for later use, it shall be stored in the DAM.

7.8 Subscription registration maintenance

7.8.1 Entering a new subscription registration

This procedure is employed by a user to enter the data for a new subscription registration in the DAM.

The user sets the PE into the special "new subscription registration" mode. The user then enters the AC previously obtained for this new subscription registration via the PE. The PE stores the AC in EF_{AC} of the new subscription registration and updates its EF_{SR}. The PE then reads the International Portable DAM Identity (IPDI) and sends it to the FT in an ACCESS-RIGHTS-REQUEST message. The FT sends an ACCESS-RIGHTS-ACCEPT message to the PE containing the IPUI, PARK and additional optional data such as the ZAP value. The PE then initiates a UAK allocation. After a successful UAK allocation the PE stores the values obtained in the ACCESS-RIGHTS-ACCEPT message in the relevant files of the new subscription registration in the DAM. If the UAK allocation is not successful the procedure is aborted.

7.8.2 Updating an existing subscription registration

This procedure is used for an addition or deletion of a PARK in the selected subscription registration. This procedure shall only be performed after a successful mutual authentication of the FT and the PT.

The PE sends an ACCESS-RIGHTS-REQUEST message in case of an addition or an ACCESS-RIGHTS-TERMINATE-REQUEST message in case of a deletion to the FT. The FT sends an ACCESS-RIGHTS-ACCEPT message or an ACCESS-RIGHTS-TERMINATE-ACCEPT message to the PE. The PE adds or deletes the PARK in the selected subscription registration.

7.8.3 Terminating an existing subscription registration

This procedure is employed by the FT to terminate a subscription registration. This procedure shall only be performed after a successful mutual authentication of the FT and the PT.

The FT sends an ACCESS-RIGHTS-TERMINATE REQUEST message to the PE. The PE sends an ACCESS-RIGHTS-TERMINATE-ACCEPT message to the FT and deletes the subscription registration in the DAM which corresponds to the IPUI obtained in the ACCESS-RIGHTS-TERMINATE-REQUEST message.

8 Description of the functions

This clause gives a functional description of the commands and their respective responses. Associated status conditions, error codes and their corresponding coding are specified in clause 9.

It shall be mandatory for all cards complying with this ETS to support all functions described in it. The command GET RESPONSE which is needed for the protocol T=0 is specified in clause 9.

The following table lists the file types and structures together with the functions which may act on them during a DECT session. These are indicated by an asterisk (*).

Table 7: Functions on files in DECT session

	File				
Function	MF	DF	EF transparent	EF linear fixed	EF cyclic
SELECT	*	*	*	*	*
STATUS	*	*	*	*	*
READ BINARY			*		
UPDATE BINARY			*		
READ RECORD				*	*
UPDATE RECORD				*	*
SEEK				*	
INCREASE					*
INVALIDATE			*	*	*
REHABILITATE			*	*	*

8.1 SELECT

This function selects a file according to the methods described in clause 6. After a successful selection the record pointer in a linear fixed file is undefined. The record pointer in a cyclic file shall address the last record which has been updated or increased.

Input:

- file ID or application identifier (if DF_{DECT} is selected using direct selection).

Output:

- if the selected file is the MF or a DF:

  file ID, total memory space available, CHV enabled/disabled indicator, CHV status;

- if the selected file is an EF:

  file ID, access conditions, indication invalidated/not invalidated, indication if readable when invalidated, structure of EF and length of the records in case of linear fixed structure or cyclic structure;

- if the selected file is an EF_{CHV}:

  the output also contains the number of remaining CHV attempts.

8.2 STATUS

This function returns information concerning the current directory. A current EF is not affected by the STATUS function.

Input:

- none.

Output:

- file ID, total memory space available, CHV disabled/enabled indicator, CHV status (identical to SELECT above).

8.3 READ BINARY

This function reads a string of bytes from the current transparent EF. This function shall only be performed if the READ access condition for this EF is satisfied.

Input:

- relative address (offset) and the length (in bytes) of the string.

Output:

- string of bytes.

8.4 UPDATE BINARY

This function updates the current transparent EF with a string of bytes. This function shall only be performed if the UPDATE access condition for this EF is satisfied. The update can be considered as a replacement of the string already present in the EF by the string given in the update command.

Input:

- relative address (offset) from the beginning of the EF and the length (in bytes) of the string;

- string of bytes.

Output:

- none.

8.5 READ RECORD

This function reads one complete record in the current linear fixed or cyclic EF. This function shall only be performed if the READ access condition for this EF is satisfied. The record pointer shall not be changed by an unsuccessful READ RECORD function.

The record to be read is described by the modes below. Six modes are defined:

- **CURRENT**: the current record is read. The record pointer is not affected;

- **ABSOLUTE**: the record given by the record number is read. The record pointer is not affected;

- **NEXT**: the record pointer is incremented before the READ RECORD function is performed and the pointed record is read. If the record pointer has not been previously set within the selected EF, then READ RECORD (next) shall read the first record and set the record pointer to this record.

  If the record pointer addresses the last record in a linear fixed EF, READ RECORD (next) shall not cause the record pointer to be changed, and no data shall be read.

  If the record pointer addresses the last record in a cyclic EF, READ RECORD (next) shall read the first record in this EF and set the record pointer to this EF;

- **PREVIOUS**: the record pointer is decremented before the READ RECORD function is performed and the pointed record is read. If the record pointer has not been previously set within the selected EF, then READ RECORD (previous) shall read the last record and set the record pointer to this record.

  If the record pointer addresses the first record in a linear fixed EF, READ RECORD (previous) shall not cause the record pointer to be changed, and no data shall be read.

  If the record pointer addresses the first record in a cyclic EF, READ RECORD (previous) shall read the last record in this EF and set the record pointer to this record;

- **FIRST**: the record pointer is set to the first record and READ RECORD (first) shall read this record;

- **LAST**: the record pointer is set to the last record and READ RECORD (last) shall read this record.

Input:

- mode, record number (absolute mode only) and the length of the record.

Output:

- the record.

8.6 UPDATE RECORD

This function updates one complete record in the current linear fixed or cyclic EF. This function shall only be performed if the UPDATE access condition for this EF is satisfied. The UPDATE can be considered as a replacement of the relevant record data of the EF by the record data given in the command. The record pointer shall not be changed by an unsuccessful UPDATE RECORD function.

The record to be updated is described by the modes below. Six modes are defined of which only PREVIOUS is allowed for cyclic files:

- **CURRENT**: the current record is updated. The record pointer is not affected;

- **ABSOLUTE**: the record given by the record number is updated. The record pointer is not affected;

- **NEXT**: the record pointer is incremented before the UPDATE RECORD function is performed and the pointed record is updated. If the record pointer has not been previously set within the selected EF, then UPDATE RECORD (next) shall set the record pointer to the first record in this EF and this record shall be updated. If the record pointer addresses the last record, UPDATE RECORD (next) shall not cause the record pointer to be changed, and no record shall be updated;

- **PREVIOUS**: for a linear fixed EF the record pointer is decremented before the UPDATE RECORD (previous) function is performed and the pointed record is updated. If the record pointer has not been previously set within the selected EF, then UPDATE RECORD (previous) shall set the record pointer to the last record in this EF and this record shall be updated. If the record pointer addresses the first record in a linear fixed EF, UPDATE RECORD (previous) shall not cause the record pointer to be changed, and no record shall be updated.

  For a cyclic EF the record containing the oldest data is updated, the record pointer is set to this record and this record becomes record number 1;

- **FIRST**: the record pointer is set to the first record and UPDATE RECORD (first) shall update this record;

- **LAST**: the record pointer is set to the last record and UPDATE RECORD (last) shall update this record.

Input:

- mode, record number (absolute mode only) and the length of the record;

- the data used for updating the record.

Output:

- none.

8.7 SEEK

This function searches through the current linear fixed EF to find a record containing the given pattern. This function shall only be performed if the READ access condition for this EF is satisfied. Two types of SEEK are defined:

- **Type 1**: the record pointer is set to the record containing the pattern, no output is available.

- **Type 2**: the record pointer is set to the record containing the pattern, the output is the record number.

SEEK is performed with the length of the pattern starting in each record at the point indicated in the offset. If the parameter P3 indicates a pattern-length greater than the record length, the DAM shall send the status information "incorrect parameter P3". If the record length is shorter than pattern length and offset, then the response is "offset out of range". The DAM shall be able to accept any pattern length from 1 to 16 bytes inclusive.

Four modes are defined:

- from the beginning forwards;

- from the end backwards;

- from the next location forwards;

- from the previous location backwards.

If the record pointer has not been previously set (its status is undefined) within the selected linear fixed EF, then the search begins:

- with the first record in the case of SEEK from the next location forwards; or

- with the last record in the case of SEEK from the previous location backwards.

After a successful SEEK, the record pointer is set to the record in which the pattern was found. The record pointer shall not be changed by an unsuccessful SEEK function.

Input:

- type and mode;

- pattern;

- length of the pattern and offset.

Output:

- type 1: none;

- type 2: record number.

8.8 INCREASE

This function adds the value given by the PE to the value of the last increased/updated record of the current cyclic EF and stores the result into the oldest record. The record pointer is set to this record and this record becomes record number 1. This function shall be used only if this EF has an INCREASE access condition assigned and this condition is fulfilled. The DAM shall not perform the increase if the result would exceed the maximum value of the record (represented by all bytes set to "FF").

Input:

- the value to be added.

Output:

- value of the increased record;

- value which has been added.

8.9 VERIFY CHV

This function verifies the CHV presented by the PE for the current Directory (see clause 7) by comparing it with the one stored in the relevant EF_{CHV}. The verification process is subject to the following conditions being fulfilled:

- CHV is not disabled;

- CHV is not blocked.

If the access condition for a function to be performed on a file is CHV1 or CHV2, then a successful verification of the relevant CHV is required prior to the use of the function on this file unless the CHV is disabled.

If the CHV presented is correct, the number of remaining CHV attempts for that CHV shall be reset to its initial value 3.

If the CHV presented is false, the number of remaining CHV attempts for that CHV shall be decremented. After 3 consecutive false CHV presentations, not necessarily in the same card session, the respective CHV shall be blocked and the access condition can never be fulfilled until the UNBLOCK CHV function has been successfully performed on the respective CHV.

Input:

- indication CHV1/CHV2, CHV.

Output:

- none.

8.10 CHANGE CHV

This function assigns a new value to the relevant CHV, subject to the following conditions being fulfilled:

- CHV is not disabled;

- CHV is not blocked.

The old and new CHV shall be presented.

If the old CHV presented is correct, the number of remaining CHV attempts for that CHV shall be reset to its initial value 3 and the new value for the CHV becomes valid.

If the old CHV presented is false, the number of remaining CHV attempts for that CHV shall be decremented and the value of the CHV is unchanged. After 3 consecutive false CHV presentations, not necessarily in the same card session, the respective CHV shall be blocked and the access condition can never be fulfilled until the UNBLOCK CHV function has been performed successfully on the respective CHV.

Input:

- indication CHV1/CHV2, old CHV, new CHV.

Output:

- none.

8.11 DISABLE CHV

The successful execution of this function has the effect that files protected by the respective CHV are now accessible as if they were marked "ALWAYS". The function DISABLE CHV shall not be executed by the DAM when the respective CHV is already disabled or blocked.

If the CHV presented is correct, the number of remaining CHV attempts for that CHV shall be reset to its initial value 3 and that CHV shall be disabled.

If the CHV presented is false, the number of remaining CHV attempts for that CHV shall be decremented and that CHV remains enabled. After 3 consecutive false CHV presentations, not necessarily in the same card session, the respective CHV shall be blocked and the access condition can never be fulfilled until the UNBLOCK CHV function has been successfully performed on the respective CHV.

Input:

- indication CHV1/CHV2, CHV.

Output:

- none.

8.12 ENABLE CHV

This function is the reverse function of DISABLE CHV. The function ENABLE CHV shall not be executed by the DAM when the respective CHV is already enabled or blocked.

If the CHV presented is correct, the number of remaining CHV attempts for that CHV shall be reset to its initial value 3 and the CHV shall be enabled.

If the CHV presented is false, the number of remaining CHV attempts for that CHV shall be decremented and that CHV remains disabled. After 3 consecutive false CHV presentations, not necessarily in the same card session, the respective CHV shall be blocked and the access condition can never be fulfilled until the UNBLOCK CHV function has been performed successfully on the respective CHV.

Input:

- indication CHV1/CHV2, CHV.

Output:

- none.

8.13 UNBLOCK CHV

This function unblocks a CHV which has been blocked by 3 consecutive wrong CHV presentations. This function may be performed whether or not the relevant CHV is blocked.

If the UNBLOCK CHV presented is correct, the new CHV value, presented together with the UNBLOCK CHV, is stored in the relevant EF_{CHV}, the number of remaining UNBLOCK CHV attempts for that UNBLOCK CHV is reset to its initial value 10 and the number of remaining CHV attempts for that CHV is reset to its initial value 3. After a successful unblocking attempt the CHV is enabled and the relevant access condition level is satisfied.

If the presented UNBLOCK CHV is false, the number of remaining UNBLOCK CHV attempts for that UNBLOCK CHV shall be decremented. After 10 consecutive false UNBLOCK CHV presentations, not necessarily in the same card session, the respective UNBLOCK CHV shall be blocked. A false UNBLOCK CHV shall have no effect on the status of the respective CHV itself.

Input:

- indication CHV1/CHV2, the Unblock CHV, the new CHV.

Output:

- none.

8.14 INVALIDATE

This function invalidates the current EF. After an INVALIDATE function the respective flag in the file status shall be changed accordingly. This function shall only be performed if the INVALIDATE access condition for the current EF is satisfied.

An invalidated file shall no longer be available within the application for any function except for the SELECT and the REHABILITATE functions. The function READ may be performed when "readable when invalidated" is indicated in the file status.

NOTE: By invalidating relevant EFs in a subscription this subscription can be blocked for further use.

Input:

- none.

Output:

- none.

8.15 REHABILITATE

This function rehabilitates the invalidated current EF. After a REHABILITATE function the respective flag in the file status byte shall be changed for the current EF accordingly. This function shall only be performed if the REHABILITATE access condition for the current EF is satisfied.

Input:

- none.

Output:

- none.

8.16 ASK RANDOM

This function requests the DAM to send a random number. This number shall be used in the subsequent FT AUTHENTICATION command.

Input:

- none.

Output:

- RAND_P.

8.17 PT AUTHENTICATION

This function authenticates the selected registration in the DAM to the FT. The DAM runs the specified algorithm using the random numbers, which are contained in the PT AUTHENTICATION command, and the key selected from EF_{UAK} or EF_{AC} to produce an authentication parameter (RES1), a ciphering key (DCK) and, if appropriate, the ZAP value. The obtained values are returned to the PE.

This function shall not be executable unless a DF_{SR} has been selected as the current directory and a successful CHV1 verification procedure has been performed (see subclause 11.3.1).

Input:

- key number, algorithm identifier, INC, RS, RAND_F.

Output:

- RES1;

- if appropriate, DCK and ZAP.

8.18 FT AUTHENTICATION

This function authenticates the FT with respect to the selected registration in the DAM. The ASK RANDOM command shall immediately precede the FT AUTHENTICATION command. The DAM runs the specified algorithm using the random number RAND_P, which it previously generated, the random number RS, which is contained in the FT AUTHENTICATION command, and the key selected from EF_{UAK} or EF_{AC} to produce an authentication parameter (XRES2). The DAM compares the derived value with RES2 which it obtained in the FT AUTHENTICATION command.

The authentication is only successful if the verification by the DAM is positive.

Input:

- key number, algorithm identifier, RS, RES2.

Output:

- none.

8.19 USER AUTHENTICATION

This function authenticates the user to the FT. The DAM runs the specified algorithm using the random numbers, which are contained in the USER AUTHENTICATION command, the key UPI obtained by the PE from the user and the key selected from EF_{UAK} to produce an authentication parameter (RES1) and a ciphering key (DCK). The obtained values are returned to the PE.

Input:

- key number, algorithm identifier, RS, RAND_F, UPI.

Output:

- RES1;

- if appropriate, DCK.

8.20 UAK ALLOCATION

This function performs the initial allocation of a UAK as specified in clause 7. It requires the presence of at least one AC in the DAM and shall only be possible after a successful PT authentication. The DAM runs the specified algorithm using the random number RAND_P, which it previously generated, the random number RS, which is contained in the UAK ALLOCATION command, and the key selected from EF_{AC} to produce an authentication parameter (XRES2) and the UAK. The DAM compares XRES2 with RES2 which it obtained in the UAK ALLOCATION command. If the two values are equal, the UAK is stored in EF_{UAK} under the number identified in the UAK ALLOCATION command.

Input:

- key number, UAK number, algorithm identifier, RS, RES2.

Output:

- none.

9 Description of the commands

This clause states the general principles for mapping the functions described in clause 8 onto Application Protocol Data Units (APDUs) which are used by the transmission protocol.

9.1 Mapping principles

An APDU can be a command APDU or a response APDU.

A command APDU has the following format:

CLA	INS	P1	P2	P3	Data

The response APDU has the following format:

Data	SW1	SW2

An APDU is transported by the T=0 transmission protocol without any change. Other protocols might embed an APDU into their own transport structure (see ISO/IEC 7816-3 [12]).

The bytes have the following meaning:

- CLA is the class of instruction (see ISO/IEC 7816-3 [12]);
- INS is the instruction code (see ISO/IEC 7816-3 [12]) as defined in subclause 9.2 for each command;
- P1, P2, P3 are parameters for the instruction. They are specified in table 8. P3 gives the length of the data element; P3="FF" shall not be used. "FF" is a valid parameter for P1 and P2;
- SW1 and SW2 are the status words indicating the successful or unsuccessful outcome of the command.

For some of the functions described in clause 8 it is necessary for T=0 to use a supplementary transport service command (GET RESPONSE) to obtain output data. For example, the SELECT function needs the following two commands:

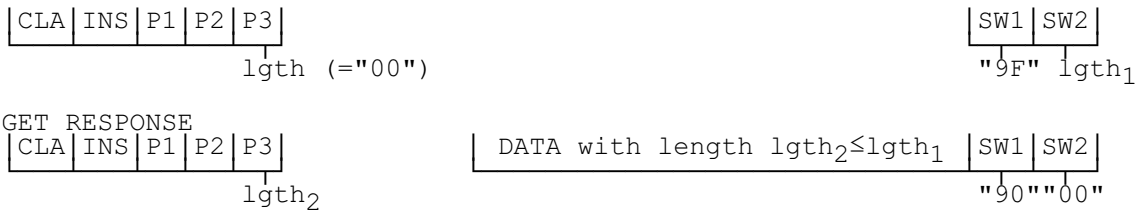
- the first command (SELECT) has both parameters and data serving as input for the function;
- the second command (GET RESPONSE) has a parameter indicating the length of the data to be returned.

If the length of the response data is not known beforehand, then its correct length may be obtained by applying the first command and interpreting the status word. SW1 shall be "9F" and SW2 shall give the total length of the data. Other status words may be present in case of an error. The various cases are:

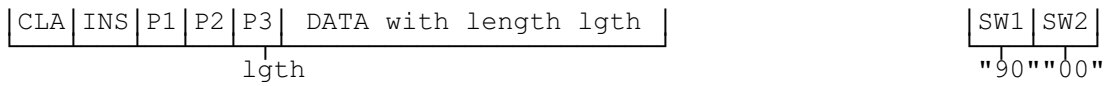
Case 1: No input / No output

Case 2: No input / Output of known length

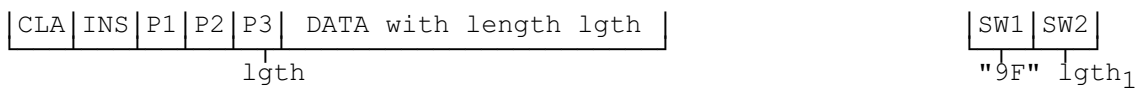
Case 3: No Input / Output of unknown length



Case 4: Input / No output



Case 5: Input / Output of known or unknown length



For cases 3 and 5, when Status Word 1 (SW1)/Status Word 2 (SW2) indicates there is response data (i.e. SW1/SW2 = "9FXX"), then, if the ME requires to get this response data, it shall send a GET RESPONSE command as described in the relevant case above.

The DAM shall accept class bytes A0-A3. For the use of class bytes A1-A3 see ISO/IEC DIS 7816-4 (annex C [1]). The PE shall use the class byte A0 as a default value.

9.2 Coding of the commands

The following table gives the coding of the commands. The direction of the data is indicated by (S) and (R), where (S) stands for data sent by the PE while (R) stands for data received by the PE. Offset for the commands READ BINARY and UPDATE BINARY is coded on 2 bytes where P1 gives the high order byte and P2 gives the low order byte. "00 00" means no offset and reading/updating starts with the first byte while an offset of "00 01" means that reading/updating starts with the second byte, etc. Offset for the command SEEK is coded on one byte.

Table 8: Coding of the commands

Command	INS	P1	P2	P3	S/R
SELECT	"A4"	"00" or "04"	"00"	"02" or lgth	S/R
STATUS	"F2"	"00"	"00"	lgth	R
READ BINARY	"B0"	offset high	offset low	lgth	R
UPDATE BINARY	"D6"	offset high	offset low	lgth	S
READ RECORD	"B2"	rec No.	mode	lgth	R
UPDATE RECORD	"DC"	rec No.	mode	lgth	S
SEEK	"A2"	offset	type/mode	lgth	S/R
INCREASE	"32"	"00"	"00"	"03"	S/R
VERIFY CHV	"20"	"00"	CHV No.	"08"	S
CHANGE CHV	"24"	"00"	CHV No.	"10"	S
DISABLE CHV	"26"	"00"	CHV No.	"08"	S
ENABLE CHV	"28"	"00"	CHV No.	"08"	S
UNBLOCK CHV	"2C"	"00"	CHV No.	"10"	S
INVALIDATE	"04"	"00"	"00"	"00"	-
REHABILITATE	"44"	"00"	"00"	"00"	-
ASK RANDOM	"84"	"00"	"00"	"08"	R
PT AUTHENTICATION	"50"	"00"	"00"	"13"	S/R
FT AUTHENTICATION	"52"	"00"	"00"	"0E"	S
USER AUTHENTICATION	"54"	"00"	"00"	"16"	S/R
UAK ALLOCATION	"56"	"00"	"00"	"0F"	S
GET RESPONSE	"C0"	"00"	"00"	lgth	R

In addition to the instruction codes specified in table 8 the instruction codes "1X", with X ranging from 0 to 8 (X even), are reserved for the enhancement of the DECT application and for use in DF_{DECT}.

Definitions and codings used in the response parameters/data of the commands are given in subclause 9.3.

9.2.1 SELECT

To select DF_{DECT} using direct selection the SELECT command has the following structure:

COMMAND	CLASS	INS	P1	P2	P3
SELECT		"A4"	"04"	"00"	lgth

Command parameters/data:

Byte(s)	Description	Length
1 - X	Application identifier	X

NOTE: The application identifier and its length are subject to the assignment by ISO/IEC [13]. They will be incorporated into this standard as soon as they have been issued by the registration authority.

To select a file according to subclause 6.6 the SELECT command has the following structure:

COMMAND	CLASS	INS	P1	P2	P3
SELECT		"A4"	"00"	"00"	"02"

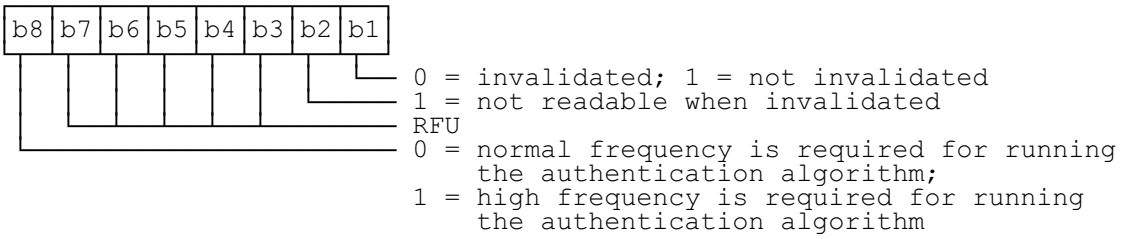
Command parameters/data:

Byte(s)	Description	Length
1 - 2	File ID	2

Response parameters/data in case of an MF or DF:

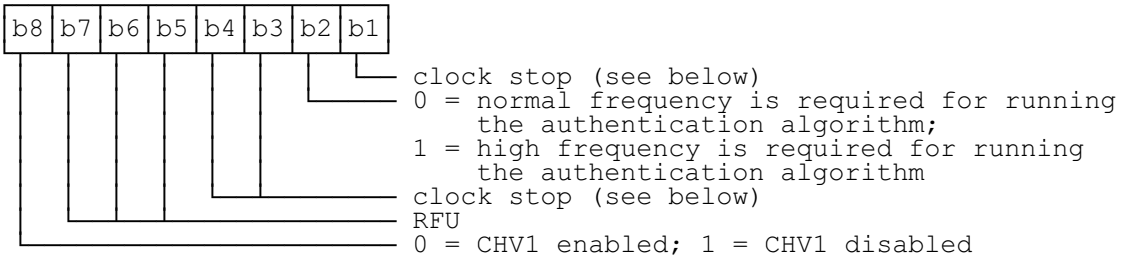
Byte(s)	Description	Length
1 - 2	Reserved for Future Use (RFU)	2
3 - 4	Total amount of memory of the selected directory which is not allocated to any of the DFs or EFs under the selected directory	2
5 - 6	File ID	2
7	Type of file (see subclause 9.3)	1
8	RFU	1
9 - 11	Access conditions (see subclause 9.3)	3
12	File status (see detail 1 below)	1
13	Length of the following data (byte 14 to the end)	1
14	Current Directory characteristics (see detail 2 below)	1
15	Number of DFs which are a direct child of the current Directory	1
16	Number of EFs which are a direct child of the current Directory	1
17	Number of CHVs, UNBLOCK CHVs and administrative codes	1
18	RFU	1
19	CHV1 status (see detail 3 below)	1
20	UNBLOCK CHV1 status (see detail 3 below)	1
21	CHV2 status (see detail 3 below)	1
22	UNBLOCK CHV2 status (see detail 3 below)	1

Detail 1 (file status, byte 12):



The option b2=0 (readable when invalidated) is not used in the DECT application.

Detail 2 (directory characteristics, byte 14):



NOTE 1: The values for normal and high frequency required by the authentication algorithm depend on the specific access profile and the implementation of the algorithm in the DAM. The values are identical for bytes 12 and 14.

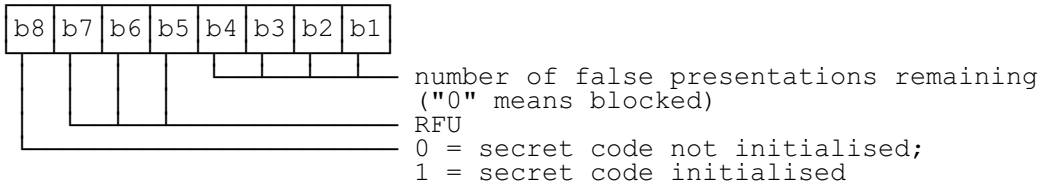
The conditions for stopping the clock shall be identical to those of byte 1 of EF_{ICC} with the following order for the bits:

Bit b1	Bit b3	Bit b4	
1	0	0	clock stop allowed, no preferred level
1	1	0	clock stop allowed, high level preferred
1	0	1	clock stop allowed, low level preferred
0	0	0	clock stop not allowed
0	1	0	clock stop only allowed on high level
0	0	1	clock stop only allowed on low level

If bit b1 (column 1) is coded 1, stopping the clock is allowed at high or low level. In this case columns 2 (bit b3) and 3 (bit b4) give information about the preferred level (high or low, resp.) at which the clock may be stopped.

If bit b1 is coded 0 the clock may be stopped only if the mandatory condition in column 2 (b3=1, i.e. stop at high level) or column 3 (b4=1, i.e. stop at low level) is fulfilled. If all 3 bits are coded 0, then the clock shall not be stopped.

Detail 3 (status byte of a secret code; bytes 19-22):



NOTE 2: Both CHV and UNBLOCK CHV are initialised when bit 1 of the CHV ACTIVATION byte in the corresponding EF_{CHV} is set to 1.

Response parameters/data in case of an EF (except for EF_{CHV}):

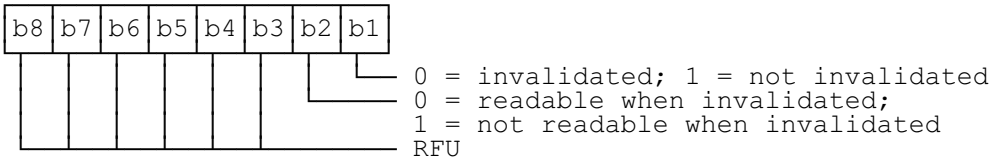
Byte(s)	Description	Length
1 - 2	RFU	2
3 - 4	File size (for transparent EF: the length of the body part of the EF) (for linear fixed or cyclic EF: record length * number of records of the EF)	2
5 - 6	File ID	2
7	Type of file (see subclause 9.3)	1
8	See detail 4 below	1
9 - 11	Access conditions (see subclause 9.3)	3
12	File status (see detail 5 below)	1
13	Length of the following data (byte 14 to the end)	1
14	Structure of EF (see subclause 9.3)	1
15	Length of a record (see detail 6 below)	

NOTE 3: Bytes 16 and following are RFU.

Detail 4 (coding of byte 8):

For transparent and linear fixed EFs this byte is RFU. For a cyclic EF bit 7 of this byte is set to 1, while all other bits are RFU.

Detail 5 (file status, byte 12):



Detail 6 (length of a record, byte 15):

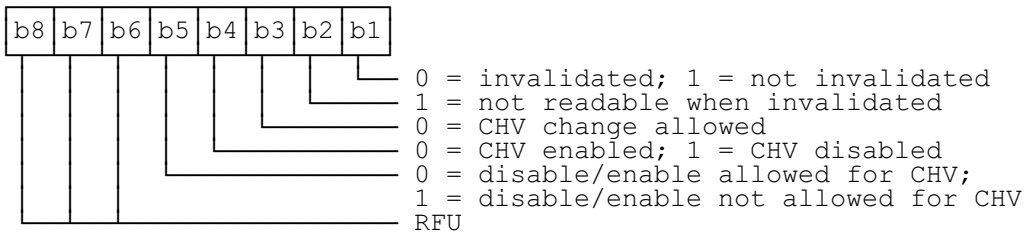
For cyclic and linear fixed EFs this byte denotes the length of a record. For a transparent EF, this byte shall be coded "00", if this byte is sent by the DAM.

Response parameters/data in case of an EF_{CHV}:

Byte(s)	Description	Length
1 - 2	RFU	2
3 - 4	Number of bytes allocated to the body	2
5 - 6	File ID	2
7	Type of file (see subclause 9.3)	1
8	RFU	1
9 - 11	Access conditions (see subclause 9.3)	3
12	File status (see detail 7 below)	1
13	Length of the following data (byte 14 to the end)	1
14	Structure of EF (see subclause 9.3)	1
15	Number of remaining CHV attempts	1
16	Type of user identification (see below)	1
17	Way to present the CHV (see below)	1
18	Coded "FF" (see below)	1
19	Number of remaining UNBLOCKING CHV attempts	1
20	Coded "FF" (see below)	1

NOTE 4: Bytes 21 and following are RFU.

Detail 7 (file status, byte 12):



The coding b2=0 is not allowed. The option b3=1 (CHV change not allowed) is not used in the DECT application.

Byte 16 shall be coded "01", which means CHV verification.

Byte 17 shall be coded as byte 2 of the respective EF_{CHV} (see clause 10).

Bytes 18 and 20 shall be coded "FF" and shall not be interpreted by a PE.

9.2.2 STATUS

COMMAND	CLASS	INS	P1	P2	P3
STATUS		"F2"	"00"	"00"	lgth

The response parameters/data are identical to the response parameter/data of the SELECT command in case of an MF or DF.

9.2.3 READ BINARY

COMMAND	CLASS	INS	P1	P2	P3
READ BINARY		"B0"	offset	offset	lgth

Offset is coded on 2 bytes, where P1 gives the high order byte and P2 the low order byte.

Response parameters/data:

Byte(s)	Description	Length
1-lgth	Data to be read	lgth

9.2.4 UPDATE BINARY

COMMAND	CLASS	INS	P1	P2	P3
UPDATE BINARY		"D6"	offset	offset	lgth

Offset is coded on 2 bytes, where P1 gives the high order byte and P2 the low order byte.

Command parameters/data:

Byte(s)	Description	Length
1 - lgth	Data	lgth

9.2.5 READ RECORD

COMMAND	CLASS	INS	P1	P2	P3
READ RECORD		"B2"	Rec. No.	Mode	lgth

Parameter P2 specifies the mode:

- "00" = first record;
- "01" = last record;
- "02" = next record;
- "03" = previous record;
- "04" = absolute mode/current mode; the record number is given in P1 with P1="00" denoting the current record.

For the modes "first", "last", "next" and "previous" P1 has no significance and shall be set to "00" by the PE.

Response parameters/data:

Byte(s)	Description	Length
1-lgth	The data of the record	lgth

9.2.6 UPDATE RECORD

COMMAND	CLASS	INS	P1	P2	P3
UPDATE RECORD		"DC"	Rec. No.	Mode	lgth

Parameter P2 specifies the mode:

- "00" = first record;
- "01" = last record;
- "02" = next record;
- "03" = previous record;
- "04" = absolute mode/current mode; the record number is given in P1 with P1="00" denoting the current record.

For the modes "first", "last", "next" and "previous" P1 has no significance and shall be set to "00" by the PE.

Command parameters/data:

Byte(s)	Description	Length
1 - lgth	Data	lgth

9.2.7 SEEK

COMMAND	CLASS	INS	P1	P2	P3
SEEK		"A2"	offset	type/mode	lgth

The parameter P1 specifies the offset (in bytes) within the record. For instance, "00" means no offset and the search starts with the first byte, while an offset of "01" means that the search starts with the second byte, ...

Parameter P2 specifies type and mode:

- "x0" = forwards from the beginning;
- "x1" = backwards from the end;
- "x2" = forwards from the next record;
- "x3" = backwards from the record preceding the present record;

with x="0" specifies type 1 and x="1" specifies type 2 of the SEEK command.

Command parameters/data:

Byte(s)	Description	Length
1-lgth	Pattern	lgth

There are no response parameters/data for a type 1 SEEK. A type 2 SEEK returns the following response parameters/data:

Byte(s)	Description	Length
1	Record number	1

9.2.8 INCREASE

COMMAND	CLASS	INS	P1	P2	P3
INCREASE		"32"	"00"	"00"	"03"

Command parameters/data:

Byte(s)	Description	Length
1 - 3	Value to be added	3

Response parameters/data:

Byte(s)	Description	Length
1 - y	Value of the increased record	y
y+1 - y+3	Value which has been added	3

NOTE: The INCREASE command is only specified in this ETS for the use with specific access profiles such as GSM or services specified as amendments to this ETS. It need only be implemented in the DAM if the DAM supports such a profile or service.

9.2.9 VERIFY CHV

COMMAND	CLASS	INS	P1	P2	P3
VERIFY CHV		"20"	"00"	CHV No.	"08"

Parameter P2 specifies the CHV:
- "01" = CHV1;

- "02" = CHV2.

Command parameters/data:

Byte(s)	Description	Length
1 - 8	CHV value	8

9.2.10 CHANGE CHV

COMMAND	CLASS	INS	P1	P2	P3
CHANGE CHV		"24"	"00"	CHV No.	"10"

Parameter P2 specifies the CHV:

- "01" = CHV1;

- "02" = CHV2.

Command parameters/data:

Byte(s)	Description	Length
1 - 8	Old CHV value	8
9 - 16	New CHV value	8

9.2.11 DISABLE CHV

COMMAND	CLASS	INS	P1	P2	P3
DISABLE CHV		"26"	"00"	CHV No.	"08"

Parameter P2 specifies the CHV:

- "01" = CHV1;
- "02" = CHV2.

Command parameters/data:

Byte(s)	Description	Length
1 - 8	CHV value	8

9.2.12 ENABLE CHV

COMMAND	CLASS	INS	P1	P2	P3
ENABLE CHV		"28"	"00"	CHV No.	"08"

Parameter P2 specifies the CHV:

- "01" = CHV1;
- "02" = CHV2.

Command parameters/data:

Byte(s)	Description	Length
1 - 8	CHV value	8

9.2.13 UNBLOCK CHV

COMMAND	CLASS	INS	P1	P2	P3
UNBLOCK CHV		"2C"	"00"	CHV No.	"10"

Parameter P2 specifies the CHV:

- "01" = CHV1;
- "02" = CHV2.

Command parameters/data:

Byte(s)	Description	Length
1 - 8	UNBLOCK CHV value	8
9 - 16	New CHV value	8

9.2.14 INVALIDATE

COMMAND	CLASS	INS	P1	P2	P3
INVALIDATE		"04"	"00"	"00"	"00"

9.2.15 REHABILITATE

COMMAND	CLASS	INS	P1	P2	P3
REHABILITATE		"44"	"00"	"00"	"00"

9.2.16 ASK RANDOM

COMMAND	CLASS	INS	P1	P2	P3
ASK RANDOM		"84"	"00"	"00"	"08"

Response parameters/data:

Byte(s)	Description	Length
1 - 8	RAND_P	8

9.2.17 PT AUTHENTICATION

COMMAND	CLASS	INS	P1	P2	P3
PT AUT		"50"	"00"	"00"	"13"

Command parameters/data:

Byte(s)	Description	Length
1	Key No.	1
2	Algorithm ID	1
3	INC	1
4 - 11	RS	8
12 - 19	RAND_F	8

Response parameters/data:

Byte(s)	Description	Length
1 - 4	RES1	4
5 - 12	DCK	8
13	ZAP	1

9.2.18 FT AUTHENTICATION

COMMAND	CLASS	INS	P1	P2	P3
FT AUT		"52"	"00"	"00"	"0E"

Command parameters/data:

Byte(s)	Description	Length
1	Key No.	1
2	Algorithm ID	1
3 - 10	RS	8
11 - 14	RES2	4

9.2.19 USER AUTHENTICATION

COMMAND	CLASS	INS	P1	P2	P3
USER AUTHENTICATION		"54"	"00"	"00"	"16"

Command parameters/data:

Byte(s)	Description	Length
1	Key No.	1
2	Algorithm ID	1
3 - 10	RS	8
11 - 18	RAND_F	8
19 - 22	UPI	4

Response parameters/data:

Byte(s)	Description	Length
1 - 4	RES1	4
5 - 12	DCK	8

9.2.20 UAK ALLOCATION

COMMAND	CLASS	INS	P1	P2	P3
UAK ALLOCATION		"56"	"00"	"00"	"0F"

Command parameters/data:

Byte(s)	Description	Length
1	Key No.	1
2	UAK No.	1
3	Algorithm ID	1
4 - 11	RS	8
12 - 15	RES2	4

9.2.21 GET RESPONSE

COMMAND	CLASS	INS	P1	P2	P3
GET RESPONSE		"C0"	"00"	"00"	lgth

The response data depends on the preceding command. Response data is available after the commands SELECT, SEEK (type 2), INCREASE, ASK RANDOM, PT AUTHENTICATION and USER AUTHENTICATION. If the command GET RESPONSE is executed, it is required that it is executed immediately after the command it is related to (no other command shall come between the command/response pair and the command GET RESPONSE). If the sequence is not respected, the DAM shall send the Status Information "technical problem with no diagnostic given" as a reaction to the GET RESPONSE.

Since the MF is implicitly selected after the activation of the DAM, GET RESPONSE is allowed as the first command after activation.

The response data itself is defined in the subclause for the corresponding command.

9.3 Definitions and coding

The following definitions and coding are used in the response parameters/data of the commands.

Coding

Each byte is represented by bits b8, b7, ... , b1, where b8 is the most significant bit and b1 is the least significant bit. In each representation the leftmost bit is the most significant one.

RFU

In a DECT specific card all bytes which are RFU shall be set to "00" and RFU bits to 0. Where the DECT application exists on a multi-application card or is built on a generic telecommunications card then other values may apply. The values will be defined in the appropriate specifications for such cards. These bytes and bits shall not be interpreted by a PE in a DECT session.

Structure of file

- "00" = transparent;
- "01" = linear fixed;
- "03" = cyclic.

Type of file

- "00" = RFU;
- "01" = MF;
- "02" = DF;
- "03" = DF with ASC;
- "04" = EF.

NOTE: The type of file "03" is used to denote DF_{DECT} in a multi-application card as application specific commands are used in the DECT application.

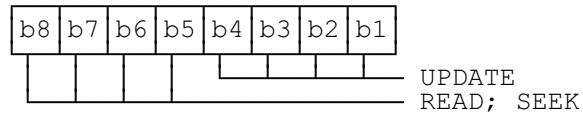
Coding of access conditions

The access conditions for the commands are coded on the bytes 9, 10 and 11 of the response data of the SELECT command. Each condition is coded on 4 bits as shown in table 9.

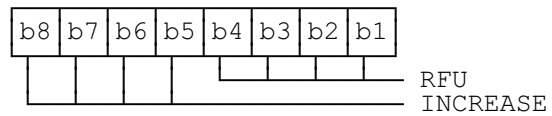
Table 9: Access conditions

ALW	"0"
CHV1	"1"
CHV2	"2"
RFU	"3"
FT-AUT	"4"
RFU	"5"
RFU	"6"
RFU	"7"
CHV1 and FT-AUT	"8"
CHV2 and FT-AUT	"9"
ADM	"A"
ADM	"B"
ADM	"C"
ADM	"D"
ADM	"E"
NEV	"F"

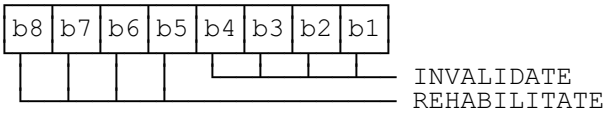
Coding of byte 9:



Coding of byte 10:



Coding of byte 11:



9.4 Status conditions returned by the DAM

This subclause specifies the coding of the status words SW1 and SW2.

9.4.1 Responses to commands which are correctly executed

SW1	SW2	Description
"90"	"00"	- normal ending of the command
"9F"	"XX"	- length "XX" of the response data

9.4.2 Memory management

SW1	SW2	Error description
"92"	"0X"	- update successful but after using an internal retry routine "X" times
"92"	"40"	- memory problem

9.4.3 Referencing management

SW1	SW2	Error description
"94"	"00"	- no EF selected
"94"	"02"	- out of range (invalid address)
"94"	"04"	- file ID not found - pattern not found
"94"	"08"	- file type is inconsistent with the command

9.4.4 Security management

SW1	SW2	Error description
"98"	"02"	- no CHV initialised
"98"	"04"	- access condition not fulfilled - unsuccessful CHV verification, at least one attempt left - unsuccessful UNBLOCK CHV verification, at least one attempt left - authentication failed
"98"	"08"	- in contradiction with CHV status
"98"	"10"	- in contradiction with invalidation status
"98"	"35"	- ASK RANDOM has not been executed
"98"	"40"	- unsuccessful CHV verification, no attempt left - unsuccessful UNBLOCK CHV verification, no attempt left - CHV blocked - UNBLOCK CHV blocked
"98"	"50"	- increase cannot be performed, maximum value reached

9.4.5 Application independent errors

SW1	SW2	Error description
"67"	"XX"	- incorrect parameter P3 (see NOTE)
"6B"	"XX"#	- incorrect parameter P1 or P2 (see ##)
"6D"	"XX"#	- unknown instruction code given in the command
"6E"	"XX"#	- wrong instruction class given in the command
"6F"	"XX"#	- technical problem with no diagnostic given

These values of "XX" are specified by ISO/IEC; at present the default value "XX"="00" is the only one defined.

When the error in P1 or P2 is caused by the addressed record being out of range, then the return code "94 02" shall be used.

NOTE: "XX" gives the correct length or states that no additional information is given ("XX" = "00").

9.4.6 Commands versus possible status responses

Table 10 shows for each command the possible status conditions returned (marked by an asterisk *).

Table 10: Commands and responses

	OK		Mem Sta		Refer. Status				Security Status							Application Independent Errors				
Commands	90 00	9F XX	92 0X	92 40	94 00	94 02	94 04	94 08	98 02	98 04	98 08	98 10	98 35	98 40	98 50	67 XX	6B XX	6D XX	6E XX	6F XX
SELECT		*		*			*									*	*		*	*
STATUS	*			*												*	*		*	*
UPDATE BINARY	*		*	*	*			*		*		*				*	*		*	*
UPDATE RECORD	*		*	*	*	*		*		*		*				*	*		*	*
READ BINARY	*			*	*			*		*		*				*	*		*	*
READ RECORD	*			*	*	*		*		*		*				*	*		*	*
SEEK	*	*		*	*		*	*		*		*				*	*		*	*
INCREASE		*	*	*	*			*		*		*			*	*	*		*	*
VERIFY CHV	*		*	*					*	*	*			*		*	*		*	*
CHANGE CHV	*		*	*					*	*	*			*		*	*		*	*
DISABLE CHV	*		*	*					*	*	*			*		*	*		*	*
ENABLE CHV	*		*	*					*	*	*			*		*	*		*	*
UNBLOCK CHV	*		*	*					*	*	*			*		*	*		*	*
INVALIDATE	*		*	*	*					*		*				*	*		*	*
REHABILITATE	*		*	*	*					*		*				*	*		*	*
ASK RANDOM	*															*	*		*	*
PT AUTHENTICATION		*	*	*				*		*				*		*	*		*	*
FT AUTHENTICATION	*			*				*		*			*			*	*		*	*
USER AUTHENTICATION		*		*				*		*						*	*		*	*
UAK ALLOCATION	*		*	*				*								*	*		*	*
GET RESPONSE	*			*												*	*		*	*

10 Contents of the EFs

This clause specifies the EFs needed for a DECT session defining access conditions, data items and coding. A data item is a part of an EF or record which represents a complete logical entity, e.g. the PUT in EF_{IPUI}.

The coding of the data items is specified for each EF. If the coding is in ASCII it shall be according to ISO 8859-1 [14]. Each byte is represented by bits b8, b7, ..., b1, where b8 is the most significant bit and b1 is the least significant bit of the byte. EFs, records or data items having an unassigned value, or, which during the DECT session, are cleared by the PE, shall have their bytes and bits set to "FF" and 1, respectively. After the administrative phase all data items shall have a defined value or have their bits set to 1.

EFs are mandatory (M) or optional (O). The file size of an optional EF may be zero. All implemented EFs with a file size greater than zero shall contain all mandatory data items. Optional data items may either be filled with "F" or, if located at the end of an EF, need not exist.

The EFs are listed for each level according to their identification numbers in ascending order. For an overview containing all files see the following figure.

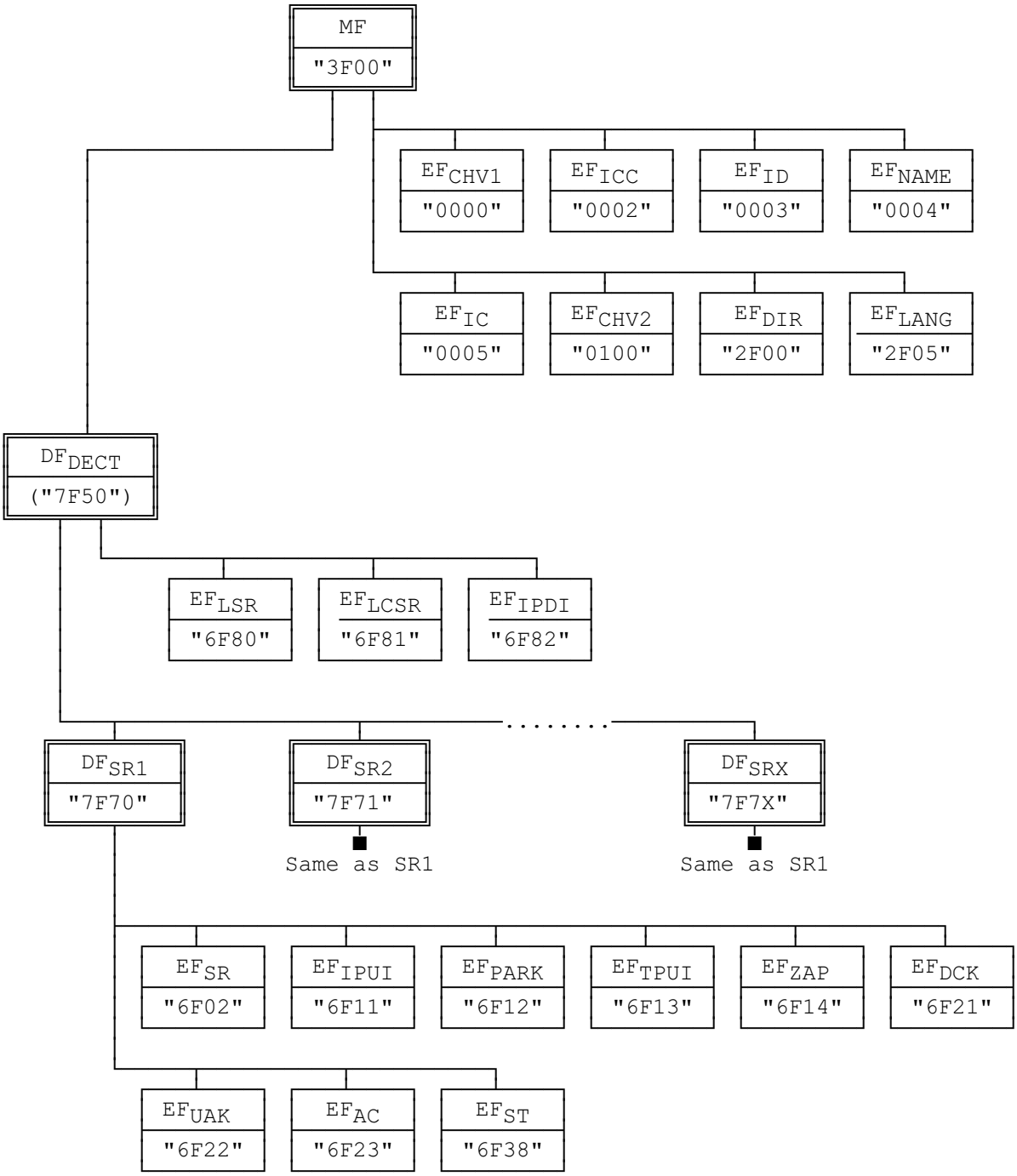


Figure 7: File identifiers and directory structure of a DAM

NOTE: For the use of the identifier "7F50" for DF_{DECT} see subclause 6.3. For conditions on CHV2 see subclause 7.6.

10.1 Contents of the EFs at the MF level

All EFs at MF level are based on the respective Elementary Files specified in prEN 726-3 [15]. They are under the responsibility of the card issuer.

10.1.1 EF_{ICC}

EF_{ICC} provides general information about manufacturing of the IC card. This EF has been incorporated into this ETS as it is a mandatory file in prEN 726-3 [15]. As long as the relevant registers have not been established and data items need to be clarified, bytes 2 to 15 and bytes 16 to 17 (if present) shall be coded "FF".

Identifier: "0002"	Structure: transparent	Mandatory
Access conditions: READ ALWAYS UPDATE NEVER INVALIDATE ADM REHABILITATE ADM		

Byte(s)	Description	M/O	Length
1	Clock stop mode	M	1 byte
2 - 5	IC card serial number	M	4 bytes
6 – 9	IC card manufacturing references	M	4 bytes
10	Card personaliser ID	M	1 byte
11 - 15	Embedder/IC assembler ID	M	5 bytes
16 - 17	IC identifier	O	2 bytes
18	Card Profile	O	1 byte
19	Type of selection	O	1 byte

1) Clock stop mode:

- purpose:
  - to indicate the conditions for stopping the clock. The level refers to the physical level at which the clock shall or should be stopped;
- contents and coding:
  - bits b4 to b8 shall be set to "1". Bits b1, b2 and b3 shall be coded according to the following table.

Table 11: Coding of clock stop

Bit 3	Bit 2	Bit 1	
1	0	0	Clock stop allowed, no preferred level
1	1	0	Clock stop allowed, high level preferred
1	0	1	Clock stop allowed, low level preferred
0	0	0	Clock stop not allowed
0	1	0	Clock stop only allowed on high level
0	0	1	Clock stop only allowed on low level

For additional information see subclause 9.2.1.

2) IC card serial number:

- contents: IC card serial number;
- purpose: to identify the card;
- coding: hexadecimal.

3) IC card manufacturing references:

- contents: IC card manufacturer identifier and related information;
- purpose: to identify the card manufacturer and give related information;
- coding: hexadecimal.

4) Card personaliser ID:

- contents: card personaliser ID as defined by the card issuer;
- purpose: to identify the personaliser of the card;
- coding: hexadecimal.

5) Embedder/IC assembler identifier:

- contents: 5 bytes in the form CCEEA. CC = alphabetic country code of the embedder, as defined in ISO 3166 [7]; EE = 2 alphanumeric characters based on the name of the embedder (there should be a registry at the national level); A = 1 alphanumeric character for other purposes, e.g. to identify the IC assembler;
- purpose: to identify the organisation which combines the IC assembly and the plastic cards;
- coding: hexadecimal.

6) IC identifier:

- contents: IC and IC manufacturer identifiers;
- purpose: to identify the IC of the card;
- coding: hexadecimal.

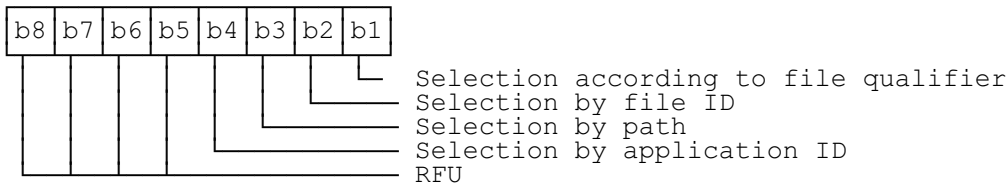
7) Card profile:

- contents: 99 (if coded);
- purpose: indication of the profile according to prEN 726-3 [15];
- coding: Binary Coded Decimal (BCD).

8) Type of selection:

- contents: Selection methods supported by the DAM;
- purpose: indication of the type(s) of selection according to prEN 726-3 [15].

Coding of byte 19:



A type of selection is not supported by the DAM if the corresponding bit is coded "0", while the coding "1" means that the type of selection is supported. The DAM shall support at least "Selection by file ID" (use of EF_{DIR}) or "Selection by application ID" (direct selection), while the PE shall support both these methods.

10.1.2 EF_{ID}

EF_{ID} provides a unique identification number for the card and conveys other card issuing information.

Identifier: "0003"	Structure: transparent	Mandatory

Access conditions:

READ	ALWAYS
UPDATE	NEVER
INVALIDATE	ADM
REHABILITATE	ADM

Byte(s)	Description	M/O	Length
1 — 10	Identification number	M	10 bytes
11 — 13	Date of activation	O	3 bytes
14 — 16	Card expiry date	O	3 bytes
17	Card sequence number	O	1 byte
18 — 19	Country code	O	2 bytes

1) Identification number:

- contents: up to 19 numeric digits, coded according to CCITT Recommendation E.118 [5];
- purpose: card identification number;
- coding: BCD, left justified and padded with "F".

NOTE: If this EF only contains the identification number, then its structure is identical to the Elementary File EF_{ICCID} with identifier "2FE2" of GSM 11.11 [16] except for the digits within a byte being transposed.

2) Date of activation of the MF:

- contents: 6 numeric digits, YYMMDD;
- purpose: to define the date of the activation of the MF;
- coding: BCD.

3) Card expiry date:

- contents: 6 numeric digits, YYMMDD;
- purpose: to state the end of the validity period;
- coding: BCD.

4) Card sequence number:

- contents: sequence number;
- purpose: the sequence number may be needed if a user has more than one card with the same identification number or if a card is replaced by a new one;
- coding: BCD.

5) Country code:

- contents: country code, 3 numeric digits according to ISO 3166 [7];
- purpose: this information element is only present for being compatible with banking cards;
- coding: BCD, right padded with "F".

10.1.3 EF_{NAME}

EF_{NAME} provides information about the user.

Identifier: "0004"	Structure: transparent		Optional
Access conditions: READ ADM UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Byte(s)	Description	M/O	Length
1 – X	Card holder name	O	X bytes

1) Card holder name:

- contents: card holder name and/or related information;
- coding: ASCII, left justified and right padded with "FF".

10.1.4 EF_{IC}

EF_{IC} provides chip related information. This EF has been incorporated into this specification as it is specified in prEN 726-3 [15]. As long as the relevant data items need to be clarified this EF shall not be contained in the DAM.

Identifier: "0005"	Structure: transparent		Optional
Access conditions: READ ALWAYS UPDATE NEVER INVALIDATE NEVER REHABILITATE NEVER			
Byte(s)	Description	M/O	Length
1 – 4	IC serial number	M	4 bytes
5 - 8	IC manufacturing references	M	4 bytes

1) IC serial number:

- contents: IC serial number;
- purpose: to identify the chip;
- coding: hexadecimal.

2) IC manufacturing references:

- contents: chip manufacturer identifier and fabrication elements;
- purpose: to identify the chip manufacturer and related information (date, site of fabrication);
- coding: hexadecimal.

10.1.5 EF_{DIR}

EF_{DIR} provides the path for selecting the DECT application and, in the case of a multi-application card, paths for other application directories.

Identifier: "2F00"	Structure: transparent	Optional

Access conditions:

READ	ADM
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Byte(s)	Description	M/O	Length
1	Application identifier tag	M	1 byte
2	Application identifier length	M	1 byte
3 - X	Application identifier	M	5-16 bytes
	Application label tag	O	1 byte
	Application label length	O	1 byte
	Application label (verbal description)	O	0-16 bytes
	Path tag	M	1 byte
	Path length	M	1 byte
	Path	M	X byte

1) Application identifier tag:

- purpose: to mark the beginning of the data belonging to an application (within the EF);
- coding: "4F".

2) Application identifier length:

- contents: number of bytes of the application identifier beginning in byte 3;
- coding: hexadecimal.

3) Application identifier:

- contents: application identifier in accordance with ISO/IEC 7816-5 [13];
- purpose: to allow the PE to verify that the DECT application is contained in the (multi-application) card;
- coding: hexadecimal.

4) Application label tag:

- purpose: to mark the application label within the applicational data;
- coding: "50".

5) Application label length:

- contents: number of bytes of the application label;
- coding: hexadecimal;

6) Application label:

- contents: optional description of the application;
- coding: according to ISO 8859-1 [14].

NOTE 1: The optional data items may be omitted from the EF though they are followed by mandatory data. For its presence in the EF is marked by a tag.

7) Path tag:

- purpose: to mark the information on the path within the applicational data.
- coding: "51".

8) Path length:

- contents: number of file IDs in the path;
- coding: hexadecimal.

9) Path:

- contents: IDs of the files in the path to the application;
- purpose: to allow the PE to select the DECT application;
- coding: hexadecimal, starting with the ID of the MF and ending with the ID of the DECT application.

NOTE 2: In a multi-application card the order of appearance of the applications within EF_{DIR} is not specified.

10.1.6 EF_{LANG}

EF_{LANG} may be used to select a language for displaying messages.

Identifier: "2F05"	Structure: transparent	Optional
Access conditions: READ ALWAYS UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

Byte(s)	Description	M/O	Length
1 – 2	First language preference	O	2 bytes
3 – 4	Second language preference	O	2 bytes
5 – 6	Third language preference	O	2 bytes
7 – 8	Fourth language preference	O	2 bytes

1) Language preference:

- contents: up to four preferences, in order of priority, according to ISO 639 [6];
- coding: ASCII.

NOTE: The equivalent file in the Subscriber Identity Module (SIM) is part of the GSM directory and not at MF level. The languages are coded on one byte using a GSM specific coding.

10.2 Contents of EFs at the parent level of the DECT application

Only two EFs are specified at this level. Both contain CHV related information.

10.2.1 EF_{CHV}

In a mono application DAM, EF_{CHV1} shall be at the MF level. Only EF_{CHV1} is used in a DECT session. The optional EF_{CHV2} may be used for administrative purposes. EF_{CHV2} may be at a different level.

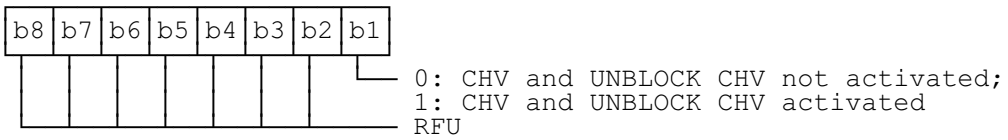
Identifier: "0000"(CHV1)	Structure: transparent	Mandatory
Identifier: "0100"(CHV2)	Structure: transparent	Optional

Access conditions:

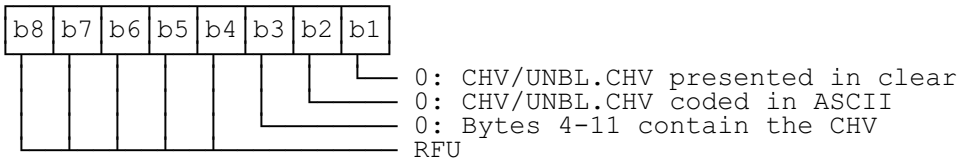
READ	NEVER
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Byte(s)	Description	M/O	Length
1	EF_{CHV} activation byte	M	1 byte
2	Way to present the CHV/UNB.CHV	M	1 byte
3	Coded "FF"	M	1 byte
4 – 11	CHV	M	8 bytes
12	"03" [Number of permissible consecutive false CHV attempts]	M	1 byte
13	Remaining CHV attempts counter	M	1 byte
14 – 21	UNBLOCK CHV	M	8 bytes
22	Remaining UNBLOCK CHV attempts counter	M	1 byte
23	Coded "FF"	M	1 byte

- EF_{CHV} activation byte:



- The way to present the CHV or the UNBLOCK CHV:



  The options b1=1 (CHV/UNBL.CHV presented enciphered), b2=1 (CHV/UNBL.CHV coded in BCD) and b3=1 (bytes 4-11 contain the path to the EF) are not used in the DECT application.

- Byte 3 is not used in the DECT application and shall be coded "FF".

- Each CHV consists of 4 to 8 (decimal) digits (0-9). They are coded in ASCII and right padded with "FF".

- Each UNBLOCK CHV consists of 8 decimal digits (0-9). They are coded in ASCII.

- Byte 23 shall be coded "FF". This implies that the UNBLOCKING CHV mechanism (see prEN 726-3, [15]) may be used an infinite number of times (subject to the correct value being entered).

10.3 Contents of the EFs at the DECT application level

The Dedicated File DF_{DECT} contains all information specifically related to the subscription registrations. At present, there are three EFs specified at DECT application level. In addition, a specific DF is contained in DF_{DECT} for each allocated and future subscription registration. All elementary files related to a given subscription registration are kept in the same DF and specified in subclause 10.4.

10.3.1 EF_{LSR}

EF_{LSR} contains a list of all subscription registrations and the file identifiers of the respective DFs. As the provision of memory space for two subscription registrations is mandatory, EF_{LSR} contains at least two records. Each record consists of the file ID followed by the name of the subscription registration.

Identifier: "6F80"	Structure: linear fixed		Mandatory
Access conditions: READ ALWAYS UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Byte(s)	Description	M/O	Length
1 - 2	File ID of "first" DF_{SR}	M	2 bytes
3 - 14	Name of "first" subscr. registr.	M	12 bytes
15 - 16	File ID of "second" DF_{SR}	M	2 bytes
17 - 28	Name of "second" subscr. registr.	M	12 bytes
29 - n	Further subscription registrations	O	

1) File ID:

   - coding: hexadecimal;

2) Name:

   - coding: ASCII, left justified and right padded with "F".

10.3.2 EF_{LCSR}

EF_{LCSR} contains the file identifier of the last chosen subscription registration which is used as the default registration.

Identifier: "6F81"	Structure: transparent		Mandatory
Access conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Byte(s)	Description	M/O	Length
1 - 2	File ID of last chosen subscr. reg.	M	2 bytes

1) File ID:

   - coding: hexadecimal.

10.3.3 EF_{IPDI}

EF_{IPDI} contains the International Portable DAM Identity (IPDI) which is used for entering the data for a new subscription registration. This identity is used in a similar way as the IPUI N (see ETS 300 175-6 [3]).

Identifier: "6F82"	Structure: transparent		Mandatory
Access conditions: READ CHV1 UPDATE NEVER INVALIDATE ADM REHABILITATE ADM			
Byte(s)	Description	M/O	Length
1 - 5	IPDI	M	5 bytes

1) IPDI:

- coding: identical to IPUI N, hexadecimal.

10.4 Contents of the EFs at the subscription registration level

This subclause specifies all EFs files related to a given subscription registration. They are all kept in the same Dedicated File DF_{SR}. One set of keys is specified for each subscription registration. They relate to the (unique) IPUI of this subscription registration and are valid for all PARKs.

10.4.1 EF_{SR}

EF_{SR} contains data related to the specific subscription registration.

Identifier: "6F02"	Structure: transparent		Mandatory
Access conditions: READ ALWAYS UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Byte(s)	Description	M/O	Length
1	DECT Phase	M	1 byte
2	Bitmap of available ACs	M	1 byte
3	Bitmap of available UAKs	M	1 byte
4	Number (0 or 1) of available DCKs	M	1 byte
5 - X	Identifiers of available authentication algorithms (see note)	M	1 or 2 bytes per algorithm

NOTE: 2 bytes are used for proprietary algorithms while 1 byte is used for the DECT DSAA and the GSM algorithm (see ETS 300 175-5, [2], subclause 7.7.4).

1) DECT Phase:

- contents: the phase of the DECT application coded in this subscription registration;
- purpose: to convey information about the phase supported by the subscription registration. The definition of the phase is up to the issuer;
- coding: hexadecimal.

2) Bitmap of available keys:

Example for coding of available keys:

b8	b7	b6	b5	b4	b3	b2	b1	Bit number
0	1	1	0	0	0	1	0	
0	1	2	3	4	5	6	7	Key number

The above coding means that keys number 1, 2 and 6 are available.

10.4.2 EF_{IPUI}

EF_{IPUI} contains the International Portable User Identity, which uniquely defines one user within the domain defined by his Access Rights related to this IPUI. The IPUI consists of a PUT and a PUN. See ETS 300 175-6 [3] for details.

Identifier: "6F11"	Structure: transparent		Mandatory
Access conditions: READ CHV1 UPDATE FT-AUT INVALIDATE ADM REHABILITATE ADM			
Byte(s)	Description	M/O	Length
1	PUT and leftmost 4 bits of PUN	M	1 byte
2 - 13	Rightmost bits of PUN	M	12 bytes

1 PUT: Portable User Type:

- contents: 4 bit value;
- purpose: the PUT shows the numbering plan structure of a PUN;
- coding: binary.

2) PUN: Portable User Number:

- contents: up to 100 bit value;
- purpose: the PUN identifies a PP and is a globally or locally unique number within one PUT;
- coding: binary.

10.4.3 EF_{PARK}

EF_{PARK} contains the Portable Access Rights Key (PARK), which defines the access rights for a PT. A PARK consists of an ARC and an ARD. It is associated to an IPUI; to one IPUI several PARKs may be associated. The number of PARKs in addition to one is up to the issuer.

NOTE: For details see ETS 300 175-6 [3], subclause 6.1.

Identifier: "6F12"	Structure: transparent		Mandatory
Access conditions: READ CHV1 UPDATE FT-AUT INVALIDATE ADM REHABILITATE ADM			
Byte(s)	Description	M/O	Length
1	PLI of the 1st PARK	M	1 byte
2	"0", ARC, leftmost 4 bits of ARD	M	1 byte
3 - 6	Rightmost bits of ARD	M	4 bytes
7	PLI of the 2nd PARK	O	1 byte
8	"0", ARC, leftmost 4 bits of ARD	O	1 byte
9 - 12	Rightmost bits of ARD	O	4 bytes
⋮	⋮	⋮	⋮
6(n-1)+1	PLI of the nth PARK	O	1 byte
6(n-1)+2	"0", ARC, leftmost 4 bits of ARD	O	1 byte
6n-3 - 6n	Rightmost bits of ARD	O	4 bytes

1) PLI:

- contents: Park length indicator;
- purpose: to indicate the "don't care" bits;
- coding: binary.

2) Bytes 2 - 6:

- contents: 0-bit followed by ARC and ARD;
- purpose: the ARC shows the type of access to a DECT network; the ARD is unique to the service provider, its structure depends on the ARC;
- coding: binary, left justified and padded with 1-bits.

10.4.4 EF_{TPUI}

EF_{TPUI} contains the individual Temporary Portable User Identity associated with the IPUI of this subscription registration. See ETS 300 175-6 [3], subclause 6.3, for details.

Identifier: "6F13"	Structure: transparent	Optional

Access conditions:

READ	CHV1
UPDATE	FT-AUT
INVALIDATE	ADM
REHABILITATE	ADM

Byte(s)	Description	M/O	Length
1 - 3	TPUI	M	3 bytes

- TPUI: Temporary Portable User Identity.
  Contents: up to 20 bit value.
  Coding: hexadecimal.

10.4.5 EF_{ZAP}

EF_{ZAP} contains the ZAP field. It allows an FT to disable the PT.

Identifier: "6F14"	Structure: transparent	Optional

Access conditions:

READ	ALWAYS
UPDATE	FT-AUT
INVALIDATE	ADM
REHABILITATE	ADM

Byte(s)	Description	M/O	Length
1	ZAP field	M	1 byte

1) ZAP field:

- contents: 4 bit value;
- purpose: the ZAP field may be incremented (by using the UPDATE command) during PT AUTHENTICATION;
- coding: binary, left justified and padded with four 1-bits.

10.4.6 EF_{DCK}

EF_{DCK} stores one Derived Cipher Key (DCK).

Identifier: "6F21"	Structure: linear fixed	Mandatory

Access conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 – 8	DCK number 0	M	8 bytes

1) Derived Cipher Key:

- contents: storage is provided for (only) one DCK per subscription registration.
- coding: binary.

NOTE: ETS 300 175-7 [4] allows up to 8 DCKs.

10.4.7 EF_{UAK}

EF_{UAK} stores up to 8 User Authentication Keys (UAK) of up to 128 bits each.

Identifier: "6F22"	Structure: linear fixed		Mandatory
Access conditions: READ NEVER UPDATE NEVER INVALIDATE ADM REHABILITATE ADM			
Byte(s)	Description	M/O	Length
1	Length of UAK number 0	M	1 byte
2 – 17	UAK number 0,	M	16 bytes
18	Length of UAK number 1	O	1 byte
19 – 34	UAK number 1,	O	16 bytes
⋮	⋮	⋮	⋮
120	Length of UAK number 7	O	1 byte
121-136	UAK number 7	O	16 bytes

EF_{UAK} consists of up to 8 records. The first byte of each record contains the number of bits of the UAK which is coded on bytes 2 to 17 of this record.

- purpose: to provide at least one User Authentication Key;
- coding: the length of the UAK is coded hexadecimal; the UAK is coded binary, left justified and padded with 1-bits.

NOTE: The length of each key needs to be stored since padding bits may otherwise be interpreted as part of the key.

10.4.8 EF_{AC}

EF_{AC} stores up to 8 Authentication Codes (AC). Each AC has a bit length of 16 or 32.

Identifier: "6F23"	Structure: linear fixed		Mandatory
Access conditions : READ NEVER UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Byte(s)	Description	M/O	Length
1	Length of AC number 0 (2 or 4)	M	1 byte
2 – 5	AC number 0	M	4 bytes
6	Length of AC number 1 (2 or 4)	O	1 byte
7 – 10	AC number 1	O	4 bytes
⋮	⋮	⋮	⋮
36	Length of AC number 7 (2 or 4)	O	1 byte
37 - 40	AC number 7	O	4 bytes

EF_{AC} consists of up to 8 records. The first byte of each record contains the number of bytes of the AC which is coded on bytes 2 to 5 of this record.

- purpose: to provide at least one Authentication Code;
- coding: the length of the AC is coded hexadecimal; the AC itself is coded binary, left justified and padded with 1-bits;

NOTE: The length of each key needs to be stored since padding bits may otherwise be interpreted as part of the key.

10.4.9 EF_{ST}

This EF indicates which services are allocated, and whether, if allocated, the service is activated. If a service is not allocated or not activated in the DAM, the PE shall not select this service.

Identifier: "6F38"	Structure: transparent		Mandatory
Access conditions : READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Byte(s)	Description	M/O	Length
1	Services n°1 to n°4	M	1 byte

1) Services:

Contents:

- service n°1: CHV1 disable function;
- service n°2: CHV2 disable function;
- service n°3: RFU;
- service n°4: RFU.

NOTE: Other services are possible in the future and will be coded on further bytes in the EF. The coding falls under the responsibility of ETSI.

2) Coding:

2 bits are used to code each service:

- first bit = 1: service allocated;
- first bit = 0: service not allocated;

where the first bit is b1, b3, b5 or b7;

- second bit = 1: service activated;
- second bit = 0: service not activated;

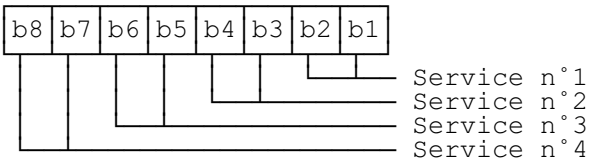
where the second bit is b2, b4, b6 or b8.

Service allocated means that the SIM has the capability to support the service. Service activated means that the service is available for the card holder (only valid if the service is allocated).

The following coding are possible:

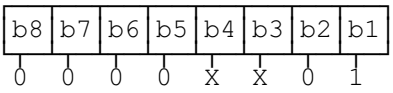
- first bit = 0: service not allocated, second bit has no meaning;
- first bit = 1 and second bit = 0: service allocated but not activated;
- first bit = 1 and second bit = 1: service allocated and activated.

First byte:



etc.

The following example of coding for the first byte means that service n°1 "CHV1-Disabling" is allocated but not activated. Bits b5 to b8 are set to 0, as these services are RFU.



11 Application protocol

When involved in DECT administrative management operations, the DAM interfaces with appropriate terminal equipment. These operations are outside the scope of this ETS.

When involved in DECT network operations the DAM interfaces with a PE with which messages are exchanged. A message can be a command or a response.

- A DECT command/response pair is a sequence consisting of one command and the associated response.

- A DECT procedure consists of one or more DECT command/response pairs which are used to perform all or part of an application-oriented task. A procedure shall be considered as a whole, that is to say that the corresponding task is achieved if and only if the procedure is completed. The PE shall ensure that, when operated according to the manufacturer's manual, any unspecified interruption of the sequence of command/response pairs which realise the procedure, leads to the abortion of the procedure itself.

- A DECT session of the DAM in the DECT application is the interval of time starting at the completion of the DAM initialisation procedure and ending either with the start of the DECT session termination procedure, or at the first instant the link between the DAM and the PE is interrupted.

During the DECT network operation phase, the PE plays the rôle of the master and the DAM plays the rôle of the slave.

Some procedures at the DAM/PE interface require Man Machine Interface (MMI) interactions. The descriptions hereafter do not intend to infer any specific implementation of the corresponding MMI. When MMI interaction is required, it is marked "MMI" in the list given below.

Some procedures are not clearly user dependent. They are directly caused by the interaction of the PT and the FT. Such procedures are marked "FT/PE" in the list given below.

Some procedures are automatically invoked by the PE. They are marked "PE" in the list given below.

The list of procedures at the DAM/PE interface in DECT network operation is as follows:

General procedures (subclause 11.1):

- reading an EF;
- updating an EF;
- increasing an EF.

DAM management procedures (subclause 11.2):

- DAM initialisation PE;
- selection of subscription registration PE;
- DECT session termination PE;
- DAM service table request PE;
- language preference list PE;
- update of a subscription registration (subclause 11.7).

CHV related procedures (subclause 11.3):

- CHV verification MMI;
- CHV value substitution MMI;
- CHV disabling MMI;
- CHV enabling MMI;
- CHV unblocking MMI.

Authentication procedures (subclause 11.4):

- authentication of a PT FT/PE;
- authentication of an FT FT/PE;
- user Authentication MMI;
- mutual Authentication FT/PE.

UAK allocation (subclause 11.5): FT/PE.

General information procedures (subclause 11.6):

- EF_{ICC} request MMI;
- EF_{ID} request MMI;
- EF_{NAME} request MMI;
- EF_{IC} request MMI.

The procedures listed in subclause 11.2 are basically required for execution of the procedures in subclauses 11.3, 11.4 and 11.5. The procedures listed in subclauses 11.3 and 11.4 are mandatory to be implemented. The procedures listed in subclause 11.6 are optional; they may or may not be supported by a PE. However, if the procedures are implemented, it shall be in accordance with subclause 11.6.

If a procedure is related to a specific service indicated in the service table in EF_{ST}, it shall only be executed if the corresponding bits denote this service as "allocated and activated" (see subclause 10.4.10). In all other cases this procedure shall not start.

11.1 General procedures

11.1.1 Reading an EF

The PE selects the EF and sends a READ command. This contains the location of the data to be read. If the access condition for READ is fulfilled, the DAM sends the requested data contained in the EF to the PE. If the access condition is not fulfilled, no data will be sent and an error code will be returned.

11.1.2 Updating an EF

The PE selects the EF and sends an UPDATE command. This contains the location of the data to be updated and the new data to be stored. If the access condition for UPDATE is fulfilled, the DAM updates the selected EF replacing the existing data in the EF with that contained in the command. If the access condition is not fulfilled, the data existing in the EF will be unchanged, the new data will not be stored, and an error code will be returned.

11.1.3 Increasing an EF

The PE selects the EF and sends an INCREASE command. This contains the value which has to be added to the contents of the last updated/increased record. If the access condition for INCREASE is fulfilled, the DAM increases the existing value of the EF by the data contained in the command, and stores the result. If the access condition is not fulfilled, the data existing in the EF will be unchanged and an error code will be returned.

NOTE: The identification of the data within an EF to be acted upon by the above procedures is specified within the command. For the procedures in clauses 11.1.1 and 11.1.2 this data may have been previously identified using a SEEK command, e.g. searching for an alphanumeric pattern.

11.2 DAM management procedures

11.2.1 DAM initialisation

After DAM activation (see clause 4), the PE selects EF_{LANG} and requests the language preference. If this EF is not available or the languages listed in the EF are not supported then the PE chooses a default language.

The PE then selects EF_{ICC}.

If the type of selection is coded in EF_{ICC}, the PE chooses one of the selection methods supported by the DAM and selects DF_{DECT} by direct selection or by selecting EF_{DIR} and the information provided there.

If the type of selection is not coded in EF_{ICC}, the PE selects EF_{DIR}. If EF_{DIR} is contained in the DAM and if EF_{DIR} contains the necessary information for selecting the DECT application, the PE selects DF_{DECT} by using the identifier or by the path given. If EF_{DIR} does not contain the necessary information or is not contained in the DAM, the PE performs the direct selection of DF_{DECT}.

The PE then runs the CHV1 verification procedure. If the CHV1 verification procedure is performed successfully, a DECT session may start. If the CHV1 verification procedure is not performed successfully, the DAM initialisation procedure is aborted.

The PE selects EF_{LCSR} and requests the Last Chosen Subscription Registration (LCSR). The PE selects EF_{LSR} and requests the Subscription Registration List. The PE selects the subscription registration DF_{SR} as chosen by the user. The subscription registration contained in EF_{LCSR} is used as the default subscription registration.

Afterwards the PE runs the following procedures:

- Subscription Registration (EF_{SR}) request;
- Service Table (EF_{ST}) request;
- IPUI (EF_{IPUI}) request;
- PARK (EF_{PARK}) request;
- TPUI (EF_{TPUI}) request;
- ZAP (EF_{ZAP}) request;
- DCK (EF_{DCK}) request.

After the DAM initialisation procedure has been completed successfully, the PT is ready for a DECT session.

11.2.2 DAM session termination

NOTE 1: This procedure is not to be confused with the deactivation procedure of the contacts in clause 4.

The DECT session is terminated by the PE as follows. The PE runs all the procedures which are necessary to transfer the following subscriber related information to the DAM:

- DCK update;
- last chosen registration update.

NOTE 2: Further procedures may be required with respect to the DECT profile supported by the DAM. They will be specified for the DAM as part of the requirements of that profile.

As soon as the DAM indicates that these procedures are completed, another subscription registration may be selected or the PE/DAM link may be deactivated.

Finally, the PE deletes all these subscriber related information elements from its memory.

NOTE 3: If the PE has already updated any of the subscriber related information, and the value has not changed until DECT session termination, the PE may omit the respective update procedure.

11.2.3 Language preference

Request: The PE performs the reading procedure with EF_{LANG}.
Update: The PE performs the updating procedure with EF_{LANG}.

11.2.4 Service table request

The PE performs the reading procedure with EF_{ST}.

11.2.5 DAM presence detection

The PE sends at frequent intervals a STATUS command during each call. This interval shall not be longer than 30 seconds. If the response data is not that of the current DF, the call shall be terminated immediately. See also subclause 5.11.

11.3 CHV related procedures

A successful completion of one of the following procedures grants the access right of the corresponding CHV for the DECT session. This right is valid for all files within the DECT application protected by this CHV.

After a third consecutive presentation of a wrong CHV to the DAM, not necessarily in the same DAM session, the respective CHV status becomes "blocked" and the access right previously granted by this CHV is lost immediately.

An access right is not granted if any of the following procedures are unsuccessfully completed or aborted.

NOTE: It is not necessary to select the relevant EF_{CHV} to perform these procedures.

11.3.1 CHV verification

The PE checks the CHV status. If the CHV status is "blocked" the procedure ends and is finished unsuccessfully.

If the CHV status is not "blocked", the PE reads the CHV enabled/disabled indicator. If this is "disabled", the procedure is finished successfully.

If the CHV status is not "blocked" and the enabled/disabled indicator set "enabled", the PE uses the VERIFY CHV function. If the CHV presented by the PE is equal to the corresponding CHV stored in the respective EF_{CHV}, the procedure is finished successfully. If the CHV presented by the PE is not equal to the corresponding CHV stored in the respective EF_{CHV}, the procedure ends and is finished unsuccessfully.

11.3.2 CHV value substitution

The PE checks the CHV status. If the CHV status is "blocked" or "disabled", the procedure ends and is finished unsuccessfully.

If the CHV status is not "blocked" and the enabled/disabled indicator is set to "enabled", the PE uses the CHANGE CHV function. If the old CHV presented by the PE is equal to the CHV stored in the respective EF_{CHV}, the new CHV presented by the PE is stored in the DAM and the procedure is finished successfully.

If the old CHV and the CHV in memory are not identical, the procedure ends and is finished unsuccessfully.

11.3.3 CHV disabling

Requirement: Service CHV disabling is both allocated and activated for the respective CHV.

The PE checks the status of the respective CHV. If this is "blocked", the procedure ends and is finished unsuccessfully.

If the respective CHV status is not "blocked", the PE reads the respective CHV enabled/disabled indicator. If this is "disabled", the procedure ends and is finished unsuccessfully.

If the respective CHV status is not "blocked" and the enabled/disabled indicator "enabled", the PE uses the DISABLE CHV function. If the CHV presented by the PE is equal to the CHV stored in the respective EF_{CHV}, the status of the respective CHV is set "disabled" and the procedure is finished successfully. If the CHV presented by the PE is not equal to the CHV stored in the respective EF_{CHV}, the procedure ends and is finished unsuccessfully.

11.3.4 CHV enabling

Requirement: The service "CHV disabling" is both allocated and activated for the respective CHV.

The PE checks the respective CHV status. If the CHV status is "blocked", the procedure ends and is finished unsuccessfully.

If the CHV status is not "blocked", the PE reads the respective CHV enabled/disabled indicator. If this is "enabled", the procedure ends and is finished unsuccessfully.

If the CHV status is not "blocked" and the enabled/disabled indicator is set "disabled", the PE uses the ENABLE CHV function. If the CHV presented by the PE is equal to the CHV stored in the respective EF_{CHV}, the status of the respective CHV is set "enabled" and the procedure is finished successfully. If the CHV presented by the PE is not equal to the CHV stored in the respective EF_{CHV}, the procedure ends and is finished unsuccessfully.

11.3.5 CHV unblocking

The execution of the CHV unblocking procedure is independent of the corresponding CHV status, i.e being blocked or not.

The PE checks the UNBLOCK CHV status. If the UNBLOCK CHV status is "blocked", the procedure ends and is finished unsuccessfully.

If the UNBLOCK CHV is not "blocked", the PE uses the UNBLOCK CHV function. If the UNBLOCK CHV presented by the PE is equal to the UNBLOCK CHV stored in the relevant EF_{CHV}, the relevant CHV status is "unblocked" and the procedure is finished successfully. If the UNBLOCK CHV presented by the PE is not equal to the UNBLOCK CHV stored in the relevant EF_{CHV}, the procedure ends and is finished unsuccessfully.

11.4 Authentication procedures

There are four types of authentication procedures based on the mechanisms specified in clause 7:

- authentication of a PT;
- authentication of an FT;
- user Authentication;
- mutual Authentication.

Generally, the procedures use the DECT Standard Authentication Algorithm (DSAA), although other algorithms may be employed instead. The specification of the data elements used in the authentication procedures can be found in ETS 300 175-5 [2].

11.4.1 Authentication of a PT

This procedure is used by the FT to authenticate the DAM when placed in the PE. The procedure is depicted in figure 8 and consists of the following steps. Steps (1), (2), (9) and (10) are not part of the DAM/PE interface. They are included for information only.

1) The FT obtains the random numbers RAND_F and RS and the value XRES1.

2) The FT sends an AUTHentication-REQUEST message to the PE. It contains:

   - AUTH-TYPE;
   - RAND_F;
   - RS;
   - cipher info (optional).

3) The PE examines the AUTH-TYPE element. If INC is set to 1, the PE shall initiate the authentication of an FT procedure (see subclause 11.4.2). If the authentication of an FT is unsuccessful, then the whole process is aborted.

4) The PE selects EF_{UAK} or EF_{AC} in the DAM.

5) The PE sends a PT AUTHENTICATION command to the DAM including:

   - a key number;
   - an algorithm identifier;
   - INC;
   - RS;
   - RAND_F.

6) The DAM calculates RES1 and DCK as well as the ZAP, if this is stored in the DAM.

7) The PE sends a GET RESPONSE command to the DAM (T=0).

8) The DAM sends RES1, DCK and, if calculated, ZAP to the PE.

9) The PE sends an AUTHentication-REPLY message to the FT. It contains:

- RES1;
- DCK (if appropriate);
- ZAP (if appropriate).

10) The FT compares RES1 with XRES1. Only if the two values are equal the authentication of the PT is successful. Furthermore, the FT may compare the ZAP with the expected value. In case of an unsuccessful authentication the FT shall drop the call.

11) If storage of the DCK is indicated in the AUTHentication-REQUEST message, the PE shall select and update EF_{DCK} and EF_{SR}.

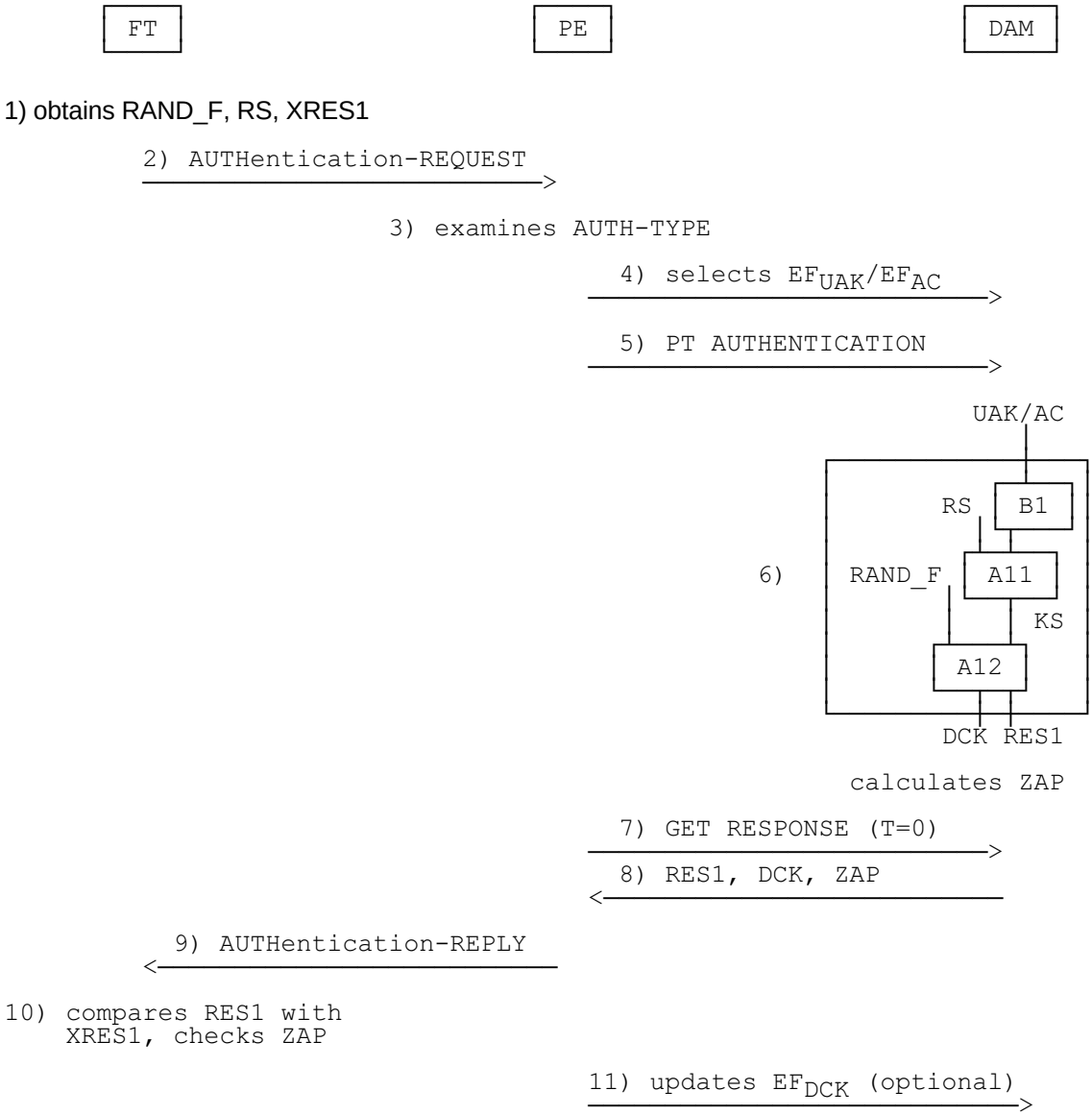


Figure 8: Authentication of a PT

11.4.2 Authentication of an FT

This procedure is used by the DAM when placed in the PE to authenticate an FT. The procedure is depicted in figure 9 and consists of the following steps. Steps (6), (7) and (8) are not part of the DAM/PE interface. They are included for information only.

1) The PE selects and reads EF_{SR} in the DAM.

2) The PE selects EF_{AC} or EF_{UAK} in the DAM.

3) The PE sends an ASK RANDOM command to the DAM.

4) The DAM generates RAND_P.

5) The DAM sends RAND_P to the PE.

6) The PE sends an AUTHentication-REQUEST message to the FT. It contains:

   - AUTH-TYPE;
   - RAND_P;
   - cipher info (optional).

7) The FT obtains RS and RES2.

8) The FT sends an AUTHentication-REPLY message to the PE. It contains:

   - RES2;
   - RS.

9) The PE sends an FT AUTHENTICATION command to the DAM containing:

   - a key number;
   - an algorithm identifier;
   - RS;
   - RES2.

10) The DAM calculates XRES2.

11) The DAM compares RES2 with XRES2. Only if the two values are equal the authentication of the FT is successful. In case of an unsuccessful authentication the DAM shall inform the PE which shall then drop the call.

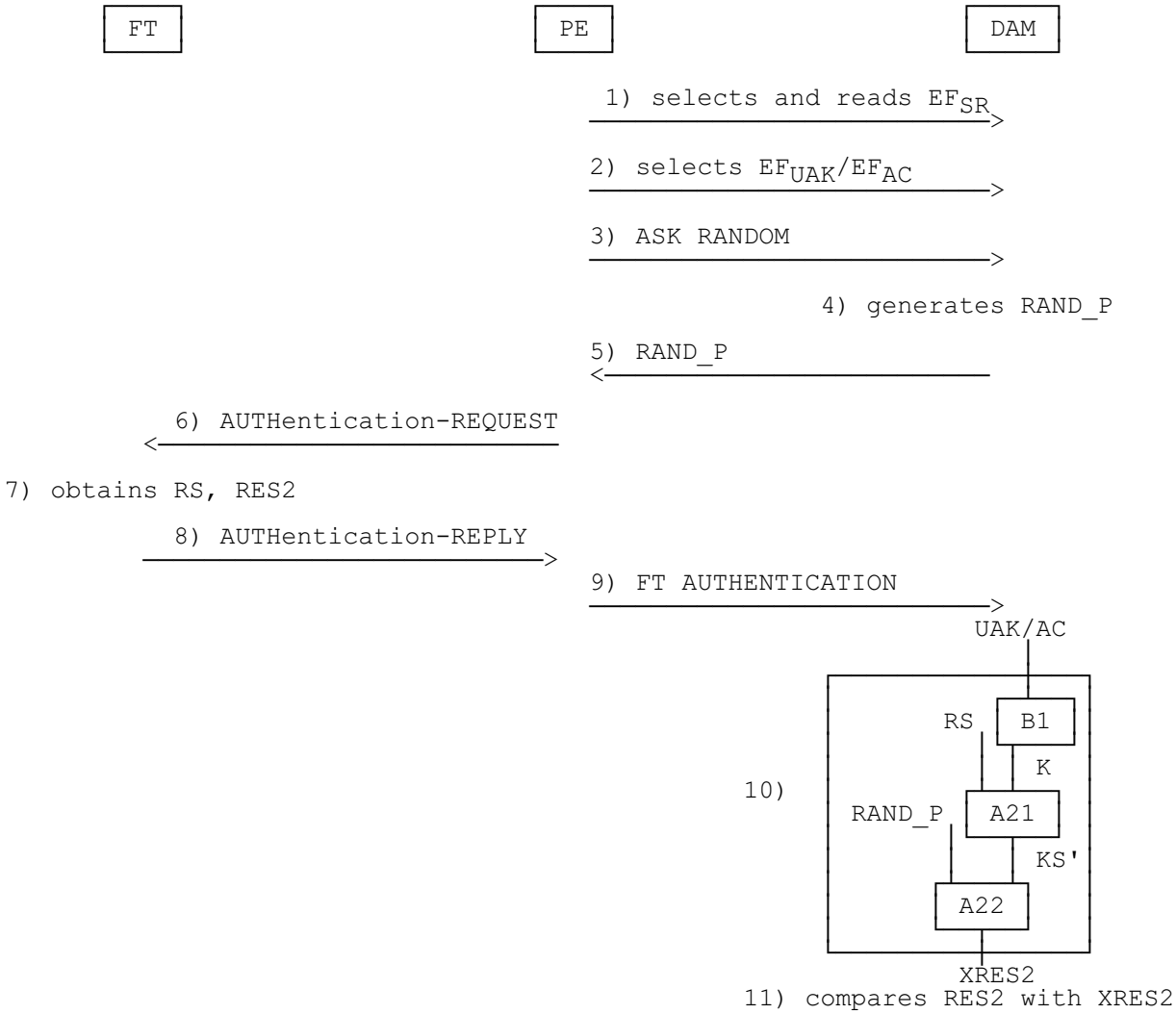


Figure 9: Authentication of an FT

11.4.3 User authentication

This procedure is used by the FT to authenticate a user of the DAM when placed in the PE. The procedure is depicted in figure 10 and consists of the following steps. Steps (1), (2), (4), (10) and (11) are not part of the DAM/PE interface. They are included for information only.

1) The FT obtains the random numbers RAND_F and RS and the value XRES1.

2) The FT sends an AUTHentication-REQUEST message to the PE. It contains:

   - AUTH-TYPE;
   - RAND_F;
   - RS;
   - cipher info (optional).

3) The PE examines the AUTH-TYPE element. If INC is set to 1, the PE shall initiate the authentication of an FT procedure (see clause 7). If the authentication of an FT is unsuccessful, then the whole procedure is aborted.

4) The PE obtains the UPI from the user.

5) The PE selects EF_{UAK} in the DAM.

6) The PE sends a USER AUTHENTICATION command to the DAM including:

- a key number;
- an algorithm identifier;
- RS;
- RAND_F;
- UPI.

7) The DAM calculates RES1 and DCK.

8) The PE sends a GET RESPONSE command to the DAM (T=0).

9) The DAM sends RES1 and DCK to the PE.

10) The PE sends an AUTHentication-REPLY message to the FT. It contains:

- RES1;
- DCK (if appropriate).

11) The FT compares RES1 with XRES1. Only if the two values are equal the authentication of the user is successful. In case of an unsuccessful authentication the FT shall drop the call.

12) If storage of the DCK is indicated in the AUTHentication-REQUEST message, the PE shall select and update EF_{DCK} and EF_{SR}.

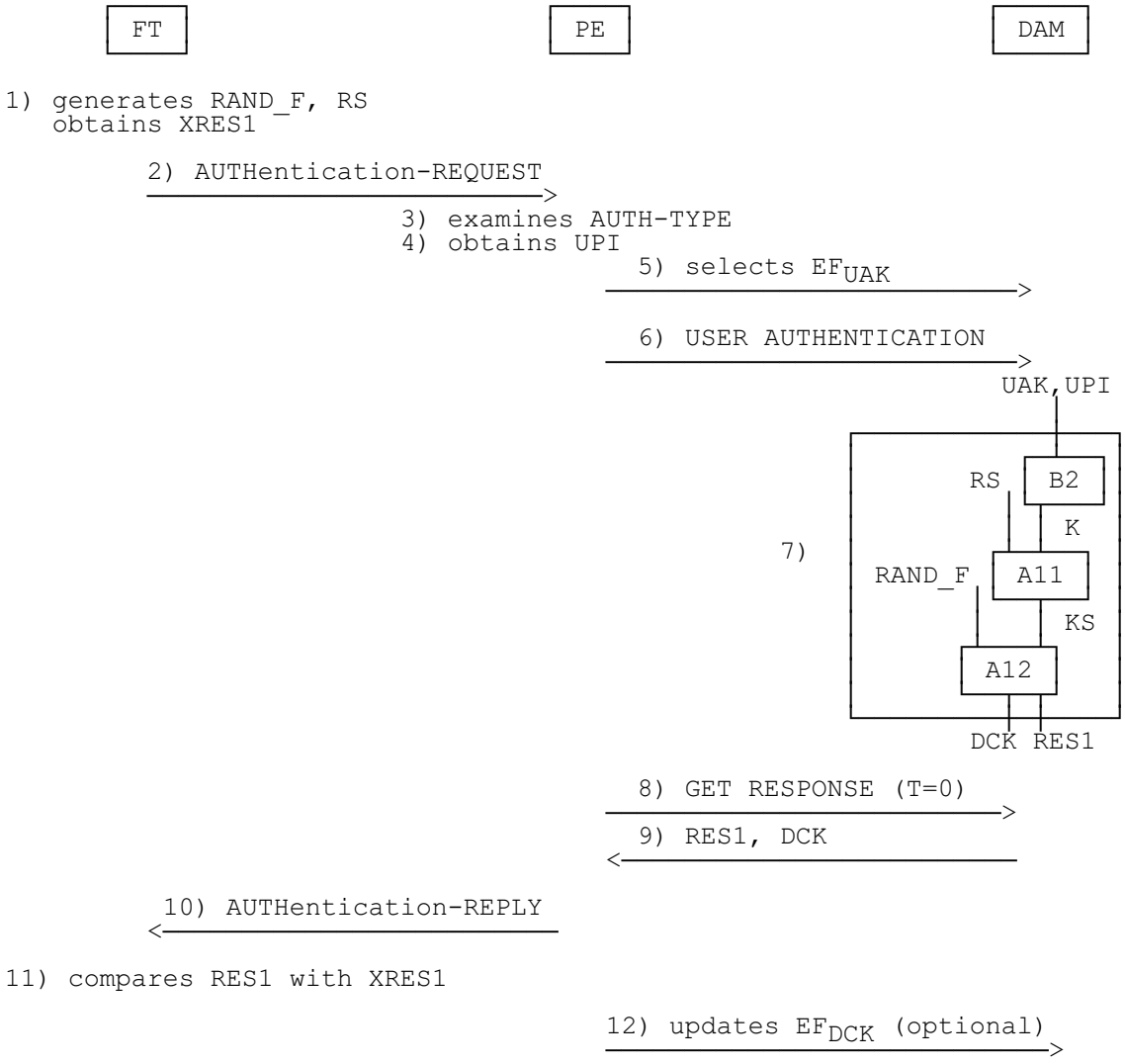


Figure 10: User authentication

11.4.4 Mutual authentication

There are two methods to achieve mutual authentication.

Direct method: an authentication of a PT (subclause 11.4.1) or a user authentication (subclause 11.4.3) is followed by an authentication of an FT (subclause 11.4.2).

Indirect method: this is a combination of an authentication of a PT (subclause 11.4.1) or a user authentication (subclause 11.4.3) and a data confidentiality service. For the data confidentiality service the PE uses the DCK derived during authentication of a PT or user authentication.

11.5 UAK allocation

The procedure for the initial allocation of a UAK derived from an AC is based on the UAK allocation specified in clause 7.

This procedure is used to derive an initial UAK from an AC. The conditions under which this optional procedure is used can be found in ETS 300 175-7 [4]. The UAK is to be associated with an IPUI or an IPUI/PARK pair and the procedure requires that an AC is already associated with that IPUI or IPUI/PARK pair.

The procedure consists of the following steps. Steps 1), 3), 12), 13), 14) and 15) are not part of the DAM/PE interface. They are included for information only.

1) The FT obtains the random numbers RAND_F and RS and the value XRES1.

2) The FT sends a KEY-ALLOCATE message to the PE. This message contains:

   - Allocation type;
   - RAND_F;
   - RS.

3) The PE selects and reads EF_{SR} in the DAM.

4) The PE examines the allocation-type element contained in the KEY-ALLOCATE message. The allocation-type element is only acceptable if the DAM supports the authentication algorithm and contains the AC, which are identified within the element.

   If the Allocation-type element is unacceptable, the PE returns an AUTHentication-REJECT message to the FT.

   If the Allocation-type element is acceptable, the PE performs the following steps.

5) The PE selects EF_{AC} in the DAM.

6) The PE sends a PT AUTHENTICATION command to the DAM including:

   - an algorithm identifier;
   - a key number;
   - RAND_F;
   - RS.

7) The DAM calculates RES1 and DCK.

8) The PE sends a GET RESPONSE command to the DAM (T=0).

9) The DAM sends RES1 and DCK to the PE.

10) The PE sends an ASK RANDOM command to the DAM.

11) The DAM generates RAND_P.

12) The DAM sends RAND_P to the PE.

13) The PE sends an AUTHentication_REQUEST message to the FT which contains:

- AUTH-TYPE;
- RAND_P;
- RES1;
- cipher info (optional).

14) The FT compares RES1 with XRES1. Only if the two values are equal the authentication of the PT is successful and the following steps are executed. Otherwise the FT shall drop the call.

15) The FT obtains RES2. The Authentication Session (KS') key, obtained during the computation of RES2, is the derived UAK. This is assigned to the UAK number identified in the original allocation-type element.

16) The FT sends an AUTHentication-REPLY message to the PE. It contains:

- RES2;
- RS, where RS is identical to the one in the KEY-ALLOCATE message in step (1).

17) The PE sends a UAK ALLOCATION command to the DAM containing:

- a key number;
- a UAK number;
- an algorithm identifier;
- RS;
- RES2, where the UAK number was identified in the original allocation-type element.

18) The DAM calculates XRES2.

19) The DAM compares RES2 with XRES2. Only if the two values are equal the authentication of the FT is successful and the DAM shall store the authentication session key KS', obtained during the process of computing XRES2, under the UAK number identified in the UAK allocation. The PE updates the used AC by setting it to the default value.

If the authentication fails, the DAM shall inform the PE which shall then drop the call.

20) The PE updates the records in EF_{SR} containing the number of available UAKs.

The generation of the UAK by the DAM is depicted in figure 11.

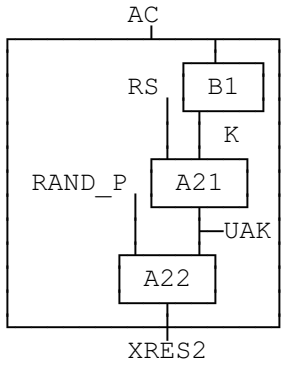


Figure 11: UAK allocation

11.6 General information procedures

These procedures may be used to read and/or update EFs at the MF level.

11.6.1 EF_{ICC} request

The PE performs the reading procedure with EF_{ICC}.

11.6.2 EF_{ID} request

The PE performs the reading procedure with EF_{ICC}.

11.6.3 EF_{NAME} request

Request:

- the PE performs the reading procedure with EF_{NAME} subject to the security policy assigned by the issuer.

Update:

- the PE performs the updating procedure with EF_{NAME} subject to the security policy assigned by the issuer.

11.6.4 EF_{IC} request

The PE performs the reading procedure with EF_{IC}.

11.7 Subscription registration maintenance

The procedures are based on the security processes specified in clause 7.

11.7.1 Entering a new subscription registration

This procedure loads the necessary data for a new subscription into a specific dedicated file DF_{SR} of a DAM, which has been provided in the DAM prior to the commencement of the procedure.

The procedure consists of the following steps:

1) the user sets the PE to the "new subscription registration" mode and selects the subscription registration;

2) the PE selects the DF_{SR} corresponding to the subscription registration requested by the user;

3) the PE selects the corresponding EF_{AC} and updates EF_{AC} by storing the AC received from the user over the MMI;

4) the PE selects and updates EF_{SR};

5) the PE selects and reads EF_{IPDI};

6) the PE sends an ACCESS-RIGHTS-REQUEST message to the FT;

7) the FT sends an ACCESS-RIGHTS-ACCEPT message to the PE;

8) the PE initiates a UAK allocation procedure as specified in subclause 11.5;

9) if the UAK allocation fails, the procedure is aborted;

10) if the UAK allocation is successful, the PE updates EF_{UAK} in DF_{SR} and executes the following step;

11) the PE selects the relevant EFs in DF_{SR} and updates these EFs by storing the new values obtained from the FT.

11.7.2 Updating an existing subscription registration

This procedure adds or deletes a PARK contained in a specific subscription registration. It shall only be performed after a successful mutual authentication between the FT and PT as specified in subclause 11.4.4.

The procedure consists of the following steps:

1) the PE selects DF_{SR};

2) FT and PT perform a mutual authentication for this subscription registration. If the authentication fails the procedure is aborted. If the authentication is successful the following steps are executed for the addition of a PARK (case a) or the deletion of a PARK (case b);

3a) the PE sends an ACCESS-RIGHTS-REQUEST message to the FT;

3b) the PE sends an ACCESS-RIGHTS-TERMINATE message to the FT;

4a) the FT sends an ACCESS-RIGHTS-ACCEPT message to the PE;

4b) the FT sends an ACCESS-RIGHTS-TERMINATE-ACCEPT message to the PE;

5a) the PE adds the PARK obtained in the ACCESS-RIGHTS-ACCEPT message to EF_{PARK};

5b) the PE deletes the PARK obtained in the ACCESS-RIGHTS-TERMINATE-ACCEPT message from EF_{PARK} by updating the value to all "F".

11.7.3 Terminating an existing subscription registration

This procedure terminates a subscription registration by "overwriting" the contents of all EFs of the corresponding DF_{SR}. It shall only be performed after a successful mutual authentication between the FT and PT as specified in subclause 11.4.4.

The procedure consists of the following steps:

1) the PE selects the DF_{SR} indicated in the ACCESS-RIGHTS-TERMINATE-REQUEST message;

2) FT and PT perform a mutual authentication for this subscription registration. If the authentication fails the procedure is aborted. If the authentication is successful the following steps are executed;

3) the PE updates all EFs in DF_{SR} by setting all bytes to "FF";

4) the PE sends an ACCESS-RIGHTS-TERMINATE-ACCEPT message to the FT.

Annex A (normative): Plug-in Card

This annex specifies the dimensions of the Plug-in Card as well as the dimensions and location of the contacts of the Plug-in Card. For further details of the Plug-in Card see clause 4.

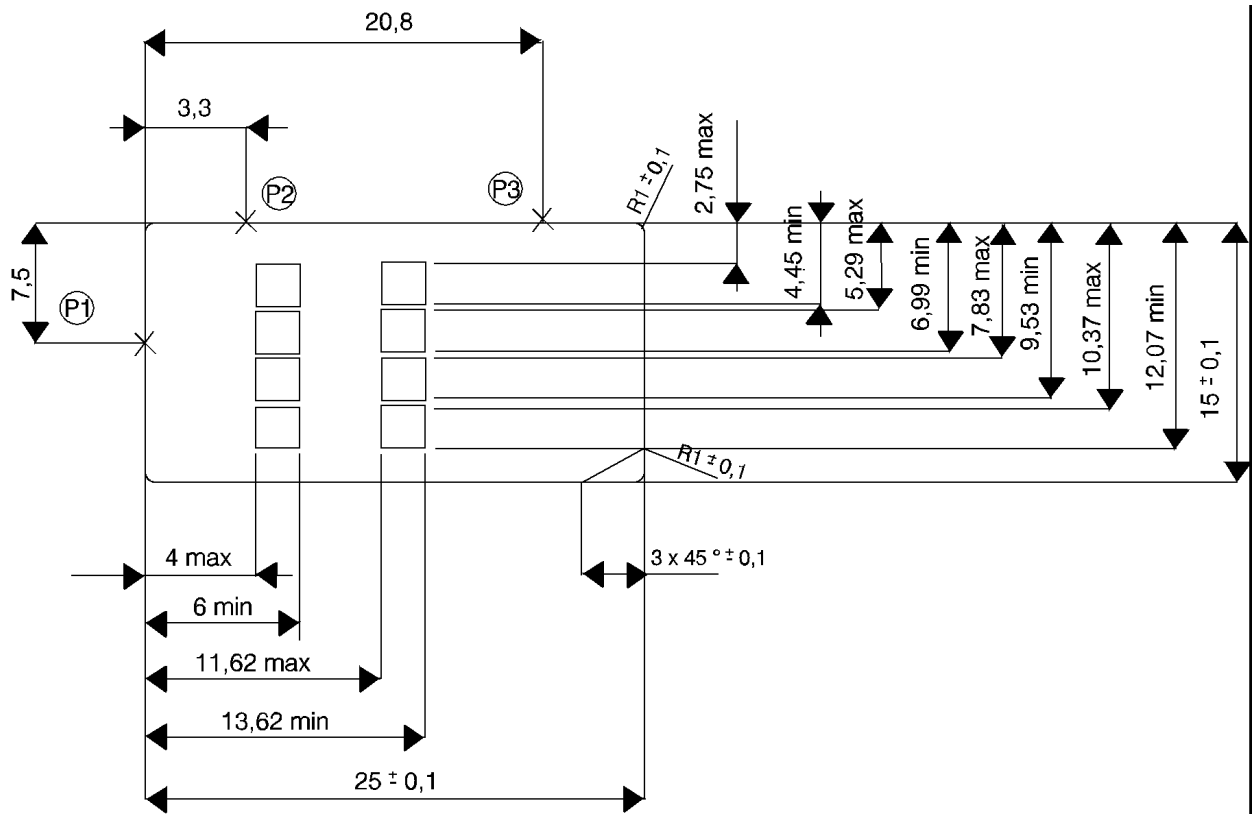


Figure A.1: Plug-in Card

NOTE: The Plug-in Card may be "obtained" by cutting away excessive plastic of an ID-1 Card. The values in parenthesis in figure A.1 show the positional relationship between the Plug-in Card and the ID-1 Card and are for information only.

Annex B (informative): Service class

The service class identifies which service a registration is allowed to use. It is given for information to be taken into consideration when specifying specific application profiles.

The following services are specified:

- one nominated number only may be called;
- as above and local calls are allowed;
- as above and national calls are allowed;
- as above and mobile and premium service;
- as above and international calls;
- as above and satellite services.

Annex C (informative): Bibliography

- ISO/IEC DIS 7816-4 (1994): "Identification cards - Integrated circuit(s) cards with contacts - Part 4: Inter-industry commands for interchange".

- prEN 726-6 (version 8): "Terminal Equipment (TE); Requirements for IC cards and terminals for telecommunication use - Part 6: Telecommunication features".

- ISO/IEC 7816-3 (1989): "Identification cards - Integrated circuit(s) cards with contacts, Part 3: Electronic signals and transmission protocols", AM2 (1993): "Protocol type select".

History

Document history	
November 1995	First Edition
February 1996	Converted into Adobe Acrobat Portable Document Format (PDF)

ISBN 2-7437-0380-6
Dépôt légal : Novembre 1995