EUROPEAN
TELECOMMUNICATION
STANDARD

ETS 300 812

November 1998

Source: TETRA

Reference: DE/TETRA-07017

ICS: 33.020

Key words: Card, security, TETRA

Terrestrial Trunked Radio (TETRA); Security aspects; Subscriber Identity Module to Mobile Equipment (SIM - ME) interface

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
Internet: secretariat@etsi.fr - http://www.etsi.org

Tel.: +33 4 92 94 42 00 - Fax: +33 4 93 65 47 16

Whilst every care has been taken in the preparation and publication of this document, errors in content, typographical or otherwise, may occur. If you have comments concerning its accuracy, please write to "ETSI Standards Making Support Dept." at the address shown on the title page.

Content

Blank page

Foreword

This European Telecommunication Standard (ETS) has been produced by the Terrestrial Trunked Radio (TETRA) Project of the European Telecommunications Standards Institute (ETSI).

Transposition dates	
Date of adoption of this ETS:	27 November 1998
Date of latest announcement of this ETS (doa):	28 February 1999
Date of latest publication of new National Standard or endorsement of this ETS (dop/e):	31 August 1999
Date of withdrawal of any conflicting National Standard (dow):	31 August 1999

Blank page

1 Scope

This ETS defines the interface between the Subscriber Identity Module (SIM) and the Mobile Equipment (ME) for use during the network operation phase of TETRA as well as those aspects of the internal organization of the SIM which are related to the network operation phase. This is to ensure interoperability between a SIM and an ME independently of the respective manufacturers and operators. The concept of a split of the MS into these elements as well as the distinction between the TETRA network operation phase, which is also called TETRA operations, and the administrative management phase is described in the User Requirement Specification ETR 295 [9].

This ETS defines:

- the requirements for the physical characteristics of the SIM, the electrical signals and the transmission protocols;
- the model which shall be used as a basis for the design of the logical structure of the SIM;
- the security features;
- the interface functions;
- the commands;
- the contents of the files required for the TETRA application;
- the application protocol.

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

The physical SIM described in this ETS is a removable Integrated Circuit (IC) card. The SIM is an optional device within TETRA MSs. This ETS does not preclude the implementation of fully functional MSs without a SIM. All references to mobile equipment in this ETS are to be taken to mean mobile equipment which have been designed to operate with a SIM.

This ETS deals with all aspects of trunked mode MS operation. For direct mode MS operation key user operation is supported by the SIM but not key holder or key generator operation. Furthermore, storage of information for direct mode MS operation in repeater and gateway mode are supported, but any extra storage required in the direct mode repeater or direct mode gateway terminals themselves is not supported.

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] ISO 7810 (1985): "Identification cards - Physical characteristics".

[2] ISO 7811-1 (1995): "Identification cards - Recording technique - Part 1: Embossing".

[3] ISO 7811-3 (1995): "Identification cards - Recording technique - Part 3: Location of embossed characters on ID-1 cards".

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

[5] ISO/ISO 7816-2 (1988): "Information technology - Identification cards - Integrated circuit(s) cards with contacts, Part 2: Dimensions and location of the contacts".

[6] ISO/IEC 7816-3 (1997): "Identification cards - Integrated circuit(s) cards with contacts, Part 3: Electronic signals and transmission protocols".

[7] ISO/IEC 7816-4 (1995): "Information technology - Identification cards - Integrated circuit(s) cards with contacts, Part 4: Interindustry commands for interchange".

[8] ENV 1375-1: "Identification card systems - Intersector integrated circuit(s) card additional formats - Part 1: ID-000 card size and physical characteristics".

[9] ETR 295: "User requirements for Subscriber Identity Module (SIM)".

[10] ETS 300 392-1: "Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 1: General Network Design".

[11] ETS 300 392-2: "Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 2: Air Interface (AI)".

[12] ETS 300 392-7: "Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 7: Security".

[13] ETS 300 396-3: "Terrestrial Trunked Radio (TETRA); Technical requirements for Direct Mode Operation (DMO); Part 3: Mobile Station to Mobile Station (MS-MS) Air Interface (AI) protocol".

[14] ETS 300 608 (1996): "Digital cellular telecommunications system (Phase 2); Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface (GSM 11.11)".

[15] ETS 300 641 (1996): "Digital cellular telecommunications system (Phase 2); Specification of the 3 Volt Subscriber Identity Module - Mobile Equipment (SIM-ME) interface (GSM 11.12)".

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

[17] ETS 300 628 (1994): "European digital cellular telecommunications system (Phase 2); Alphabets and language-specific information (GSM 3.38)".

[18] CCITT Recommendation T.50: "International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) – Information technology – 7-bit coded character set for information interchange)".

[19] ETS 300 396-6: "Terrestrial Trunked Radio (TETRA); Direct Mode Operation (DMO); Part 6: Security".

[20] ETS 300 392-12-22: "Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 12: Supplementary services stage 3; Sub-part 22: Dynamic Group Number Assignment (DGNA)".

[21] CCITT Recommendation E.118: "The international telecommunication charge card".

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

[23] ETS 300 394-2 (1997): "Terrestrial Trunked Radio (TETRA); Conformance testing specification; Part 2: Protocol testing specification for Voice plus Data (V+D)".

3 Definitions, abbreviations and symbols

3.1 Definitions

For the purposes of this ETS, the following definitions apply. For further information and definitions refer to ETS 300 392-1 [10].

access conditions: A set of security attributes associated with access to an Elementary File (EF):

ADM (administrative):

indicates an access condition defined by the card issuer. Before issue of the card ADM serves as a placeholder for an access condition to be defined by the card issuer. Any access condition may be assigned. The assigned access condition is used during the usage phase of the SIM.

AUTI (authorized immediate):

defines access conditions to an EF under which access shall is only possible immediately following successful authentication of the Switching and Management Infrastructure (SwMI).

CHVn (card holder verification):

defines the access condition to an EF which requires verification of the user identity (n = 1 or n = 2).

NEV (never):

access to the EF is never allowed across the SIM-ME interface.

RAU (reserved for administrative use):

defines access conditions to an EF which is restricted to the administrative phase of the SIM.

administrative phase: That part of the card life between the manufacturing phase and the usage phase.

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

application protocol: The set of procedures required by the application which are located and used in the Integrated Circuit (IC) card and outside the IC card (external application).

card holder verification: Authentication of the user to the SIM card.

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 Dedicated File (DF) selected.

current Elementary File (EF): The latest EF selected.

current file: The latest MF, DF, or EF selected.

Dedicated File (DF): A file containing access conditions and, optionally, EFs or other 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 organized set of bytes or records in the SIM.

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

key generator: A secure system entity authorized to generate Static Cipher Keys (SCKs) for Direct Mode Operation (DMO).

key holder: A secure system entity authorized to distribute SCKs for DMO.

key user: A standard Direct Mode (DM) terminal which uses SCKs provided by an authorized key holder.

ID-1 SIM: The SIM having the format of an ID-1 card (see ISO 7816-1 [4]).

input: Signifies data input to the SIM functions (defined in clause 8):

Input from SIM: input from the SIM internal memory;
Input from EF: internal input from an EF on the SIM;
Input from ME: data contained in a command APDU passed across the SIM-ME interface.

Master File (MF): The unique mandatory DF representing the root.

mobile equipment: That part of the MS which interfaces to the SIM card.

Mobile Station (MS): The entirety of the equipment needed to communicate with the infrastructure (in trunked mode of operation) or direct with another MS (in direct mode of operation).

output: Signifies data output from the SIM functions (defined in clause 8):

Output to SIM: data shall be stored on the SIM in non-permanent memory for the duration of the TETRA session;
Output to EF: internal updating of an EF on the SIM;
Output to ME: data contained in a response APDU passed across the SIM-ME interface.

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

plug-in SIM: A second format of SIM (specified in clause 4).

record: A string of bytes within an EF handled as a single entity (see clause 6).

record number: The number which identifies a record within an EF.

record pointer: The pointer which addresses one record in an EF.

Subscriber Identity Module (SIM) or SIM card: An integrated circuit card containing network related subscriber information.

TETRA application: A set of security mechanisms, files, data and protocols required by TETRA.

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

TETRA SIM: A subscriber identity module used in a TETRA MS.

usage phase: That part of the card life, after the administrative phase, when the card is being used for operational purposes.

virtual SIM: A conceptual (or hypothetical) SIM which may exist in a physical Mobile Equipment (ME). A virtual SIM has the same functional data structure and potential content as a physical SIM but has no physical or independent existence outside of the ME.

5 V technology SIM: A SIM operating at 5 V ± 10 %.

3 V technology SIM: A SIM operating at 3 V ± 10 % and 5 V ± 10 %.

3 V technology ME: An ME operating the SIM - ME interface at 3 V ± 10 % according to GSM 11.12 (ETS 300 641) [15] and 5 V ± 10 % according to GSM 11.11 (ETS 300 608) [14].

3 V only ME: An ME only operating the SIM - ME interface at 3 V ±10 % according to GSM 11.12 (ETS 300 641) [15].

3.2 Abbreviations

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

ADN	Abbreviated Dialling Number
ADM	ADMinistrative (see definitions)
ALW	ALWays
APDU	Application Protocol Data Unit
ATR	Answer To Reset
AUTI	AUThorized Immediate (see definitions)
BCD	Binary Coded Decimal
CCK	Common Cipher Key
CCK-id	CCK identifier
CHV	Card Holder Verification (see definitions)
CLA	CLAss
CLK	CLocK
DCK	Derived Cipher Key
DCK1	Part 1 of the DCK
DCK2	Part 2 of the DCK
DF	Dedicated File
DGNA	Dynamic Group Number Assignment
DMO	Direct Mode Operation
EF	Elementary File
FDN	Fixed Dialling Number
GCK	Group Cipher Key
GCK-VN	GCK Version Number
GSSI	Group Short Subscriber Identity
GTSI	Group Tetra Subscriber Identity
IC	Integrated Circuit
ID	IDentifier
ISSI	Individual Short Subscriber Identity
ITSI	Individual TETRA Subscriber Identity
I/O	Input/Output
K	individual subscriber authentication key
KE	Enhanced security Key
KS, KS'	Session authentication Key
LAI	Location Area Information
LND	Last Number Dialled
LSB	Least Significant Bit
MCC	Mobile Country Code
ME	Mobile Equipment
MF	Master File
MGCK	Modified Group Cipher Key

MMI	Man Machine Interface
MNC	Mobile Network Code
MS	Mobile Station
MSB	Most Significant Bit
NET	NETwork
NEV	NEVer (see definitions)
OTAR	Over The Air Re-keying
PABX	Private Automatic Branch Exchange
PSTN	Public Switched Telephone Network
RAND1	RANDom challenge 1
RAND2	RANDom challenge 2
RES1	RESponse 1
RES2	RESponse 2
RAU	Reserved for Administrative Use (see definitions)
RFU	Reserved for Future Use
RS	Random Seed
RSO	Random Seed for OTAR
RST	ReSeT
Rx	Receive
SCCK	Sealed CCK
SCK	Static Cipher Key
SCK-VN	SCK version number
SCKN	SCK number
SDN	Service Dialling Number
SDS	Short Data Service
SGCK	Sealed GCK
SIM	Subscriber Identity Module
SSCK	Sealed SCK
SS-DGNA	Supplementary Service Dynamic Group Number Assignment
SSI	Short Subscriber Identity
SW1/SW2	Status Word 1 / Status Word 2
SwMI	Switching and Management Infrastructure
TAnm	TETRA algorithm nm
TE	TETRA algorithm for enhanced security on SIM-ME interface
TP	Transfer layer Protocol
UNBLOCK CHV1/2	value to unblock CHV1/CHV2

3.3 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

4 SIM characteristics

Two physical types of SIM are specified. These are the "ID-1 SIM" (see ISO 7810 [1]) and the "Plug-in SIM" (see ENV 1375-1 [8]). In addition the concept of a virtual SIM is introduced (see definition in subclause 3.1). None of the physical characteristics defined below apply to a virtual SIM.

The dimensions and mechanical characteristics of both types of physical SIM shall be in accordance with ISO/IEC 7816-1 [4] and ISO/IEC 7816-2 [5] unless otherwise specified. The following additional requirements shall be applied to ensure proper operation in the TETRA environment.

4.1 Format and layout

The identification number as defined in EF_{ICCID} (see subclause 10.2.1) shall be present on the outside of the ID-1 card. The information on the outside of the plug-in SIM should include at least the individual account identifier and the check digit of the IC card Identification.

4.1.1 SIM

Format and layout of the ID-1 SIM shall be in accordance with ISO/IEC 7816-1 [4] and ISO/IEC 7816-2 [5].

The card shall have a polarization mark which indicates how the user should insert the card into the Mobile Equipment (ME).

The ME shall accept embossed ID-1 cards. The embossing shall be in accordance with ISO 7811-1 [2] and ISO 7811-3 [3]. The contacts of the ID-1 SIM shall be located on the front (embossed face, see ISO 7810 [1]) of the card.

4.1.2 Plug-in SIM

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

Annexes A.1 and A.2 of ISO 7816-1 [4] do not apply to the plug-in SIM.

Annex A of ISO 7816-2 [5] 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 [5] are replaced by the corresponding values of figure A.1.

4.1.3 Virtual SIM

The abstract concept of a virtual SIM has been introduced in this ETS to describe the operation of a fully functional radio which does not have a removable SIM. A virtual SIM can be used instead of, or as well as, a physical SIM. The functionality and data structure of the virtual SIM, if implemented, is similar to that of the physical SIMs. The virtual SIM is supported internally by the ME. The virtual SIM shall contain the data items contained in those files marked as mandatory in clause 10. The precise format and layout of data items in the virtual SIM are not defined.

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

ME: There need not be any contacting elements in positions C4 and C8. Contact C6 need not be provided.

SIM: Contacts C4 and C8 need not be provided by the SIM. Contact C6 shall not be bonded in the SIM.

4.3.2 Activation and deactivation

The ME shall connect, activate and deactivate the SIM in accordance with the operating procedures specified in ISO/IEC 7816-3 [6].

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

NOTE 1: Soft power switching is when the radio is powered down normally. This is in contrast to abnormal power down, for instance if the battery is removed during operation, so that the voltage sequence can not be respected. Soft power switching is not defined in the TETRA specification.

NOTE 2: It is recommended that whenever possible the deactivation sequence defined in ISO/IEC 7816-3 [6] should be followed by the ME on all occasions when the ME is powered down.

If the SIM clock is already stopped and is not restarted, the ME is allowed to deactivate all the contacts in any order, provided that all signals reach low level before Vcc leaves high level. If the SIM clock is already stopped and is restarted before the deactivation sequence, then the deactivation sequence specified in ISO/IEC 7816-3 [6] subclause 5.4 shall be followed.

4.3.3 Inactive contacts (contact conditions in the ME switched-off state)

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

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

4.4 Precedence (multiple SIM operation)

A ME may make provision for the presence of multiple SIM cards. However only one SIM card shall be "active" at any one time.

NOTE 1: In a ME, which accepts both an ID-1 SIM and a plug-in SIM, it is suggested that the normal default should be that, if it is present, the ID-1 SIM takes precedence over the plug-in SIM. The default precedence setting may optionally be set by the user or by a suitably authorized manager using the Man Machine Interface (MMI).

NOTE 2: For other MEs, which accepts only an ID-1 physical SIM, the normal default may be that the ID-1 SIM takes precedence over the virtual SIM, if it is activated. The default precedence setting may optionally be set by the user or by a suitably authorized manager using the MMI. If a virtual SIM is not activated (either by manufacturer implementation or network operator choice) or has been de-activated by the SwMI, then the operation of the ME will be severely restricted. Usually an ME without an active SIM (either physical or virtual) cannot make or receive calls nor participate in Mobility Management (MM) functions such as registration or authentication.

4.5 Static protection

The ME manufacturer shall take adequate precautions (in addition to the protection diodes inherent in the SIM) to safeguard the ME, SIM and SIM/ME interface from static discharges at all times, and particularly during SIM insertion into the ME.

5 Electronic signals and transmission protocols

Electronic signals and transmission protocols shall be in accordance with ISO/IEC 7816-3 [6] unless specified otherwise. The following additional requirements shall be applied to ensure proper operation in the TETRA environment.

The choice of the transmission protocol(s), to be used to communicate between the SIM and the ME, shall at least include that specified and denoted by T = 0 in ISO/IEC 7816-3 [6].

As an option the T = 1 protocol specified in ISO/IEC 7816-3 [6] may also be supported.

5.1 Supply voltage Vcc (contact C1)

5 V and 3 V technology SIMs are defined in subclause 3.1.

5.1.1 5 V technology SIM

The TETRA SIM shall operate on 5 V ±10 % according to GSM 11.11 (ETS 300 608) [14]. The electrical characteristics of Vcc and Icc under normal and transient operating conditions are defined in GSM 11.11 (ETS 300 608) [14], subclause 5.1.

5.1.2 3 V technology SIM

The SIM shall operate on both 5 V ± 10 % according to GSM 11.11 (ETS 300 608) [14], and on 3 V ± 10 % according to GSM 11.12 (ETS 300 641) [15]. If the ME supplies 5 V to the SIM, both the ME and the SIM shall operate according to GSM 11.11 (ETS 300 608) [14]. The logical operation of the 3 V technology SIM shall be as defined in GSM 11.11 (ETS 300 608) [14].

Clock stop mode shall be supported by the SIM. The SIM shall indicate "Clock Stop Allowed" in the file characteristics of the status information as specified in GSM 11.11 (ETS 300 608) [14].

5.1.3 3 V technology SIM identification

The 3 V technology SIM shall contain an identification. The identification is coded on bit 5 in byte 14 of the status information (see subclause 9.2.1) as follows:

"0" : 5 V only SIM;

"1" : 3 V technology SIM.

In the case that the ME offers full compatibility by being able to operate the SIM interface at both 3 V and 5 V, then bit 5 in byte 14 of the status information, when set to "1", indicates that the SIM may be operated at 3 V.

The procedure for deriving the identification bit shall be performed by the ME immediately after the Answer To Reset (ATR) and before issuing any other command. The procedure consists of the two commands "SELECT TETRA" and "STATUS/GET RESPONSE".

5.1.4 3 V technology ME

The 3 V technology ME shall initially activate the SIM with 5 V according to GSM 11.11 (ETS 300 608) [14]. If the SIM indicates 3 V operation as defined in subclause 5.1.3, the ME may switch to 3 V operation as defined in subclause 5.1.7. If switching is performed it shall take place before issuing any further commands.

5.1.5 3 V Only ME

The 3 V only ME activates the SIM at 3 V.

If the ME is able to detect a 5 V only SIM according to the procedure in subclause 5.1.3, or if the procedure cannot be completed, the ME shall deactivate and reject the SIM immediately (maximum of 5 s) without issuing any further command. This rejection ensures that a SIM which appears to operate successfully during these early procedures is not allowed to continue further into the TETRA session where it may subsequently give unreliable operation at 3 V.

5.1.6 Activation and deactivation of 3 V technology SIM

The ME shall connect, activate and deactivate the 3 V SIM in accordance with the operating procedures specified in GSM 11.11 (ETS 300 608) [14] taking into account the electrical characteristics specified in GSM 11.12 (ETS 300 641) [15], clause 5 . In particular, Vcc is powered when it has a value between 2,7 V and 3,3 V.

5.1.7 Supply voltage switching

MEs supporting both 3 V and 5 V operation may switch between the two supply voltages. Switching shall always be performed by deactivating the SIM and activating it at the new supply voltage. Activation and deactivation of the SIM with 5 V shall be according to GSM 11.11 (ETS 300 608) [14], whereas activation and deactivation of the SIM with 3 V shall be according to GSM 11.12 (ETS 300 641) [15].

5.1.8 Cross compatibility

Cross compatibility means that the ME supports 3 V and 5 V operation. This is, however, optional for the ME. In case of the 3 V technology ME, full cross compatibility is provided, whereas, a 3 V only ME requires a 3 V technology SIM for operation. However, the 3 V technology SIM (see definitions) ensures full cross compatibility.

5.1.9 Technology outlook

Due to technology development it is possible in the future, when sub-micron technology is introduced, that ICs used in MEs may not withstand the 5 V supply voltage. This may, in particular, be the case for ICs operating in the power supply range of 1,5 V to 3,6 V. It may therefore be necessary in the future to specify a low voltage only SIM interface.

NOTE: When a low voltage only SIM is inserted into an ME which is supplying 5 V, the SIM may be destroyed. In some cases this could cause permanent damage to the ME. Precautions should be taken by the IC manufacturers to prevent the low voltage ICs from being damaged at 5 V.

5.2 Reset (RST) (contact C2)

For 5 V operation the ME shall operate the SIM within the limits defined in GSM 11.11 (ETS 300 608) [14], table 2.

For 3 V operation the ME shall operate the SIM within the limits defined in GSM 11.12 (ETS 300 641) [15], table 3.

5.3 Programming voltage Vpp (contact C6)

The SIM need not provide contact C6. If the SIM provides contact C6, then contact C6 shall not be connected.

5.4 Clock CLK (contact C3)

The clock shall be supplied by the ME. No "internal clock" SIMs shall be used.

When supplied with the 5 V ±10 % supply voltage, as specified in GSM 11.11 (ETS 300 608) [14], the SIM shall support 1 MHz to 5 MHz clock frequency operation.

When supplied with the 3 V ±10 % supply voltage, as specified in GSM 11.12 (ETS 300 641) [15], the SIM shall be operated with a clock frequency of 1 MHz to 4 MHz.

The required electrical characteristics of the ME clock for 5 V operation are defined in GSM 11.11 (ETS 300 608) [14]. For 3 V operation table 3 in GSM 11.11 (ETS 300 608) [14] is replaced by table 2 in GSM 11.12 (ETS 300 641) [15].

5.5 Input/Output (I/O) (contact C7)

For 5 V operation the electrical characteristics of the I/O (contact C7) are defined in GSM 11.11 (ETS 300 608) [14], subclause 5.5 (including table 4).

For 3 V operation the electrical characteristics of the I/O (contact C7) are defined in GSM 11.12 (ETS 300 641) [15], table 1.

5.6 States

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

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

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

The SIM may support a clock stop mode. The clock shall only be switched off subject to the conditions specified in the directory characteristics (see subclause 8.17.2).

Clock stop mode:

An ME shall wait at least 1 860 clock cycles after having received the last character of the response, including guard time (744 clock cycles), before it switches off the clock (if it is allowed to do so). It shall wait at least 744 clock cycles before it sends the first command after having started the clock.

5.7 Baud rate

The baud rate for all communications shall be as defined in GSM 11.11 (ETR 300 608) [14] subclause 5.7.

5.8 Answer To Reset (ATR)

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

The table explaining the structure and content of the ATR characters (as specified in ISO/IEC 7816-3 [6]) and the requirements for their use in TETRA, follows that for GSM as defined in GSM 11.11 (ETS 300 608) [14].

The protocol type selection procedures and speed selection procedures defined in GSM 11.11 (ETS 300 608) [14] shall apply to the TETRA SIM.

5.9 Bit/character duration and sampling time

The bit/character duration and sampling time specified in ISO/IEC 7816-3 [6], subclauses 6.1.1 and 6.1.2, are valid for all communications.

5.10 Error handling

If an ATR is corrupted or not received by the ME, error handling according to subclause 5.10 of GSM 11.11 (ETS 300 608) [14] shall apply.

6 Logical model

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

6.1 General description

Figure 1 shows the general structural relationships which may exist between files. The files are organized 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.

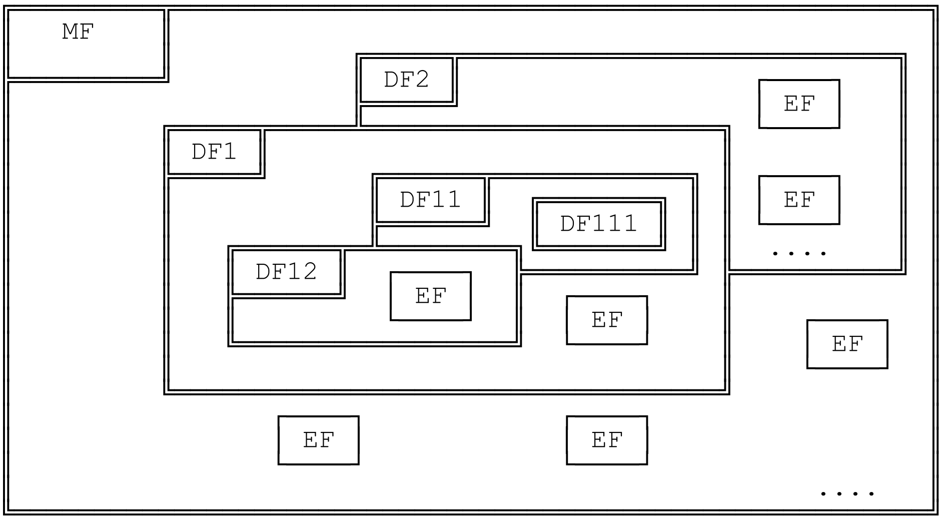


Figure 1: Organization of memory

Files are composed of a header, which is internally managed by the SIM, 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, and for TETRA is:

- '3F' = MF;
- '7F' = DF;
- '6F' = EF under a MF;
- '2F' = EF under a DF.

File IDs shall be subject to the following conditions:

- the file ID shall be assigned at the time of 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 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" only of a header part.

The DF defined in this ETS is DF_{TETRA} which contains the application for TETRA.

The file is an immediate child of the MF and may coexist on a multi-application card.

6.4 Elementary Files (EF)

An EF is composed of a header and a body part. The following subclauses give the three structures of an EF are used by TETRA.

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 00'. The total data length of the body of the EF is indicated in the header of the EF.

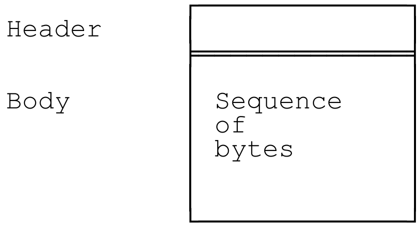


Figure 2: Structure of a transparent EF

6.4.2 Linear fixed EF

An EF with linear fixed structure consists of a sequence of 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.

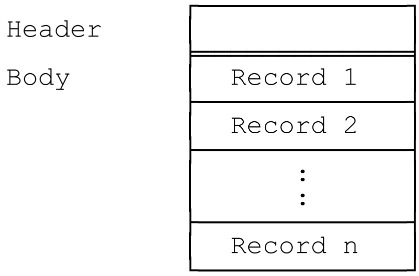


Figure 3: Structure of a linear fixed file

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 Key EF

A key EF consists of a sequence of 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.

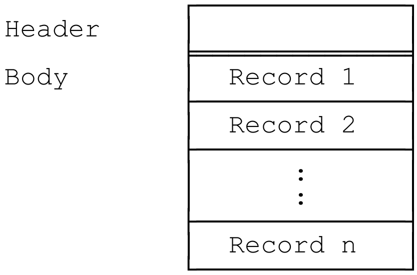


Figure 4: Structure of a key file

Records in an EF of this type are accessed using the absolute record number.

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

6.4.4 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.

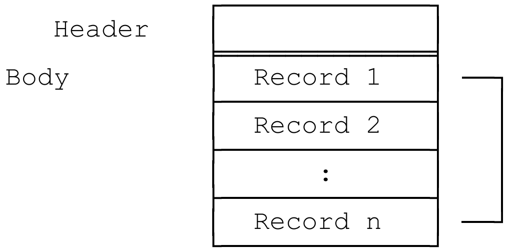


Figure 5: Structure of a cyclic file

For update operations only PREVIOUS record shall be used. For reading operations, the methods of addressing are "NEXT", "PREVIOUS", "CURRENT" and "RECORD NUMBER".

After selection of a cyclic file (for either operation), the record pointer shall address 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 a file

After the ATR, the MF is implicitly selected and becomes the "Current Directory". Each file may then 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 TETRA application. TETRA defines only one level of DFs under the MF.

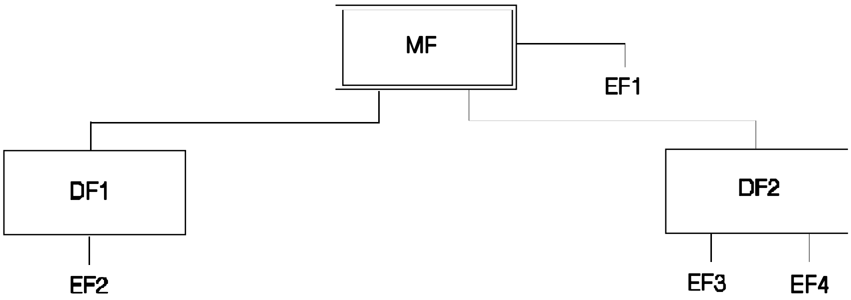


Figure 6: Logical structure

The table 1 gives the valid selections for TETRA for the logical structure in figure 6. Reselection of the last selected file is also allowed but not shown.

Table 1: File selection

Last selected file	Valid Selections
MF	DF1, DF2, EF1
DF1	MF, DF2, EF2
DF2	MF, DF1, EF3, EF4
EF1	MF, DF1, DF2
EF2	MF, DF1, DF2
EF3	MF, DF1, DF2, EF4
EF4	MF, DF1, DF2, EF3

6.6 Reservation of file IDs

In addition to the identifiers used for the files specified in this ETS, the following file IDs are reserved for use by TETRA.

DF:

- administrative use:

  '7F 8X';

- operational use:

  '7F 90' (DF_{TETRA}), and '7F 9X ', where X ranges from '2' to 'F'.

EFs:

- administrative use:

  '6F XX' in the DFs '7F 8X';
  '6F CX' in the DFs '7F90', '7F 9X';
  '2F XX ', in the MF '3F 00';

- operational use:

  '6F XX in '7F 90' and '7F 9X';
  '2F 1X' in the MF '3F 00'.

In all the above cases X ranges from '0' to 'F', unless otherwise stated.

The value 'FF FF' shall not be used.

NOTE: When choosing file IDs, care should be taken to avoid conflicts with IDs already used in other standards concerning IC cards for telecommunications use.

7 Security features

The security aspects of TETRA are described in ETS 300 392-7 [12] and ETS 300 396-6 [19]. This clause gives information related to security features supported by the SIM to enable the following:

- authentication of the subscriber identity to the network;
- data confidentiality over the air interface;
- confidentiality of air interface keys when passed over the SIM-ME interface;
- file access conditions.

7.1 Authentication and cipher key generation procedure

This subclause describes the authentication mechanism and cipher key generation which are invoked by the network and the SIM.

The names and parameters of the authentication algorithms supported by the SIM are defined in ETS 300 392-7 [12]. These are:

- algorithms TA11/TA12 to authenticate the SIM to the SwMI;

- algorithms TA21/TA22 to authenticate the SwMI to the SIM.

The cipher key generation algorithm supported by the SIM is defined in ETS 300 392-7 [12]. This is:

- algorithm TB4 to generate the Derived Cipher Key (DCK).

These algorithms may exist either discretely or combined within the SIM.

7.2 Support of Over The Air Re-keying (OTAR) distribution of cipher keys

The names and parameters of the OTAR algorithms supported by the SIM are defined in ETS 300 392-7 [12] and ETS 300 396-6 [19]. These are:

- algorithm TA32 to obtain the Common Cipher Key (CCK) from the Sealed CCK (SCCK);

- algorithm TA82 to obtain the Group Cipher Key (GCK) from the Sealed Group Cipher Key (SGCK);

- algorithm TA41/52 to obtain the Static Cipher Key (SCK) from the Sealed SCK (SSCK);

- algorithm TA71 to obtain the Modified Group Cipher Key (MGCK) from the GCK.

These algorithms may exist either discretely or combined within the SIM.

7.3 Support of SIM-ME enhanced security

Enhanced security for DCK, CCK, SCK and MGCK on the SIM-ME interface is supported by use of the TETRA algorithm for enhanced security on SIM-ME interface (TE) algorithm. When enhanced SIM-ME security is required (SIM Service 20 set):

- algorithm immediately following TB4 algorithm (see subclause 8.16.4);

- CCK, SCK and MGCK are sealed by the TE algorithm as part of the "Read Key" command (see subclause 8.6).

7.4 File access conditions

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.

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 access condition levels are defined in table 2.

Table 2: Access condition level coding

Level	Access Condition
0	ALWays
1	CHV1
2	CHV2
3	Reserved for Future Use (RFU)
4	RFU
5	AUTI
6 to 14	Reserved for Administrative Use (RAU)
15	NEVer

NEV: The action cannot be performed over the SIM/ME interface. The SIM may perform the action internally.

AUTI: The mobile can perform the action over the SIM/ME but only if it is the next action over the SIM/ME interface following a successful authentication.

So far as the SIM/ME interface is concerned the status AUTI shall be granted immediately following a successful authentication (as indicated by the return code on running the TA21/22 ALGORITHM). The AUTI status shall be withdrawn when there is subsequent activity over the SIM/ME interface. During an authentication message exchange, the ME may need to manage presence check procedures (for instance sending a SIM presence check immediately prior to initiating the authentication) so that these messages do not invalidate the AUTI status.

RAU: Allocation of these levels and the respective requirements for their fulfilment are the responsibility of the appropriate administrative authority (see also subclause 3.1).

CHV1: 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 SIM during the current card session;
- the CHV1 enabled/disabled indicator is set to "disabled";
- UNBLOCK CHV1 has been successfully performed during the current card session.

CHV2: The action shall only be possible if one of the following two conditions is fulfilled:

- a correct CHV2 value has already been presented to the SIM during the current card session;
- UNBLOCK CHV2 has been successfully performed during the current card session.

ALW: The action is always possible.

Condition levels are not hierarchical. For instance, correct presentation of CHV2 does not allow actions to be performed which require presentation of CHV1. A condition level which has been satisfied remains valid until the end of the TETRA session as long as the corresponding secret code remains unblocked, i.e. 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 ME shall determine whether CHV2 is available by using the response to the STATUS command. If CHV2 is "not initialized" then CHV2 commands, e.g. VERIFY CHV2, shall not be executable.

NOTE: The personalization phase of the card is normally agreed between the manufacturer and the network operator. Security during this phase is outside the scope of this ETS and needs to be carefully controlled.

7.5 Storage of CHV information

CHV information shall be stored in the relevant EF_{CHV}. For any file, the relevant EF_{CHV} is either a son, or if an EF_{CHV} does not exist there, the relevant EF_{CHV} of the parent file shall be used.

7.6 Storage of DCK

After successful authentication DCK shall be stored on the SIM for further use to unseal cipher keys but only for the duration of the TETRA session.

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 subclause 9.3.

Cards complying with this ETS shall support all functions described in this clause. In addition the command GET RESPONSE which is needed for the protocol T = 0 (specified in subclause 9.3) shall be supported.

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

Table 3: Functions which operate on files in a TETRA session

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

The commands and responses are defined in terms of where they obtain their inputs and where they place their outputs. The definitions of inputs and outputs to or from SIM, EF or ME are given in clause 3.

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 from ME: file ID.

Output to ME:

- 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, file size, access conditions, invalidated/not invalidated indicator, structure of EF and length of the records in case of linear fixed structure or cyclic structure.

8.2 STATUS

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

Input from ME: none.

Output to ME: file ID, total memory space available, CHV enabled/disabled indicator, CHV status.

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 from ME: relative address and the length of the string.

Output to ME: 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. An update can be considered as a replacement of the string already present in the EF by the string given in the update command.

Input from ME: relative address and the length of the string; string of bytes.

Output to ME: none.

8.5 READ RECORD

This function reads one complete record in the current linear fixed or cyclic EF. The record to be read is described by the modes below. 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.

Four modes are defined.

1) CURRENT:

the current record is read. The record pointer is not affected.

2) ABSOLUTE:

the record given by the record number is read. The record pointer is not affected.

3) 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 set the record pointer to the first record in this EF and this record shall be read.

4) 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 set the record pointer to the last record in this EF and this record shall be read.

Input from ME: mode, record number (absolute mode only) and the length of the record.

Output to ME: the record.

8.6 READ KEY

This function reads one complete record in the current key EF. The record to be read is described by the mode below.

ABSOLUTE:

the record given by the record number is read.

If SIM Service 20 is set (Enhanced SIM-ME security) the enhanced security algorithm TE shall be automatically run by the SIM to seal the OTAR keys (MGCK, CCK or SCK) with Enhanced Security Key (for protection of OTAR information on SIM-ME interface) (KE) before sending them to the ME.

Input from ME: record number and the length of the record.

Input from SIM: optionally KE (if SIM Service 20 is set).

Output to ME: the record (sealed by KE if service 20 is set).

8.7 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. Four 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 in a linear fixed EF, 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 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.

Input from ME:

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

- the data used for updating the record.

Output to ME: none.

8.8 SEEK

This function searches through the current linear fixed EF to find a record starting with 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.

The SIM shall be able to accept any pattern length from 1 to 16 bytes inclusive. The length of the pattern shall not exceed the record length.

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 from ME:

- type and mode;
- pattern;
- length of the pattern.

Output to ME:

- type 1: none;
- type 2: status/record number.

8.9 VERIFY CHV

This function verifies the CHV presented by the ME by comparing it with the relevant one stored in the SIM. 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 the last selected 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 from ME: indication CHV1/CHV2, CHV.

Output to ME: 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 from ME:	indication CHV1/CHV2, old CHV, new CHV.
Output to ME:	none.

8.11 DISABLE CHV

This function may only be applied to CHV1. The successful execution of this function has the effect that files protected by CHV1 are now accessible as if they were marked "ALWAYS". The function DISABLE CHV shall not be executed by the SIM when CHV1 is already disabled or blocked.

If the CHV1 presented is correct, the number of remaining CHV1 attempts shall be reset to its initial value 3 and CHV1 shall be disabled.

If the CHV1 presented is false, the number of remaining CHV1 attempts shall be decremented and CHV1 remains enabled. After 3 consecutive false CHV1 presentations, not necessarily in the same card session, CHV1 shall be blocked and the access condition can never be fulfilled until the UNBLOCK CHV function has been successfully performed on CHV1.

Input from ME:	CHV1.
Output to ME:	none.

8.12 ENABLE CHV

This function may only be applied to CHV1. It is the reverse function of DISABLE CHV. The function ENABLE CHV shall not be executed by the SIM when CHV1 is already enabled or blocked.

If the CHV1 presented is correct, the number of remaining CHV1 attempts shall be reset to its initial value 3 and CHV1 shall be enabled.

If the CHV1 presented is false, the number of remaining CHV1 attempts shall be decremented and CHV1 remains disabled. After 3 consecutive false CHV1 presentations, not necessarily in the same card session, CHV1 shall be blocked and the access condition can never be fulfilled until the UNBLOCK CHV function has been successfully performed on CHV1.

Input from ME:	CHV1.
Output to ME:	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 value of the CHV, presented together with the UNBLOCK CHV, is assigned to that 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 from ME: indication CHV1/CHV2, the UNBLOCK CHV and the new CHV.

Output to ME: 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 unless the file status of the EF indicates that READ and UPDATE may also be performed.

Input from ME: none.

Output to ME: none.

8.15 REHABILITATE

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

Input from ME: none.

Output to ME: none.

8.16 TETRA authentication algorithms

These functions support the air interface authentication applications of the Individual TETRA Subscriber Identity (ITSI).

The algorithms shall not be executable unless DF_{TETRA} has been selected as the Current Directory and a successful CHV1 verification procedure has been performed.

Security procedures internal to the SIM shall ensure that the authentication algorithms can only be run in the order specified in ETS 300 392-7 [12] and ETS 300 396-6 [19].

8.16.1 GET RANDOM

This function produces a random number for use in the authentication algorithms.

Input from ME: none.

Output to ME: RANDom challenge 2 (RAND2).

Output to SIM: RAND2.

The result RAND2 shall be stored internally on the SIM and also output to the ME for onward transfer to the SwMI. RAND2 shall be used as the input random number for the SIM initiated authentication procedure TA21/22.

8.16.2 TA11/12 ALGORITHM

This function, initiated by the SwMI, is used for authenticating the SIM to the TETRA network (SwMI).

Input from ME: RANDom challenge 1 (RAND1), Random Seed (RS).

Input from SIM: K.

Output to SIM: DCK1.

Output to ME: Response 1 (RES1).

NOTE: RES1 can be obtained from the SIM by use of the GET RESPONSE command.

8.16.3 TA21/22 ALGORITHM

This function, initiated by the SIM, is used for authenticating the TETRA network (SwMI) to the SIM.

Input from ME: Response 2 (RES2), RS.

Input from SIM: K, RAND2.

Output to SIM: DCK2.

NOTE: The ME is informed about the success of the operation [R2] via the status conditions returned by the SIM (see also subclause 9.4.4).

8.16.4 TB4/TE ALGORITHM

This function is used to obtain the DCK from its two parts DCK1 and DCK2 by use of the specified algorithm TB4. If SIM Service 20 is set (enhanced SIM-ME security) the enhanced security algorithm TE is automatically run by the SIM to seal DCK with KE before sending it to the ME.

Input from SIM: DCK1, DCK2, optionally KE (if SIM Service 20 is set).

Output to SIM: DCK.

Output to ME: DCK (sealed by KE if service 20 is set).

In the case of mutual authentication (SIM< = >SwMI) the inputs DCK1 and DCK2 shall be obtained internally from the TA11/12 and TA21/22 algorithms respectively. In the case of unilateral authentication, either DCK1 or DCK2 shall be set to zero; for SIM authentication DCK2 = 0; for SwMI authentication DCK1 = 0.

8.17 OTAR algorithms

These algorithms support the distribution of sealed cipher keys over the radio air interface using the OTAR procedures defined in ETS 300 392-7 [12] and ETS 300 396-6 [19].

The algorithms shall not be executable unless DF_{TETRA} has been selected as the Current Directory and a successful CHV1 verification procedure has been performed.

Security procedures internal to the SIM shall ensure that the OTAR algorithms can only be run in the order specified in ETS 300 392-7 [12] and ETS 300 396-6 [19].

8.17.1 TA32 ALGORITHM

This function is used to obtain the CCK from the SCCK by use of the specified algorithm TA32. The SCCK can be delivered to the ME in sealed format by an OTAR procedure. The SCCK shall be unsealed on the SIM and the CCK stored on the SIM for subsequent use in the ME.

Input from ME: SCCK, CCK-id.

Input from SIM: DCK.

Output to EF: CCK, CCK-id.

Output to ME: None.

8.17.2 TA82 ALGORITHM

This function is used to obtain the GCK from its input parts Group Tetra Subscriber Identity (GTSI), SGCK, GCK Version Number (GCK-VN) and DCK by use of the specified algorithm TA82. The GCK can be delivered to the ME in sealed format together with the appropriate GTSI and GCK-VN by an OTAR procedure. The SGCK shall be unsealed on the SIM and the GCK stored on the SIM for subsequent use in the ME.

Input from ME: Record number in EF_{GCK}, GTSI, SGCK, GCK-VN.

Input from SIM: DCK.

Output to EF: GCK (to EF_{GCK}), GCK-VN (to EF_{MGCK}).

Output to ME: None.

NOTE: GCKs are not accessible over the SIM-ME interface. Following the download of a new GCK, algorithm TA71 (see subclause 8.17.4) is run to update the associated MGCK.

8.17.3 TA41/52 ALGORITHM

This function is used to obtain the SCK from the SSCK which may be distributed by OTAR. The SSCKs shall be unsealed on the SIM and the SCK stored on the SIM for subsequent use in the ME.

Input from ME: SSCK, SCK-VN, Random Seed for OTAR (RSO).

Input from SIM: K.

Output to EF: SCK, SCK-VN.

Output to ME: None.

The SCK shall be stored in EF_{SCK} in the record number indicated by the SCK Number (SCKN).

8.17.4 TA71 ALGORITHM

This function shall be used to obtain the MGCK from the GCK and the CCK by use of the specified algorithm TA71. The algorithm shall be run whenever a new GCK is distributed or when a new CCK is issued (for instance caused by entering a new location area).

Input from ME: Record number in EF_{MGCK}, record number in EF_{CCK} to be used.

Input from EF: GCK, CCK.

Output to EF: MGCK.

Output to ME: 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 (APDU) 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 general format:

CLA	INS	P1	P2	P3	Data

The response APDU has the following general 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 [6]).

The bytes have the following meaning:

- CLA is the class of instruction (ISO/IEC 7816-3 [6]), 'A0' is used in the TETRA application;

- INS is the instruction code (ISO/IEC 7816-3 [6]) as defined in this subclause for each command;

- P1, P2, P3 are parameters for the instruction. They are specified in table 4. 'FF' is a valid value for P1, P2 and P3. P3 gives the length of the data element. P3 = '00' introduces a 256 byte data transfer from the SIM in an outgoing data transfer command (response direction). In an ingoing data transfer command (command direction), P3 = '00' introduces no transfer of data;

- 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 the 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.

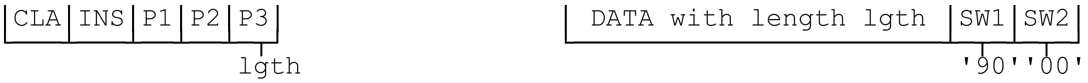
If the length of the response data is not known beforehand, then its correct length may be obtained by applying the first command and interpreting the status words. SW1 shall be '9F' and SW2 shall give the total length of the data. Other status words may be present in case of an error. The various cases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

Case 1: No input/No output

CLA	INS	P1	P2	P3
				lgth (='00')

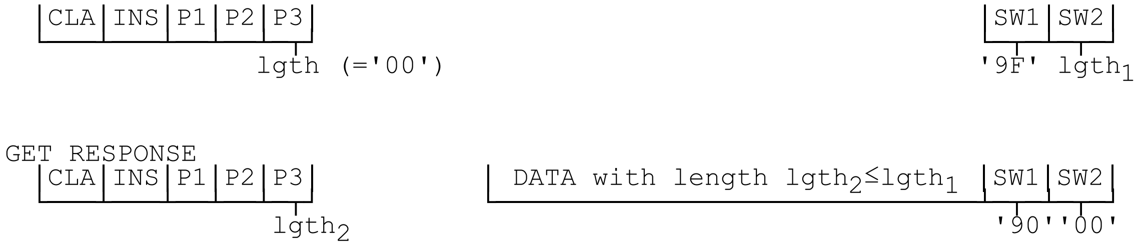
SW1	SW2
'90'	'00'

Case 2: No input/Output of known length

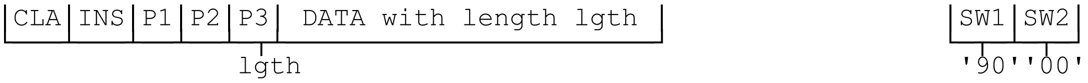


NOTE: lgth = '00' causes a data transfer of 256 bytes.

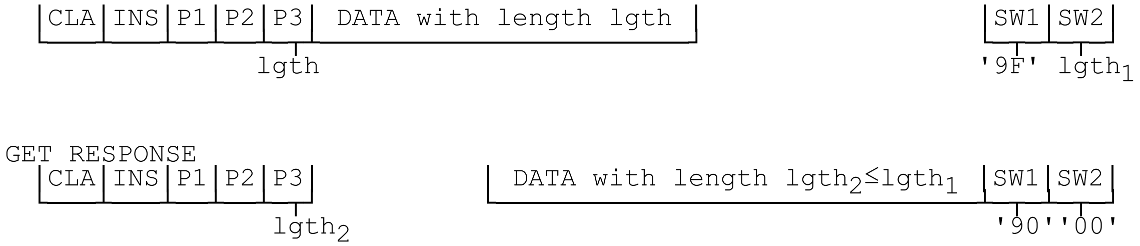
Case 3: No Input/Output of unknown length



Case 4: Input/No output



Case 5: Input/Output of known or unknown length



For cases 3 and 5, when Status Word 1/Status Word 2 (SW1/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.

If the TETRA application is one of several applications in a multi-application card, other commands with CLA not equal to 'A0' may be sent by the terminal. This shall not influence the state of the TETRA application.

9.2 Coding of the commands

Table 4 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 ME while (R) stands for data received by the ME. Offset is coded on 2 bytes where P1 gives the high order byte and P2 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.

In addition to the instruction codes specified in table 4 the following codes are reserved:

- TETRA operational phase:
  - '7X' with X even.
- Administrative management phase:
  - '22', '2E', '38, '3A', '3C', '3E', '58', '5A', '5C' and '5E'.

Table 4: Coding of the commands

COMMAND	INS	P1	P2	P3	S/R
SELECT	'A4'	'00'	'00'	'02'	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'	'00'	type/mode	lgth	S/R
READ KEY	'BE'	rec No.	'04'	lgth	R
VERIFY CHV	'20'	'00'	CHV No.	'08'	S
CHANGE CHV	'24'	'00'	CHV No.	'10'	S
DISABLE CHV	'26'	'00'	'01'	'08'	S
ENABLE CHV	'28'	'00'	'01'	'08'	S
UNBLOCK CHV	'2C'	'00'	see note	'10'	S
INVALIDATE	'04'	'00'	'00'	'00'	-
REHABILITATE	'44'	'00'	'00'	'00'	-
GET RANDOM	'CE'	'00'	'00'	'0A'	R
TA11/12 ALGORITHM	'40'	'00'	'00'	'14'	S/R
TA21/22 ALGORITHM	'42'	'00'	'00'	'0E'	S
TB4/TE ALGORITHM	'46'	'00'	'00'	'0A'	R
TA32 ALGORITHM	'48'	'00'	'00'	'11'	S
TA82 ALGORITHM	'4A'	rec No.	'00'	'17'	S
TA41/52 ALGORITHM	'4C'	'00'	'00'	'1B'	S
TA71 ALGORITHM	'4E'	target record no	input record no.	'0A'	-
GET RESPONSE	'C0'	'00'	'00'	lgth	R

NOTE: If the UNBLOCK CHV command applies to CHV1 then P2 is coded '00'; if it applies to CHV2 then P2 is coded '02'.

Definitions and codings used in the response parameters/data of the commands are given in subclause 9.2.23.

9.2.1 SELECT

COMMAND	CLASS	INS	P1	P2	P3
SELECT	'A0'	'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	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 - 12	RFU	5
13	Length of the following data (byte 14 to the end)	1
14 - 34	TETRA specific data	21

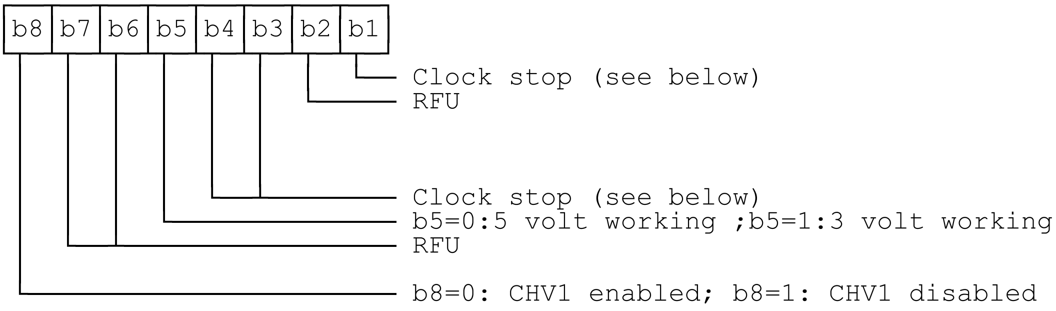
TETRA specific data:

Byte(s)	Description	Length
14	File characteristics (see detail 1)	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 2)	1
20	UNBLOCK CHV1 status (see detail 2)	1
21	CHV2 status (see detail 2)	1
22	UNBLOCK CHV2 status (see detail 2)	1
23	RFU	1
24 - 34	Reserved for the administrative management (optional)	0≤lgth≤11

NOTE 1: Byte 35 and following are RFU.

NOTE 2: The STATUS information of the MF and DF_{TETRA} provide some identical application specific data, e.g. CHV status. On a multi-application card the MF should not contain any application specific data. Such data is obtained by terminals from the specific application directories. ME manufacturers should take this into account and therefore not use application specific data which may exist in the MF of a mono-application SIM.

Detail 1: File characteristics.


b8
b7
b6
b5
b4
b3
b2
b1
Clock stop (see below)
RFU
Clock stop (see below)
b5=0:5 volt working ;b5=1:3 volt working
RFU
b8=0: CHV1 enabled; b8=1: CHV1 disabled

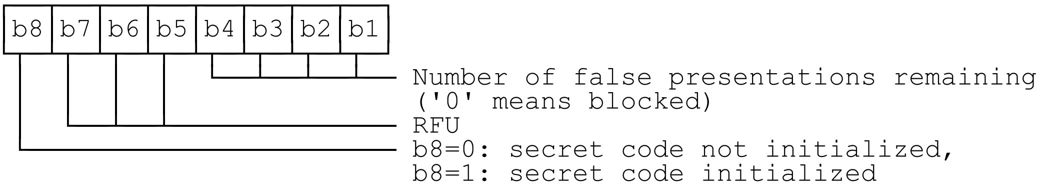
The coding of the conditions for stopping the clock is as follows:

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 not allowed, unless at high level;
0	0	1	clock stop not allowed, unless at 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 2: Status byte of a secret code.


b8
b7
b6
b5
b4
b3
b2
b1
Number of false presentations remaining
('0' means blocked)
RFU
b8=0: secret code not initialized,
b8=1: secret code initialized

Response parameters/data in case of an EF:

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, cyclic or key EF: record length multiplied by the number of records of the EF)	2
5 - 6	File ID	2
7	Type of file (see subclause 9.3)	1
8	See detail 3	1
9 - 11	Access conditions (see subclause 9.3)	3
12	File status (see subclause 9.3)	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 4)	1
NOTE: Byte 16 and following are RFU.		

Detail 3: Byte 8.

- For transparent, linear fixed and key EFs this byte is RFU. For a cyclic EF all bits except bit 7 are RFU; b7 = 1 indicates that the INCREASE command is allowed on the selected cyclic file.

Detail 4: Byte 15.

- For cyclic, linear fixed and key 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 SIM.

9.2.2 STATUS

COMMAND	CLASS	INS	P1	P2	P3
STATUS	'A0'	'F2'	'00'	'00'	lgth

The response parameters/data are identical to the response parameters/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	'A0'	'B0'	offset high	offset low	lgth

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	'A0'	'D6'	offset high	offset low	lgth

Command parameters/data:

Byte(s)	Description	Length
1 - lgth	Data	lgth

9.2.5 READ RECORD

COMMAND	CLASS	INS	P1	P2	P3
READ RECORD	'A0'	'B2'	Rec.No.	Mode	lgth

Parameter P2 specifies the mode:

- '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.

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	'A0'	'DC'	Rec.No.	Mode	lgth

Parameter P2 specifies the mode:

- '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.

Command parameters/data:

Byte(s)	Description	Length
1 - lgth	Data	lgth

9.2.7 READ KEY

COMMAND	CLASS	INS	P1	P2	P3
READ RECORD	'A0'	'BE'	rec No.	'04'	lgth

Response parameters/data:

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

9.2.8 SEEK

COMMAND	CLASS	INS	P1	P2	P3
SEEK	'A0'	'A2'	'00'	Type/Mode	lgth

Parameter P2 specifies type and mode:

- 'x0' = from the beginning forward;
- 'x1' = from the end backward;
- 'x2' = from the next location forward;
- 'x3' = from the previous location backward;

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

9.2.9 VERIFY CHV

COMMAND	CLASS	INS	P1	P2	P3
VERIFY CHV	'A0'	'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	'A0'	'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	'A0'	'26'	'00'	'01'	'08'

Command parameters/data:

Byte(s)	Description	Length
1 - 8	CHV1 value	8

9.2.12 ENABLE CHV

COMMAND	CLASS	INS	P1	P2	P3
ENABLE CHV	'A0'	'28'	'00'	'01'	'08'

Command parameters/data:

Byte(s)	Description	Length
1 - 8	CHV1 value	8

9.2.13 UNBLOCK CHV

COMMAND	CLASS	INS	P1	P2	P3
UNBLOCK CHV	'A0'	'2C'	'00'	CHV No.	'10'

Parameter P2 specifies the CHV:

- 00 = CHV1;
- 02 = CHV2.

NOTE: The coding '00' for CHV1 differs from the coding of CHV1 used for other commands.

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	'A0'	'04'	'00'	'00'	'00'

9.2.15 REHABILITATE

COMMAND	CLASS	INS	P1	P2	P3
REHABILITATE	'A0'	'44'	'00'	'00'	'00'

9.2.16 GET RANDOM

COMMAND	CLASS	INS	P1	P2	P3
GET RANDOM	'A0'	'CE'	'00'	'00'	'0A'

Response parameters/data:

Byte(s)	Description	Length
1 - 10	RAND2	10

9.2.17 TA11/12 ALGORITHM

COMMAND	CLASS	INS	P1	P2	P3
TA11/12 ALGORITHM	'A0'	'40'	'00'	'00'	'14'

Command parameters/data:

Byte(s)	Description	Length
1 - 10	RAND1	10
11 - 20	RS	10

Response parameters/data:

Byte(s)	Description	Length
4	RES1	4

See ETS 300 392-7 [12], subclause 4.4.8.27 for use of RES1 and subclause 4.6 for size of the cryptographic parameters.

9.2.18 TA21/22 ALGORITHM

COMMAND	CLASS	INS	P1	P2	P3
TA21/22 ALGORITHM	'A0'	'42'	'00'	'00'	'0E'

Command parameters/data:

Byte(s)	Description	Length
1 - 4	RES2	4
5 - 14	RS	10

9.2.19 TB4/TE ALGORITHM

COMMAND	CLASS	INS	P1	P2	P3
TB4 ALGORITHM	'A0'	'46'	'00'	'00'	'0A'

Response parameters/data:

Byte(s)	Description	Length
1 - 10	DCK	10

9.2.20 TA32 ALGORITHM

COMMAND	CLASS	INS	P1	P2	P3
TA32 ALGORITHM	'A0'	'48'	'00'	'00'	'11'

Command parameters/data:

Byte(s)	Description	Length
1 - 15	SCCK	15
16 - 17	CCK-id	2

9.2.21 TA82 ALGORITHM

COMMAND	CLASS	INS	P1	P2	P3
TA82 ALGORITHM	'A0'	'4A'	'Rec.No.'	'00'	'17'

- P1 specifies the target record number of the record within the EF_{GCK}.

Command parameters/data:

Byte(s)	Description	Length
1 - 6	GTSI	6
7 - 21	SGCK	15
22 - 23	GCK-VN	2

9.2.22 TA41/52 ALGORITHM

COMMAND	CLASS	INS	P1	P2	P3
TA41/52 ALGORITHM	'A0'	'4C'	'00'	'00'	'1B'

Command parameters/data:

Byte(s)	Description	Length
1 - 15	SSCK	15
16 - 17	SCK-VN	2
18 - 27	RSO	10

9.2.23 TA71 ALGORITHM

COMMAND	CLASS	INS	P1	P2	P3
TA71 ALGORITHM	'A0'	'4E'	target record	input record	'00'

Parameter P1 specifies the target record number in EF_{MGCK}:

P1 = '00' Update all MGCKs;
P1 ≠ '00' Parameter P1 gives the record number to be updated.

Parameter P2 specifies the record number (1 or 2) in EF_{CHV} from which the CCK shall be retrieved:

P2 = '01' or '02' and denotes the record number in EF_{CHV}.

9.2.24 GET RESPONSE

COMMAND	CLASS	INS	P1	P2	P3
GET RESPONSE	'A0'	'C0'	'00'	'00'	lgth

The response data depends on the preceding command. Response data is available after the commands TA11/12 ALGORITHM, SEEK (type 2) and SELECT. 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 SIM 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 activation of the SIM, GET RESPONSE is also allowed as the first command after activation.

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

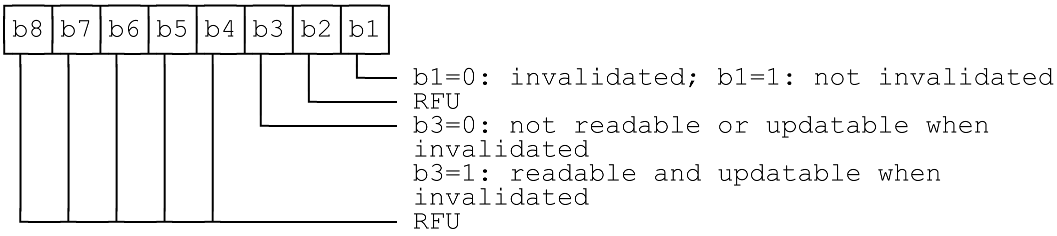
9.3 Definitions and coding

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

Coding: Each byte is represented by bits b8 to b1, where b8 is the Most Significant Bit (MSB) and b1 is the Least Significant Bit (LSB). In each representation the leftmost bit is the MSB.

RFU: In a TETRA specific card all bytes which are RFU shall be set to '00' and RFU bits to 0. Where the TETRA application exists on a multi-application card or is built on a generic telecommunications card (e.g. TE9) 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 an ME in a TETRA session.

File status:


b8
b7
b6
b5
b4
b3
b2
b1
b1=0: invalidated; b1=1: not invalidated
RFU
b3=0: not readable or updatable when invalidated
b3=1: readable and updatable when invalidated
RFU

Bit b3 may be set to 1 in special circumstances when it is required that the EF can be read and updated even if the EF is invalidated, e.g. reading and updating the EF_{ADN} when the Fixed Dialling Number (FDN) feature is enabled.

Structure of file:

- '00' transparent;
- '01' linear fixed;
- '03' cyclic;
- '11' key.

Type of File:

- '00' RFU;
- '01' MF;
- '02' DF;
- '04' EF.

Coding of CHVs and UNBLOCK CHVs

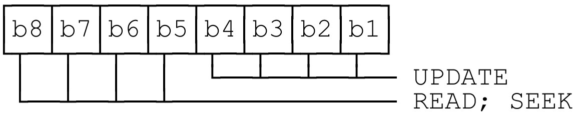
A CHV is coded on 8 bytes. Only (decimal) digits (0-9) shall be used, coded in CCITT Recommendation T.50 [18] with bit 8 set to zero. The minimum number of digits is 4. If the number of digits presented by the user is less than 8 then the ME shall pad the presented CHV with 'FF' before sending it to the SIM.

The coding of the UNBLOCK CHVs is identical to the coding of the CHVs. However, the number of (decimal) digits is always 8.

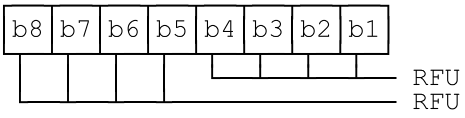
Coding of access conditions

The access conditions for the commands are coded on bytes 9 and 10 of the response data of the SELECT command. Each condition is coded on 4 bits as shown in table 2.

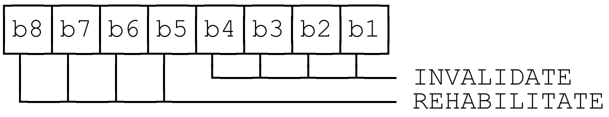
Byte 9:



Byte 10:



Byte 11:



9.4 Status conditions returned by the card

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'	command successful but after using an internal update 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 is inconsistent with the command

9.4.4 Security management

SW1	SW2	Error description
'98'	'02'	no CHV initialized
'98'	'04'	access condition not fulfilled unsuccessful CHV verification, at least one attempt left unsuccessful UNBLOCK CHV verification, at least one attempt left
'98'	'08'	in contradiction with CHV status
'98'	'10'	in contradiction with invalidation status
'98'	'40'	unsuccessful CHV verification, no attempt left unsuccessful UNBLOCK CHV verification, no attempt left CHV blocked UNBLOCK CHV blocked
'98'	'60'	manipulation flag set
'98'	'70'	SwMI authentication unsuccessful

9.4.5 Application independent errors

SW1	SW2	Error description
'67'	'XX'	incorrect parameter P3 (see note 3)
'6B'	'XX' (note 1)	incorrect parameter P1 or P2 (see note 2)
'6D'	'XX' (note 1)	unknown instruction code given in the command
'6E'	'XX' (note 1)	wrong instruction class given in the command
'6F'	'XX' (note 1)	technical problem with no diagnostic given

NOTE 1: These values of 'XX' are specified by ISO/IEC; at present the default value 'XX' = '00' is the only one defined.

NOTE 2: 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 3: 'XX' gives the correct length or states that no additional information is given ('XX' = '00').

9.4.6 Commands versus possible status responses

Table 5 shows for each command the possible status conditions returned (marked by an asterisk *).

Table 5: Commands and status words

	OK		Mem Sta-tus		Refer. Status				Security status							Application Independent Errors				
Commands	9000	9FXX	920X	9240	9400	9402	9404	9408	9802	9804	9808	9810	9840	9860	9870	67XX	6BXX	6DXX	6EXX	6FXX
Select		*		*			*									*	*		*	*
Status	*			*												*	*		*	*
Update Binary	*		*	*	*			*		*		*				*	*		*	*
Update Record	*		*	*	*	*		*		*		*				*	*		*	*
Read Binary	*			*	*			*		*		*				*	*		*	*
Read Record	*			*	*	*		*		*		*				*	*		*	*
Read Key	*			*	*	*		*		*		*				*	*		*	*
Seek	*	*		*	*		*	*		*		*				*	*		*	*
Verify CHV	*		*	*					*	*	*		*			*	*		*	*
Change CHV	*		*	*					*	*	*		*			*	*		*	*
Disable CHV	*		*	*					*	*	*		*			*	*		*	*
Enable CHV	*		*	*					*	*	*		*			*	*		*	*
Unblock CHV	*		*	*					*	*	*		*			*	*		*	*
Invalidate	*		*	*	*					*		*				*	*		*	*
Rehabilitate	*		*	*	*					*		*				*	*		*	*
Get Random	*			*																
TA11/12 Algorithm		*		*						*						*	*		*	*
TA21/22 Algorithm	*			*						*					*	*	*		*	*
TB4/TE Algorithm	*			*						*						*	*		*	*
TA32 Algorithm	*			*						*				*		*	*		*	*
TA82 Algorithm	*			*						*				*		*	*		*	*
TA41/52 Algorithm	*			*						*				*		*	*		*	*
TA71 Algorithm	*			*						*						*	*		*	*
Get Response	*			*												*	*		*	*

10 Contents of the EFs

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

EFs or data items having an unassigned value, or, which during the TETRA session, are cleared by the ME, shall have their bytes set to 'FF'. After the administrative phase all data items shall have a defined value or have their bytes set to 'FF'. If a data item is "deleted" during a TETRA session by the allocation of a value specified in another TETRA TS, then this value shall be used, and the data item is not unassigned.

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.

Using the command GET RESPONSE the ME can determine the length of variable length records (e.g. 1 to X).

NOTE: The field "Update activity" has only meaning to the card manufacturer to help choosing proper memory management for EFs. If an EF is updated very seldom, e.g. once during the administrative phase, it is set to "low". If an EF is updated or may be updated in every TETRA session it is set to "high". The actual update activity of certain EFs also depends on the system. Therefore the update activity of an EF is set to high if it may be updated frequently in some systems. For example, high security systems may want to update cipher keys frequently, but less secure systems may update keys only when a particular reason to do it arises.

10.1 Contents of EFs located either at application level or above

10.1.1 EF$_{CHV}$

EF$_{CHV}$ is a transparent file (MF, DF level).

Identifier: '0 000' (CHV1)	Structure: transparent	Mandatory
Identifier: '0 100' (CHV2)	Structure: transparent	Optional

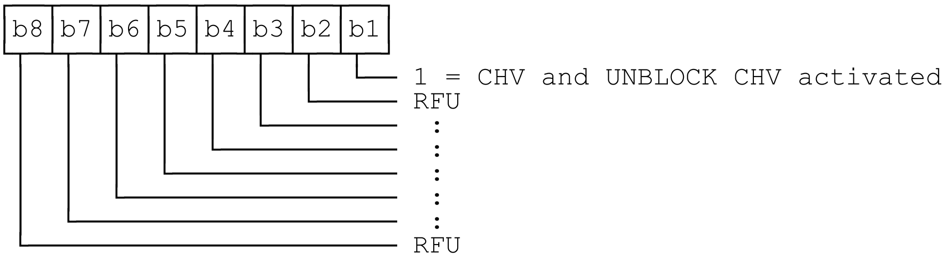
File size: 23 bytes	Update activity: low

Access Conditions:

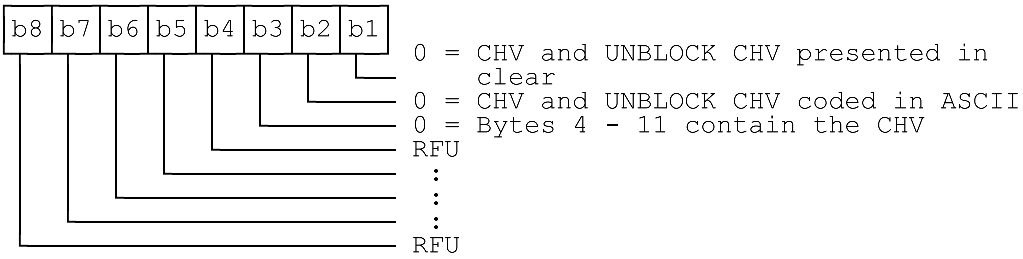
READ	NEV
UPDATE	NEV
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	EF$_{CHV}$ activation byte	M	1
2	Way to present the CHV/UNBLOCK CHV	M	1
3	Not used in the TETRA application - coded 'FF'.	M	1
4 - 11	CHV	M	8
12	CHV attempts Pre-set value N	M	1
13	Remaining CHV attempt counter	M	1
14 - 21	UNBLOCK CHV	M	8
22	Remaining UNBLOCK CHV attempt counter	M	1
23	Number of remaining UNBLOCK CHV mechanism use	M	1

Byte 1: EF$_{CHV}$ activation byte



Byte 2: Way to present the CHV / UNBLOCK CHV


b8
b7
b6
b5
b4
b3
b2
b1
0 = CHV and UNBLOCK CHV presented in clear
0 = CHV and UNBLOCK CHV coded in ASCII
0 = Bytes 4 - 11 contain the CHV
RFU
:
:
:
RFU

Byte 4-11: CHV, 4 to 8 decimal digits, coded in ASCII, right padded with 'FF'.

Byte 12: Pre-set value:

Byte 12 is set to '03'.

Byte 14-21: UNBLOCK CHV, 8 decimal digits, coded in ASCII.

Byte 23: Number of remaining UNBLOCK CHV mechanism use:

Byte 23 shall be coded 'FF'.

NOTE: This implies that the UNBLOCK CHV mechanism may be used an infinite number of times (subject to the correct value being entered).

10.2 Contents of the EFs at the MF level

There are four EFs at the MF level.

10.2.1 EF$_{ICCD}$ (Card Identification)

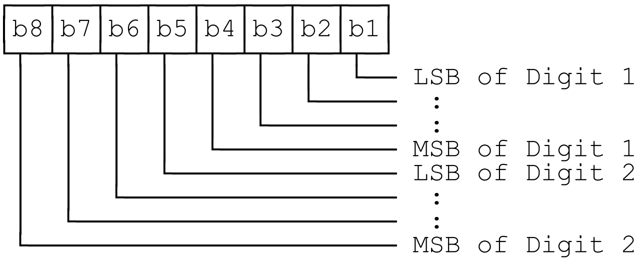
This EF provides a unique identification number for the SIM.

<table>
<tr><td colspan="2">Identifier: '2FE2'</td><td colspan="2">Structure: transparent</td><td>Mandatory</td></tr>
<tr><td colspan="2">File size: 10 bytes</td><td colspan="3">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ ALW
UPDATE NEV
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1 - 10</td><td colspan="2">Identification number</td><td>M</td><td>10</td></tr>
</table>

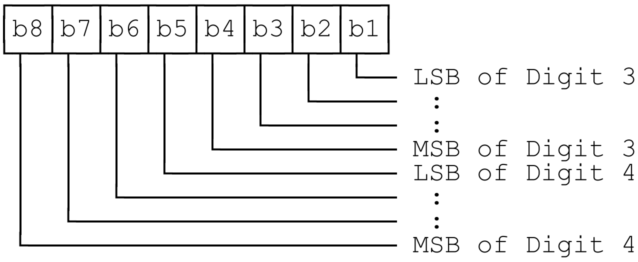
- Identification number:

  Contents: Card identification number according to CCITT Recommendation E.118 [21].
  Coding: Binary Coded Decimal (BCD), left justified and padded with 'F'.

Byte 1:



Byte 2:



etc.

10.2.2 EF_{DIR} (Application directory)

An EF containing a list of applications supported by the card, and optional related data elements defined in ISO 7816-5 [16]. EF_{DIR} need not exist on the Virtual SIM.

<table>
<tr><td colspan="2">Identifier: '2F00 '</td><td colspan="2">Structure: transparent</td><td>Mandatory</td></tr>
<tr><td colspan="3">File size: X bytes</td><td colspan="2">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ ALW/CHV1 (see note 1)
UPDATE ADM
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td colspan="2">Application identifier tag ('4F')</td><td>M</td><td>1</td></tr>
<tr><td>2</td><td colspan="2">Application identifier length</td><td>M</td><td>1</td></tr>
<tr><td>3</td><td colspan="2">Application identifier (see note 2)</td><td>M</td><td>1-16</td></tr>
<tr><td></td><td colspan="2">Application label tag ('50')</td><td>M</td><td>1</td></tr>
<tr><td></td><td colspan="2">Application label length</td><td>M</td><td>1</td></tr>
<tr><td></td><td colspan="2">Application label (verbal description)</td><td>M</td><td>0-16</td></tr>
<tr><td></td><td colspan="2">Path tag ('51')</td><td>M</td><td>1</td></tr>
<tr><td></td><td colspan="2">Path length</td><td>M</td><td>1</td></tr>
<tr><td></td><td colspan="2">Path</td><td>M</td><td>X</td></tr>
</table>

. . .

	Second application information		

NOTE 1: Access conditions to this file are defined during the administrative phase.

NOTE 2: Application identifiers are allocated by ETSI.

10.2.3 EF$_{LP}$ (Language Preference)

This EF contains the codes for one or more languages. This information, determined by the user/operator, defines the preferred languages of the user in order of priority. This information may be used by the ME for MMI purposes and for short message handling.

Identifier: '2F05 '	Structure: transparent	Mandatory
File size: 1-n bytes	Update activity: low	

Access Conditions:

READ	ALW
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	1st language code (highest priority)	M	1
2	2nd language code	O	1
n	nth language code (lowest priority)	O	1

Coding: As in GSM 03.38 (ETS 300 628) [17].

Using the command GET RESPONSE, the ME can determine the size of the EF.

10.3 Contents of the EFs at the TETRA application level

10.3.1 EF$_{SST}$ (SIM Service Table)

The purpose of this EF is to indicate which of the optional services and EFs are available.

NOTE: Having optional services indicated simplifies their handling for the ME.

Identifier: '6F01'	Structure: transparent	Mandatory
File size: 4 bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Services n°1 to no.8	M	1
2	Services n°9 to no.16	M	1
3	Services n°17 to no.24	M	1
4	Services n°25 to no.32	M	1

- Services:

Contents:

Service no.1:	CHV1 disable function
Service no.2:	ADNTETRA (Internal TETRA Phone Book) and Extension A
Service no.3:	ADN (External phones) and Extension 1 and Gateway table
Service no.4:	FDNTETRA and Extension B
Service no.5:	FDN and Extension 2 and Gateway table
Service no.6:	SDNTETRA
Service no.7:	SDN and Extension 3 and Gateway table
Service no.8:	LNDTETRA and Extension A
Service no.9:	LND and Extension 1 and Gateway table
Service no.10:	CHV2 disable function
Service no.11:	CCK and CCK location areas
Service no.12:	SCK
Service no.13:	GCK and MGCK
Service no.14:	Service Provider Name
Service no.15:	Preferred Networks
Service no.16:	Username
Service no.17:	Authentication algorithms TA11/12, TA21/22, TB4.
Service no.18:	OTAR unsealing algorithms TA32, TA82, TA41/52, TA71.
Service no.19:	RFU
Service no.20:	Enhanced SIM-ME
Service no.21:	RFU
Service no.22:	Status message texts
Service no.23:	SDS1 message texts
Service no.24:	SDS 123 Storage
Service no.25:	SDS 4 Storage
Service no.26:	Call Modifiers
Service no.27:	DMO channel information and MS allocation of DMO channels
Service no.28:	List of key holders
Service no.29:	DMO repeater and gateway list
Service no.30:	RFU
Service no.31:	RFU
Service no.32:	RFU

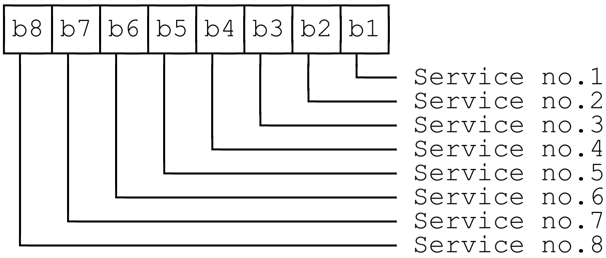
NOTE: Other services are possible in the future and will be coded on further bytes in the EF.

The coding falls under the responsibility of ETSI.

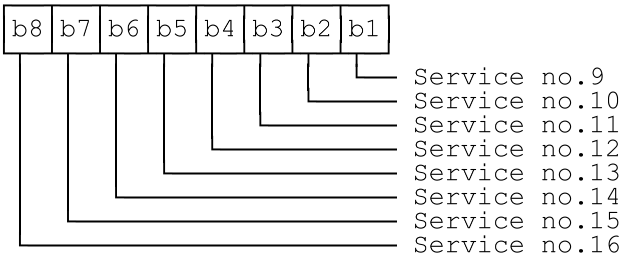
Coding:

1 bit is used to code each service:
bit = 1: service available
bit = 0: service not available

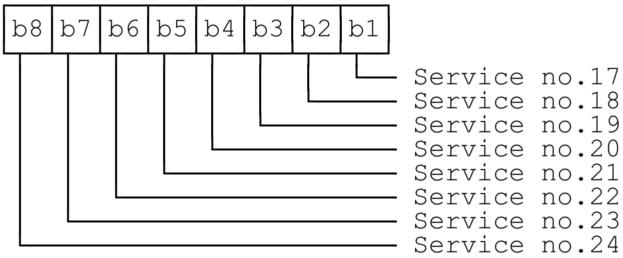
Byte 1:



Byte 2:

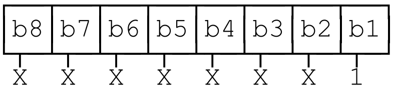


Byte 3:



etc.

The following example of coding for the first byte means that service no.1 "CHV1-Disabling" is available



10.3.2 EF$_{ITSI}$ (Individual Tetra Subscriber Identity)

This EF contains the Individual Tetra Subscriber Identity number (ITSI). This EF shall not readable or updateable when invalidated.

Identifier: '6F02'	Structure: transparent	Mandatory

File size: 6 bytes	Update activity: low

Access Conditions:

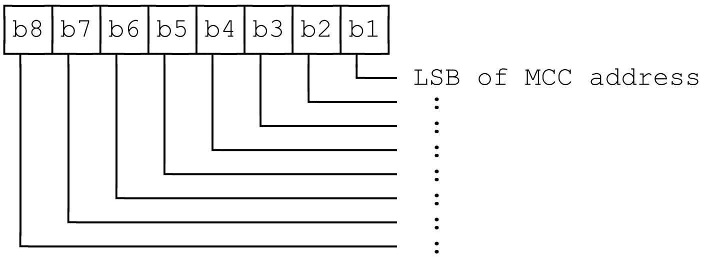
READ	CHV1
UPDATE	ADM
INVALIDATE	AUTI
REHABILITATE	NEV

Bytes	Description	M/O	Length
1 - 6	ITSI	M	6

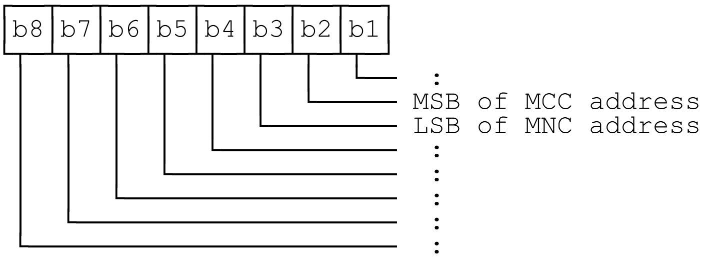
- ITSI:

Contents: ITSI consists of Mobile Country Code (MCC), Mobile Network Code (MNC) and Individual Short Subscriber Identity (ISSI).
Coding:

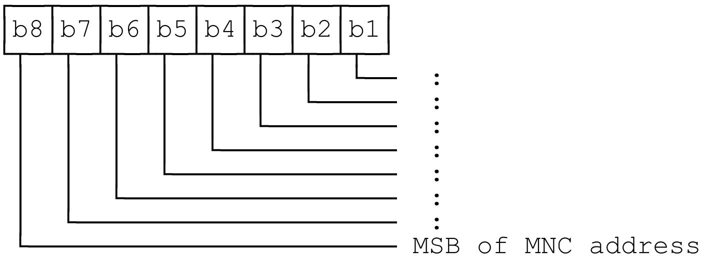
Byte 1:



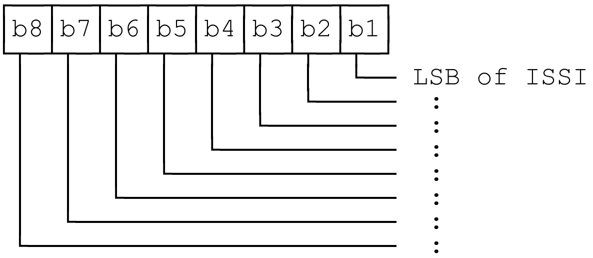
Byte 2:



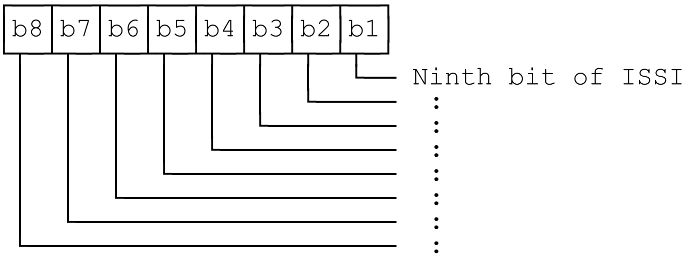
Byte 3:



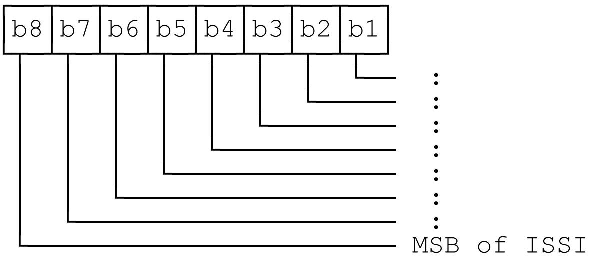
Byte 4:



Byte 5:



Byte 6:



NOTE: The network address of the ITSI shall be used as preferred network address.

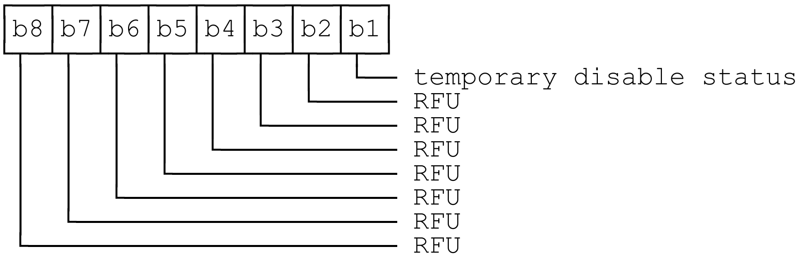
10.3.3 EF$_{\text{ITSIDIS}}$ (ITSI Disabled)

This EF indicates if the ITSI is temporarily disabled.

<table>
<tr><td colspan="2">Identifier: '6F03 '</td><td colspan="2">Structure: transparent</td><td>Mandatory</td></tr>
<tr><td colspan="3">File size: 1 byte</td><td colspan="2">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ CHV1
UPDATE AUTI
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td colspan="2">Status</td><td>M</td><td>1</td></tr>
</table>

- Status:
  Contents: The status bit indicates the temporary disable status of ITSI: 0 not temporarily disabled, 1 temporarily disabled.

Coding: Currently only bit b1 is used. The remaining bits are RFU.



10.3.4 EF$_{UNAME}$ (Username)

This EF contains the alphanumeric name corresponding to the ITSI.

Identifier: '6F04 '	Structure: transparent		Optional
File size: 20 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1-20	Name	M	20

- Name:

  Contents: The common name of the card holder to be displayed.

  Coding: According to the default 8-bit alphabet ISO 8859-1 [22]. Unused bytes shall be set as 'FF'.

10.3.5 EF$_{SCT}$ (Subscriber Class Table)

This EF records the subscriber class membership of the ITSI subscription. The subscriber class membership shall be defined at subscription. The subscriber class element is used to subdivide the MS population in up to 16 classes.

The ITSI subscriber class may only be changed via the MMI by an authorized administrator or via the SwMI by the Network Operator or authorized system manager.

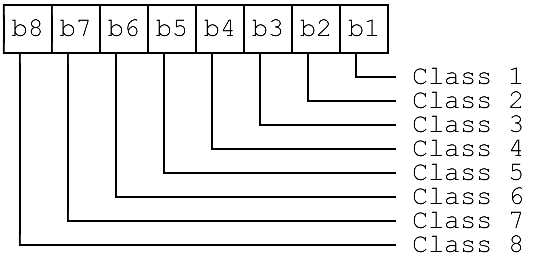
Identifier: '6F05 '	Structure: transparent		Mandatory
File size: 4 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	Classes from 1 to 8	M	1
2	Classes from 9 to 16	M	1
3-4	Energy saving information	O	2

- Classes from 1 to 8:

  Contents: Indicates the class membership for classes from 1 to 8.

  Coding: Bit value 1 means that user is a member, value 0 that user is not a member.

Byte 1:

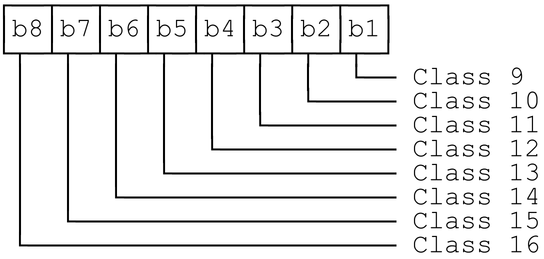


- Classes from 9 to 16:

  Contents: Indicates the class membership for classes from 9 to 16.

  Coding: Bit value 1 means that user is a member, value 0 that user is not a member.

Byte 2:



- Energy Saving Information:

  Contents: Indicates which energy saving scheme (if any) is in operation and the starting point of the energy economy mode.

  Coding: As per ETS 300 392-2 [11], subclause 16.10.10 (14 bits) with b8 and b7 of first byte RFU.

10.3.6 EF_{PHASE} (Phase identification)

This EF contains information concerning the phase of the SIM.

<table>
<tr><td colspan="2">Identifier: '6F06 '</td><td colspan="2">Structure: transparent</td><td>Mandatory</td></tr>
<tr><td colspan="3">File size: 1 byte</td><td colspan="2">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ ALW
UPDATE ADM
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td colspan="2">SIM Phase</td><td>M</td><td>1 byte</td></tr>
</table>

- SIM Phase:

  Coding:
  Phase 1: '00', all other values reserved.

10.3.7 EF$_{CCK}$ (Common Cipher Key)

This EF shall contain 2 records.

Identifier: '6F07 '	Structure: key	Optional
Record size: 12 bytes	Update activity: high	

Access Conditions:

READ	CHV1 (see note 1)
UPDATE	NEV (see note 2)
INVALIDATE	NEV
REHABILITATE	NEV

Bytes	Description	M/O	Length
1-2	CCK-id	M	2
3-12	Common cipher key CCK	M	10

NOTE 1: Read access to this file is only possible by use of Read Key command.

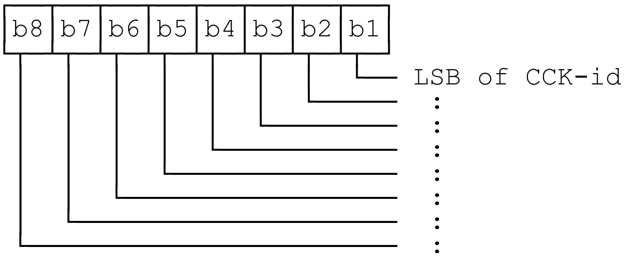
NOTE 2: This EF is updated using the TA32 algorithm on the SIM (i.e. not over the SIM-ME interface).

- CCK-id:

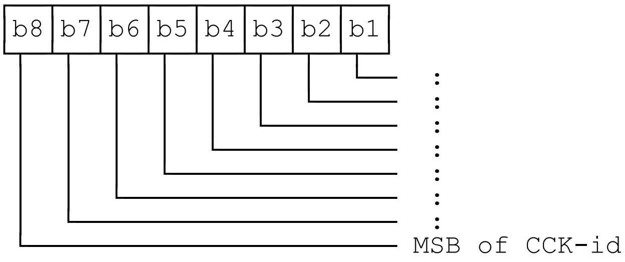
Contents: Common cipher key identity.

Coding:

Byte 1:



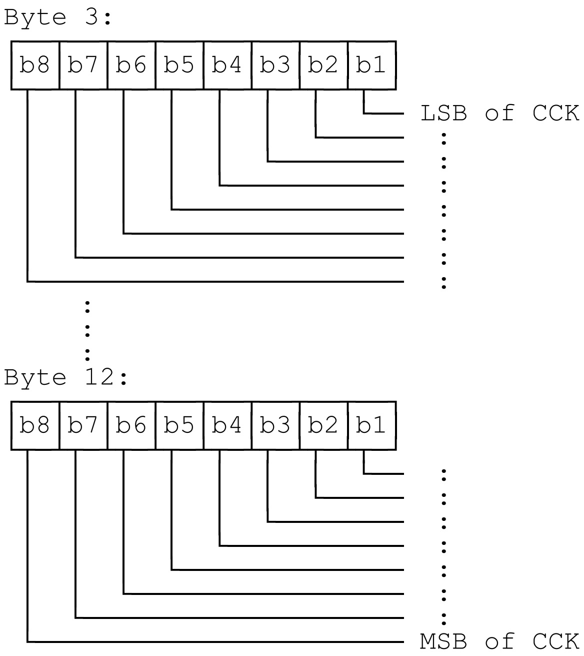
Byte 2:



- Common Cipher Key (CCK):

  Contents: CCK.

  Coding: CCK is coded in 10 bytes according to the following diagram:



10.3.8 EF$_{CCKLOC}$ (CCK location areas)

This EF defines the location area(s) the CCK is valid. If no location areas are defined the CCK is valid in the whole system.

Identifier: '6F08 '	Structure: transparent	Optional
File size: 31 bytes	Update activity: high	

Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Number of location areas	M	1
2-31	Location area	O	30

- Number of location areas:

  Contents: indicates the number location area elements there are to follow in 'Location area'.

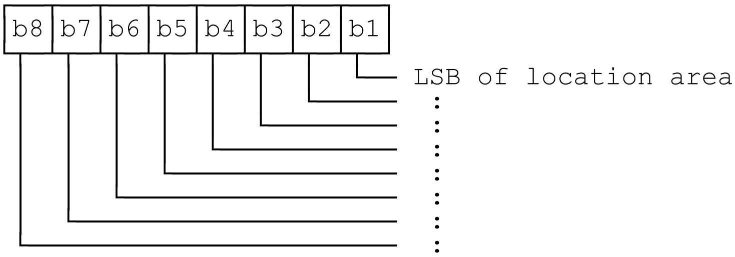
  Coding: binary coded from 0 to 15. If value is 0, the CCK is valid system wide (see also in ETS 300 392-7 [12], subclause 4.4.8.17).

- Location area:

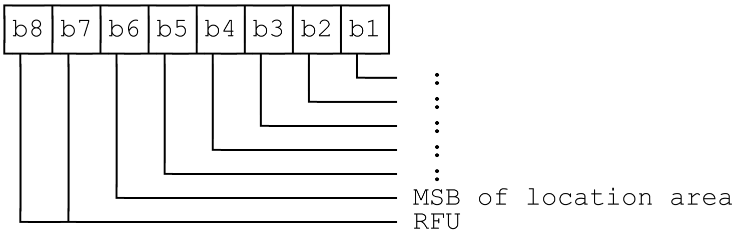
  Contents: a list of location areas where CCKs are valid.

  Coding: Each element is coded in 2 bytes, 14 bits. The first element (bytes 2 and 3) is shown below. See also ETS 300 392-7 [12], subclause 4.4.8.12.

Byte 2:



Byte 3:



10.3.9 EF$_{SCK}$ (Static Cipher Keys)

This EF shall contain up to 32 records.

Identifier: '6F09 '	Structure: key	Optional
Record length: 12 bytes	Update activity: high	
Access Conditions: READ CHV1 (see note 1) UPDATE NEV (see note 2) INVALIDATE NEV REHABILITATE NEV		

Bytes	Description	M/O	Length
1-2	Static Cipher Key Version Number	M	2
3-12	Static Cipher Key	M	10

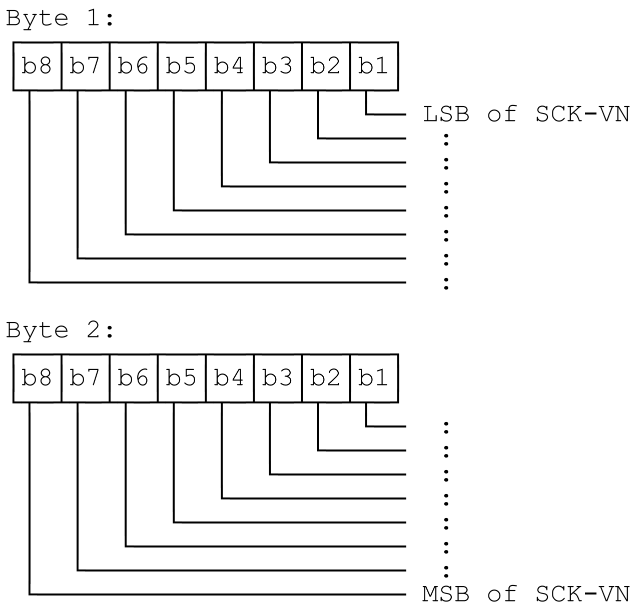
NOTE 1: Read access to this file is only possible by use of Read Key command.

NOTE 2: This EF is updated using the TA41/52 algorithms on the SIM (i.e. not over the SIM-ME interface).

- Static Cipher Key Version Number:

  Contents: The Static Cipher Key Version Number.

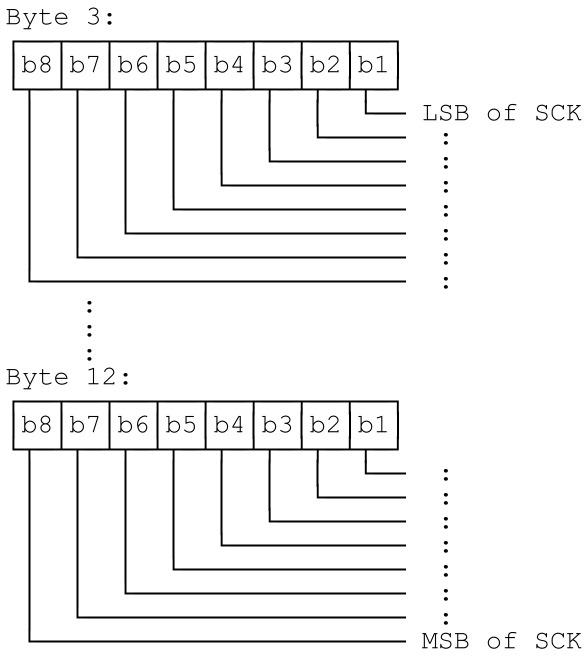
  Coding: The Static Cipher Key Version Number is coded according to the following diagram:



- Static Cipher Key:

  Contents: The Static Cipher Key.

  Coding: The Static Cipher Key is coded in 10 bytes according to the following diagram:



10.3.10 EF_{GSSIS} (Static GSSIs)

This EF contains the pre-programmed (by the operator or organization) group identities.

NOTE 1: Suggested number of static groups is between 1 and 10.

NOTE 2: Static GSSIs can not be updated after the administrative phase.

<table>
<tr><td colspan="2">Identifier: '6F0A'</td><td colspan="2">Structure: linear fixed</td><td>Mandatory</td></tr>
<tr><td colspan="3">Record length: X + 4 bytes</td><td colspan="2">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ CHV1
UPDATE ADM
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1 to X</td><td colspan="2">Group name</td><td>M</td><td>X</td></tr>
<tr><td>X + 1</td><td colspan="2">Network address record number</td><td>M</td><td>1</td></tr>
<tr><td>X + 2-X + 4</td><td colspan="2">Group Identity (GSSI)</td><td>M</td><td>3</td></tr>
</table>

- Group name:

  Contents: Alphanumeric names for the static groups stored on the SIM.

  Coding: The value of X may range from zero to 251.

- Network address record number:

  Contents: Record number of the corresponding network address. Network addresses are stored in EF_{NWT}.

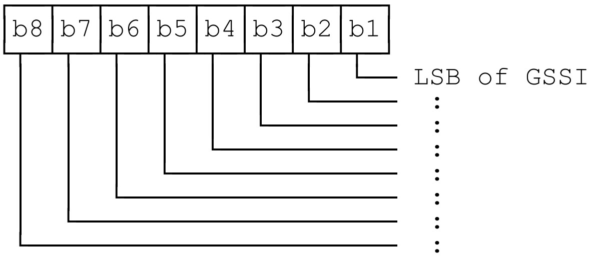
  Coding: binary. Free records are indicated by NULL value ('00').

- Group Identity (GSSI):

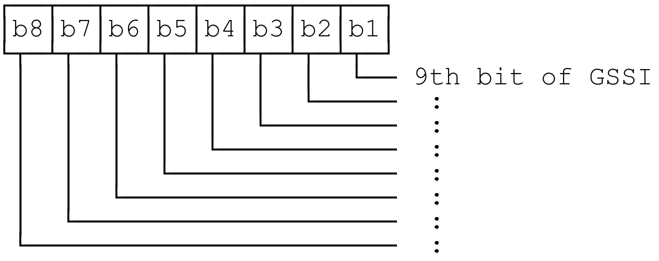
  Contents: The short subscriber identity for the group.

  Coding: Length of the GSSI is 24 bits.

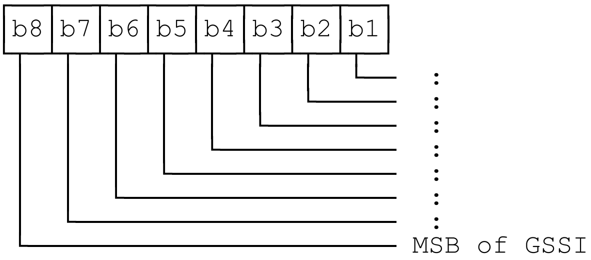
Byte X+3:



Byte X+4:



Byte X+5:



10.3.11 EF$_{GRDS}$ (Group related data for static GSSIs)

This EF contains information related to each static GSSI. There shall be a 1:1 relationship between each record in EF$_{GRDS}$ and the corresponding record in EF$_{GSSIS}$.

Identifier: '6F0B '	Structure: linear fixed		Mandatory
Record size: 2 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	GCK record number	M	1
2	Group related data	M	1

- GCK record number:

  Contents: Record number of the corresponding GCK in the EF$_{GCK}$-file.

  Coding: binary. If there is no GCK defined for this group, GCK record number shall be NULL value ('00').

- Group related data:

  Contents:

  Class of usage (3 bits). Shall indicate the importance of the group for the user and define the participation rules for the groups defined with Class of usage. (ETS 300 392-2 [11], subclause 16.10.6 and ETS 300 392-12-22 [20], subclause 5.6.14).

  Activation data (2 bits). Shall indicate the activation status of the group identity. The activation data contents is defined by the home network. The activation data definition is done either by using attachment/detachment of group identities procedure (ETS 300 392-2 [11], subclause 16.8) and/or by using Supplementary Service Dynamic Group Number Assignment (SS-DGNA) assignment (ETS 300 392-12-22 [20], table 23). The mapping between activation data, Group identity attachment lifetime and Group identity detachment downlink is defined in ETS 300 392-2 [11], subclauses 16.10.16 and 16.10.20.

Table 6: Activation data contents

Information element	Length	Value	Remark
Activation Data	2	00	Permanently inactive, the group cannot be activated by the MS user
		01	Temporarily inactive, the group attachment of the group identity may be triggered by the MS user
		10	Permanently active, the group is allowed for the user always, no attachment of the group identity is required/allowed
		11	Reserved

Table 7: Mapping between Group Identity Attachment Lifetime and Activation Data

Group Identity Attachment Lifetime	Activation Data
Session based, attachment not allowed	Permanently inactive, the group cannot be activated by the MS user
Permanent, attachment for next session needed	Temporarily inactive, the group attachment of the group identity may be triggered by the MS user
Permanent, attachment not needed	Permanently active, the group is allowed for the user always, no attachment of the group identity is required/allowed

Table 8: Mapping between Group identity detachment downlink and Activation Data

Group identity detachment downlink	Activation Data

Unknown group identity;
Permanent detachment;
Temporary detachment if Group Identity Attachment Lifetime was set to Session based, attachment not allowed;
Session based if Group Identity Attachment Lifetime was set to Session based, attachment not allowed.
	Permanently inactive, the group cannot be activated by the MS user.

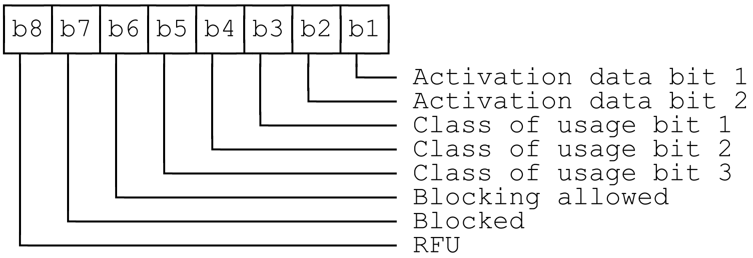
Temporary detachment if Group Identity Attachment Lifetime was set Permanent, attachment for next session required; Session based detachment if Group Identity Attachment Lifetime was set to Permanent, attachment for next session required.	Temporarily inactive, the group attachment of the group identity may be triggered by the MS user.
Temporary detachment if Group Identity Attachment Lifetime was set to Permanent, attachment not needed; Session based detachment if Group Identity Attachment Lifetime was set to Permanent, attachment not needed.	Permanently active, the group is allowed for the user always, no attachment of the group identity is required/allowed.
NOTE: If Activation data for a group is set to permanently active and group identity detachment downlink indicates permanent detachment, then the activation data is changed to Permanently inactive. Other detachments need an attachment in order to make the detachment take place and thus the mapping is valid for other cases. It is not possible to change de-activated group's activation data by using Group identity detachment downlink element.	

Blocking allowed (1 bit): Indicates if the user is allowed to block this group from the list of groups to be listened.

Blocked (1 bit): Indicates if the user has blocked this group from the list of groups to be listened.

Coding: All bits are coded into one byte:

Byte 2:



10.3.12 EF$_{GSSID}$ (Dynamic GSSIs)

This EF contains the dynamic group identities.

Identifier: '6F0C '	Structure: linear fixed	Mandatory
Record length: X + 4 bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	AUTI
INVALIDATE	AUTI
REHABILITATE	NEV

Bytes	Description	M/O	Length
1 to X	Group name	M	X
X + 1	Network address record number	M	1
X + 2-X + 4	Group Identity (GSSI)	M	3

- See EF$_{GSSIS}$ (Static GSSIs) for contents and coding.

10.3.13 EF$_{GRDD}$ (Group related data for dynamic GSSIs)

This EF contains information related to each dynamic GSSI. There shall be a 1:1 relationship between each record in EF$_{GRDD}$ and the corresponding record in EF$_{GSSID}$.

Identifier: '6F0D'	Structure: linear fixed	Mandatory
Record size: 2 bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	GCK record number	M	1
2	Group related data	M	1

- See EF$_{GRDS}$ for contents and coding.

10.3.14 EF$_{GCK}$ (Group Cipher Keys)

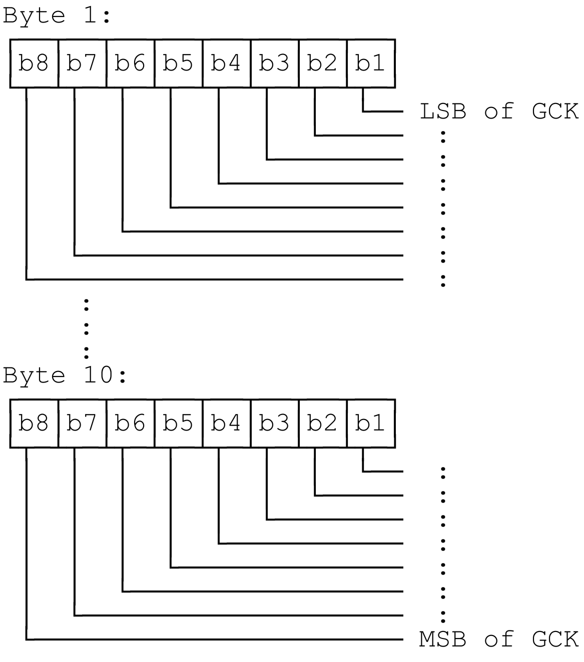
This EF contains the group cipher keys associated with the group identities. There shall be a 1:1 relationship between each MGCK in EF$_{MGCK}$ and the corresponding record of GCK in EF$_{GCK}$.

Identifier: '6F0E '	Structure: linear fixed		Optional
Record length: 10 bytes		Update activity: high	
Access Conditions: READ NEV (see note 1) UPDATE NEV (see note 2) INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1-10	GCK	M	10
NOTE 1: There is no access to this EF over the SIM-ME interface. NOTE 2: GCK is updated on the SIM by use of the TA82 algorithm. NOTE 3: A record is free if no (static or dynamic) GSSI points to it.			

- GCK:

Contents: The Group Cipher Keys.

Coding: The key is stored in 10 bytes according to the following diagram:



10.3.15 EF$_{MGCK}$ (Modified Group Cipher Keys)

This EF contains the modified group cipher keys associated with the group identities. There shall be a 1:1 relationship between each MGCK in EF$_{MGCK}$ and the corresponding record of GCK in EF$_{GCK}$.

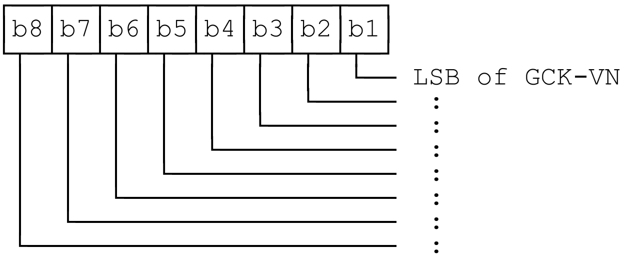
<table>
<tr><td colspan="2">Identifier: '6F0F '</td><td colspan="2">Structure: key</td><td>Optional</td></tr>
<tr><td colspan="3">Record length: 12 bytes</td><td colspan="2">Update activity: high</td></tr>
<tr><td colspan="5">Access Conditions:
READ CHV1 (see note 1)
UPDATE NEV (see note 2)
INVALIDATE NEV
REHABILITATE NEV</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1-2</td><td colspan="2">GCK-VN</td><td>M</td><td>2</td></tr>
<tr><td>3-12</td><td colspan="2">MGCK</td><td>M</td><td>10</td></tr>
<tr><td colspan="5">NOTE 1: Read access to this file is only possible by use of Read Key command.
NOTE 2: Updating of this EF is performed by the TA71 algorithm on the SIM (i.e. not over SIM-ME interface).
NOTE 3: A record is free if no (static or dynamic) GSSI points to it.</td></tr>
</table>

- GCK-VN:

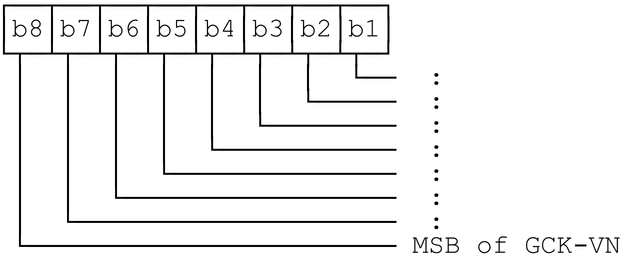
  Contents: Group cipher key Version Number

  Coding:

  Byte 1:



  Byte 2:



- MGCK:

  Contents: The Modified Group Cipher Key.

  Coding: As for GCK above.

10.3.16 EF$_{GINFO}$ (User's group information)

This EF contains the user's last active group, user's default group and information about using these group addresses.

Identifier: '6F10 '	Structure: transparent	Mandatory
File size: 9 bytes	Update activity: high	

Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

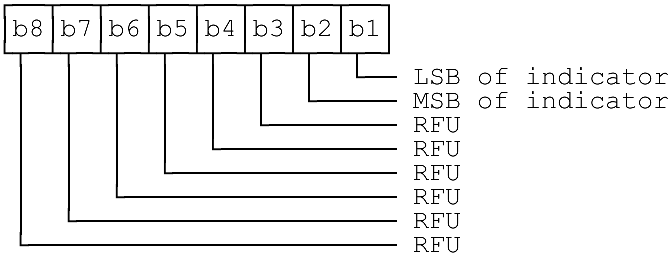
Bytes	Description	M/O	Length
1	Usage information	M	1
2	Network address record number of last active group	M	1
3-5	GSSI of the last active group	M	3
6	Network address record number of user's home group	M	1
7-9	GSSI of the user's home group	M	3

- Usage information:

Contents: Two bits indicate the use of addresses. 00 indicates that neither of the addresses are used, 01 indicates that the last group address is to be used and 10 indicates that the home group address is to be used. 11 is reserved for future use.

Coding:

Byte 1:



- Network address record number of last active group:

Contents: Record number of the corresponding network address in EF$_{NWT}$.

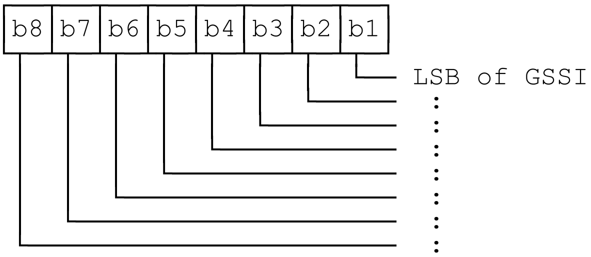
Coding: Binary. NULL value ('00') indicates that no GSSI is stored.

- GSSI of the last active group:

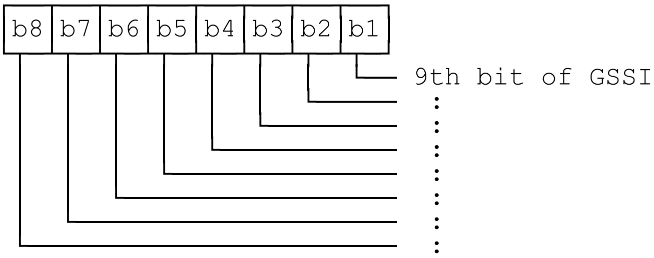
  Contents: The short subscriber identity for the group that was last active.

  Coding: Length of the GSSI is 24 bits.

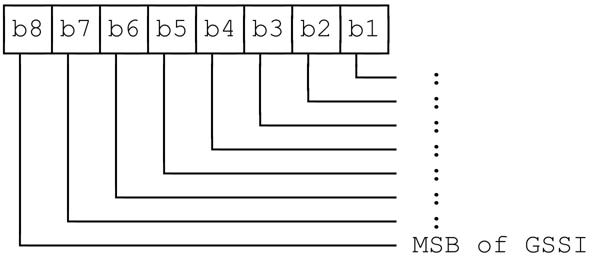
  Byte 3:



  Byte 4:



  Byte 5:



- Network address record number of user's home group:

  Contents: Record number of the corresponding network address in EF_{NWT}.

  Coding: binary. NULL value ('00') indicates that no GSSI is stored.

- GSSI of the user's home group:

  Contents: The short subscriber identity for the user's home group.

  Coding: Length of the GSSI is 24 bits. Coded as GSSI of the last active group above, except with bytes 7-9.

NOTE 1: This record is updated at the beginning of a group call.

10.3.17 EF_{SEC} (Security settings)

This EF indicates the values for the security settings.

Identifier: '6F11'	Structure: transparent		Mandatory
File size: 1 byte		Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	Security settings	M	1

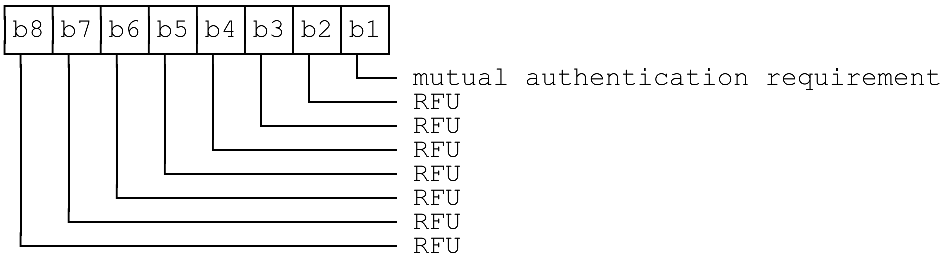
Security settings:

Contents: bit 1 indicates whether the SIM requests a mutual authentication when it is authenticated by the SwMI.

Coding:

bit = 1: mutual authentication required;
bit = 0: mutual authentication not required.

Byte 1:



10.3.18 EF_{FORBID} (Forbidden networks)

This EF contains the coding for Forbidden networks. It is read by the ME as part of the SIM initialization procedure and indicates networks which the MS shall not automatically attempt to access.

A network address is written to the EF if a network rejects a Location Update with the following causes "Illegal MS" and "Migration not supported" as in ETS 300 392-2 [11], subclause 16.10.42. The ME shall update the list by using the "next" mode of the update record command.

NOTE 1: By using the "next" mode in update operations the oldest record will be overwritten in the case the file is full.

NOTE 2: This EF should have at least as many records as is the expected amount of forbidden networks. Otherwise the ME may find the same forbidden networks in the beginning of every TETRA session and rewrite them to the list.

Identifier: '6F12 '	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

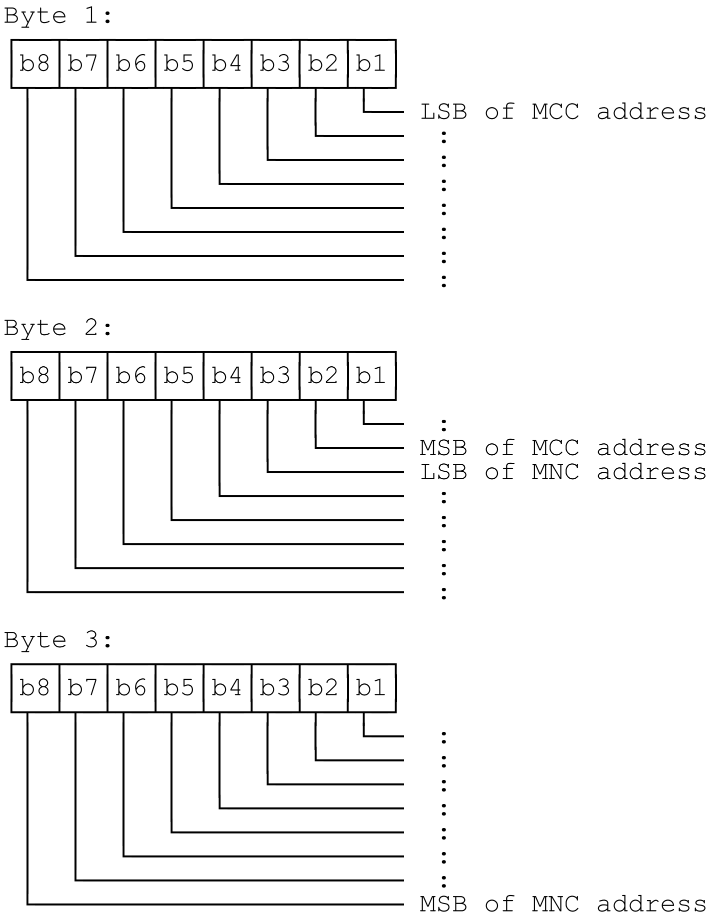
READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1-3	Network address	M	3

- Network address:

Contents: The address consists of MCC and MNC addresses, 10 and 14 bits respectively.

Coding: according to the following diagram. Empty records shall be set to 'FF'.



10.3.19 EF_{PREF} (Preferred networks)

This EF contains a list of preferred network addresses. The networks are listed in the order of preference. The first record corresponds to the highest preference.

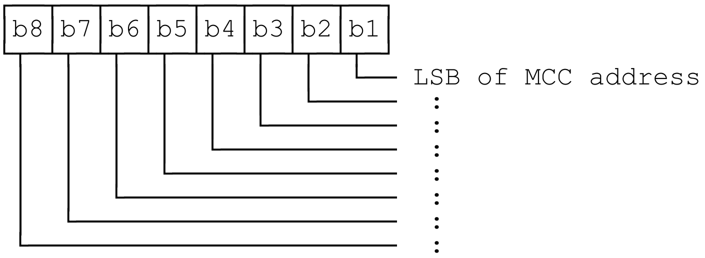
<table>
<tr><td colspan="2">Identifier: '6F13 '</td><td colspan="2">Structure: linear fixed</td><td>Optional</td></tr>
<tr><td colspan="3">Record length: 3 bytes</td><td colspan="2">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ CHV1
UPDATE CHV1/ADM (see note)
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1-3</td><td colspan="2">Network address</td><td>M</td><td>3</td></tr>
<tr><td colspan="5">NOTE: Card issuer will choose between CHV1 or ADM protection.</td></tr>
</table>

- Network address:

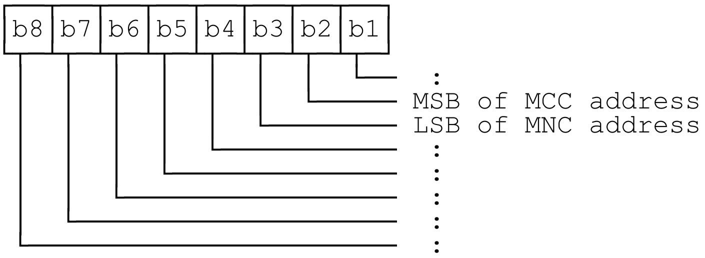
  Contents: The address consists of MCC and MNC addresses, 10 and 14 bits respectively.

  Coding: according to the following diagram. Empty records shall be set to 'FF'.

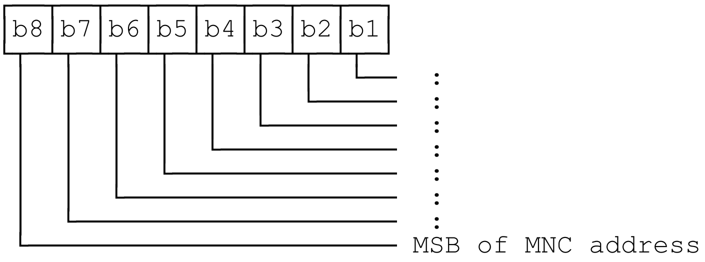
Byte 1:



Byte 2:



Byte 3:



10.3.20 EF$_{SPN}$ (Service Provider Name)

This EF contains the service provider name and appropriate requirements for the display by the ME.

Identifier: '6F14 '	Structure: transparent	Optional
File size: 17 bytes	Update activity: low	

Access Conditions:

READ	ALW
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Display Condition	M	1
2-17	Service Provider Name	M	16

- Display condition:

Contents: Display condition for the service provider name in respect to the network.

Coding:

Byte1:

Bit	b1;
0	:display of registered network not required;
1	:display of registered network required;

Bits b2 to b8 are RFU.

- Service provider name:

Contents: Service provider string to be displayed.

Coding: The string shall use the default 8-bit alphabet ISO 8859-1 [22]. The string shall be left justified. Unused bytes shall be set to 'FF'.

10.3.21 EF$_{LOCI}$ (Location information)

This EF contains the following information:

- Alias Short Subscriber Identity (ASSI);

- Network address record number for ASSI;

- Location Area (LA);

- Network address record number for Location Area.

Identifier: '6F15 '	Structure: transparent	Mandatory
File size: 7 bytes	Update activity: high	

Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 - 3	ASSI	M	3
4	ASSI Network address record number	M	1
5-6	Location Area	O	2
7	Location Area Network address record number	O	1

- ASSI:

  Contents: Alias Short Subscriber Identity.

  Coding: Short Subscriber Identity (SSI) according to EF_{ITSI}.

- ASSI Network address record number:

  Contents: Network address record number for ASSI. Refers to addresses in EF_{NWT}.

  Coding: binary. NULL value ('00') indicates that no ASSI stored.

- Location Area:

  Contents: Location Area of last successful registration.

  Coding: As per ETS 300 392-2 [11], subclause 16.10.34 (14 bits) with b8 and b7 of first byte RFU.

- LA Network address record number:

  Contents: Network address record number for LA. Refers to addresses in EF_{NWT}.

  Coding: binary. NULL value ('00') indicates that no Location Area stored.

10.3.22 EF_{DNWRK} (Broadcast network information)

This EF contains information concerning the D-NWRK-BROADCAST according to ETS 300 392-2 [11], subclause 18.4.1.4.1. It shall contain 7 records (see ETS 300 392-2 [11], subclause 18.5.19).

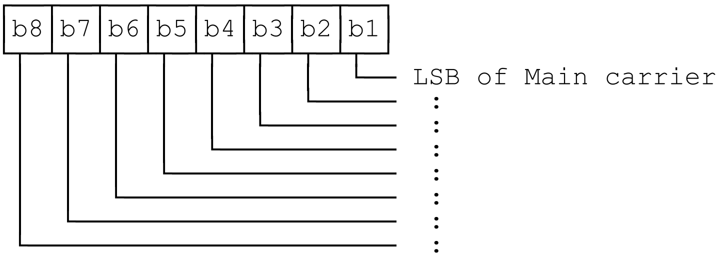
Storage of neighbour cell information may reduce the extent of a MS's search for MCCH carriers when selecting a cell.

<table>
<tr><td colspan="2">Identifier: '6F16 '</td><td colspan="2">Structure: linear fixed</td><td>Mandatory</td></tr>
<tr><td colspan="3">Record size: 3 bytes</td><td colspan="2">Update activity: high</td></tr>
<tr><td colspan="5">Access Conditions:
READ CHV1
UPDATE CHV1
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1 - 3</td><td colspan="2">MCCH information</td><td>M</td><td>3 bytes</td></tr>
</table>

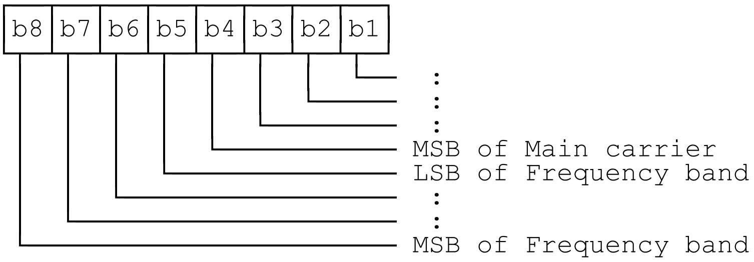
- MCCH information:

Coding: The information is coded as defined in ETS 300 392-2 [11], subclauses 18.5.10 and 18.5.11. Free records are indicated in bit 7 of byte 3.

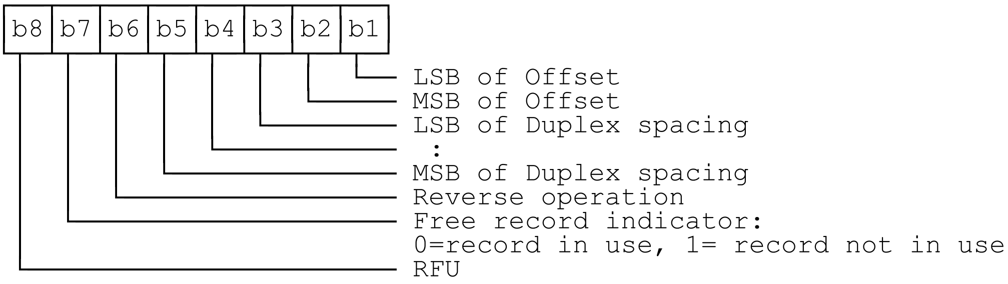
Byte 1:



Byte 2:



Byte 3:



10.3.23 EF$_{NWT}$ (Network table)

This EF contains the network part of the TETRA addresses. These addresses are used and updated by several EFs (EF$_{GSSIS}$, EF$_{GSSID}$, EF$_{GINFO}$, EF$_{GWT}$, EF$_{ADNTETRA}$, EF$_{SDNTETRA}$, EF$_{FDNTETRA}$, EF$_{LDNTETRA}$). The files reference to this file using the record number of network addresses on this file.

Identifier: '6F17 '	Structure: linear fixed	Mandatory
Record size: 5 bytes	Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

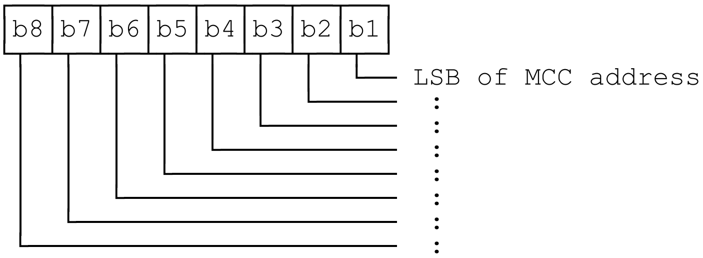
Bytes	Description	M/O	Length
1-3	Network address (MCC and MNC)	M	3
4-5	Record pointer counter	M	2

- Network address:

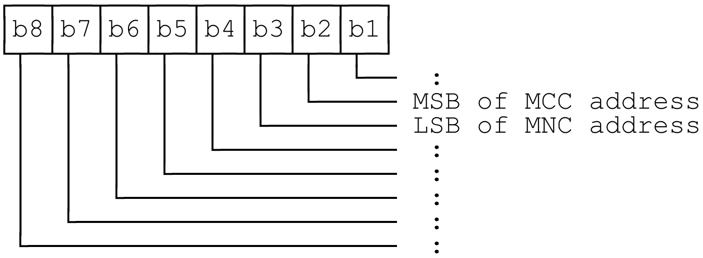
Contents: The address consists of MCC and MNC addresses, 10 and 14 bits respectively. The user's home address (from ITSI) is stored as the first record (0) of the file.

Coding:

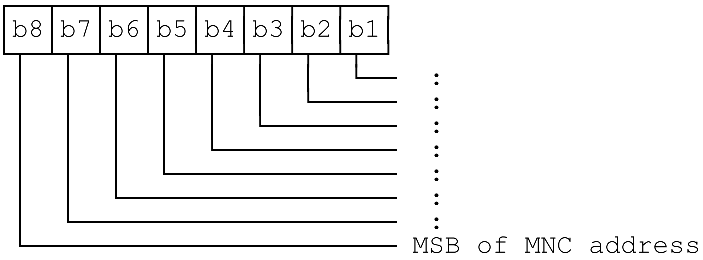
Byte 1:



Byte 2:



Byte 3:



- Record pointer counter:

  Contents: The records in this file can be referenced from several other files. This counter is incremented each time a new reference to a record is created. Also when the reference is deleted, this counter should be decremented.

  Coding: Binary. NULL value ('00') indicates a free record.

NOTE: This file is updated by the ME when updating EFs which reference this file.

10.3.24 EF_{GWT} (Gateway table)

This EF contains the names and addresses for Private Automatic Branch Exchange (PABX) and Public Switched Telephone Network (PSTN) gateways in a TETRA network. This file is referenced by EF_{ADN}, EF_{FDN}, EF_{LDN} and EF_{SDN}. The files reference to this file using the record number of gateway names and addresses on this file.

Identifier: '6F18 '	Structure: linear fixed		Optional
Record size: 14 bytes		Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1-8	Name	M	8
9	Network address record number	M	1
10-12	SSI of the Gateway	M	3
13-14	Record pointer counter	M	2

- Name:

  Contents: The alphanumeric name for the corresponding Gateway.

  Coding: The string shall use the default 8-bit alphabet. The string shall be left justified. Unused bytes shall be set to 'FF'.

- Network address record number:

  Contents: Record number of the corresponding network address in EF_{NWT}.

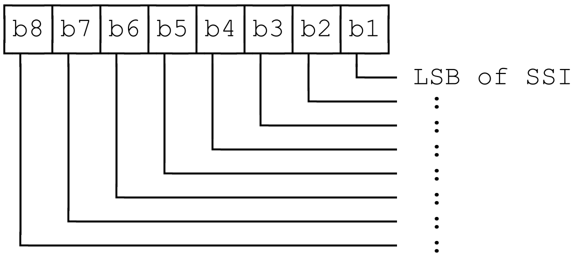
  Coding: binary

- SSI of the Gateway:

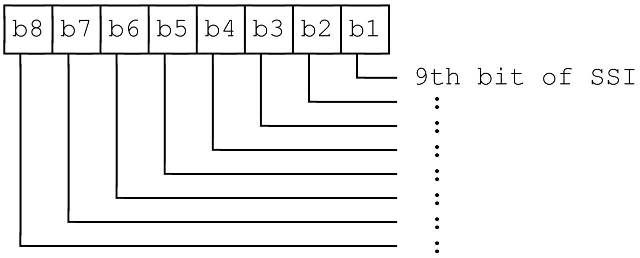
Contents: The short subscriber identity of the PABX-gateway used.

Coding: Length of the SSI is 24 bits.

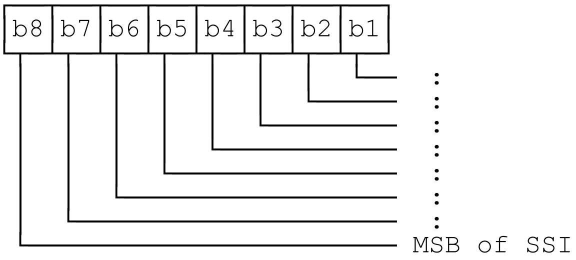
Byte 10:



Byte 11:



Byte 12:



- Record pointer counter:

Contents: The records in this file can be referenced from several other files. This counter is incremented each time a new reference to a record is created. Also when the reference is deleted, this counter should be decremented.

Coding: binary. NULL value ('00') indicates a free record.

NOTE: This file is updated by the ME when updating EFs which reference this file.

10.3.25 EF$_{CMT}$ (Call Modifier Table)

This EF indicates the values for the call modifiers required by the ME on a per call basis. These are intended to provide a sensible set of call modifiers for use where the user does not, or can not, enter them during call set-up. It is proposed that there are different sets of modifiers for different types of calls and that these sets are selected by the ME according to the call type. Alternatively, the ME may allow the user to select a set of call modifiers via the MMI. The alphanumeric field is intended to assist the user in selecting a proper call modifier.

To allow default values to be defined on subscription for each of the call types, the first 12 entries in the table are designated for particular call types in fixed positions. The user may add more call modifiers after the first 12 entries.

Each record in phonebooks may refer to a call modifier in this EF.

<table>
<tr><td colspan="2">Identifier: '6F19 '</td><td colspan="2">Structure: linear fixed</td><td>Optional</td></tr>
<tr><td colspan="3">Record length: X + 4 bytes</td><td colspan="2">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ CHV1
UPDATE CHV1/CHV2 (see note)
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1 to X</td><td colspan="2">Name</td><td>M</td><td>X</td></tr>
<tr><td>X + 1 to X + 4</td><td colspan="2">Call modifiers</td><td>M</td><td>4</td></tr>
<tr><td colspan="5">NOTE: Card issuer will choose between CHV1 or CHV2 protection.</td></tr>
</table>

- Name:

Contents: An alphanumeric identifier for the set of call modifier values.

Coding: According to the default 8-bit alphabet ISO 8859-1 [22] A free record is indicated by filling this field with 'FF'.

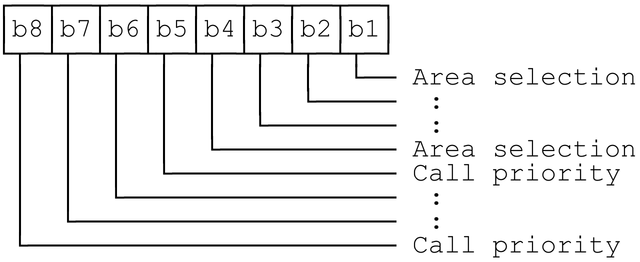
- Call modifiers:

Contents: The file consists of the following pieces of information:

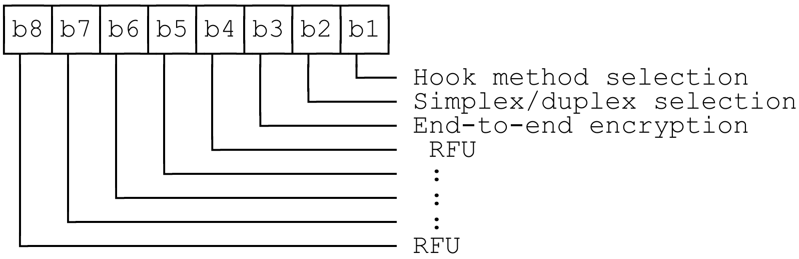
Area selection	4 bits;
Call priority	4 bits;
Hook method selection	1 bit;
Simplex/duplex selection	1 bit;
End-to-end encryption	1 bit;
Basic service information	16 bits.

Coding: All bits are coded into four bytes.

Byte 1:



Byte 2:



Bytes 3 and 4 are coded as "basic service information" in ETS 300 392-2 [11], subclause 14.8.2.

- Fixed call modifier sets:

  the default call modifier sets are placed in EF_{CMT} in a standard order to allow selection of the set by call type.

Record in EF_{CMT}	**Call Type**	**Call features**
Record 1	Voice call	Intra-TETRA, individual call
Record 2	Voice call	Intra-TETRA, group call
Record 3	Voice call	Intra-TETRA, acknowledged
Record 4	Voice call	Intra-TETRA, broadcast call
Record 5	Voice call	PABX call
Record 6	Voice call	PSTN call
Record 7	Circuit mode data call	Intra-TETRA, individual call
Record 8	Circuit mode data call	Intra-TETRA, group call
Record 9	Circuit mode data call	Intra-TETRA, acknowledged
Record 10	Circuit mode data call	Intra-TETRA, broadcast call
Record 11	Circuit mode data call	PABX call
Record 12	Circuit mode data call	PSTN call

NOTE: This EF references the ETS 300 392-2 [11], subclause 14.7.2.10.

10.3.26 EF_{ADN} (Abbreviated Dialling Number)

This EF contains ADN. In addition it contains record numbers of associated gateway, call modifier and extension records.

NOTE: When calling to phone numbers contained in this EF from within a TETRA network, the PSTN/PABX gateway address is added in front of the phone number. This implementation requires that there is one universally acknowledged TETRA address for PSTN gateways in all different networks.

Identifier: '6F1A '	Structure: linear fixed		Optional
Record length: X+12 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE CHV2 REHABILITATE CHV2			
Bytes	Description	M/O	Length
1 to X	Name	O	X
X+1	Length of PSTN or PABX number contents	M	1
X+2 to X+9	PSTN or PABX number	M	8
X+10	Gateway address record number	M	1
X+11	Call modifier record number	M	1
X+12	Extension1 record number	M	1

- Name:

  Contents: The alphanumeric name the user has assigned for corresponding phone number.

  Coding: According to the default 8-bit alphabet ISO 8859-1 [22].

- Length of PSTN or PABX number contents:

  Contents: this field gives the number of digits of the following "PSTN or PABX number" -field containing an actual BCD number. This means that the maximum value is 16, even when the actual ADN length is greater than 16 digits. When an ADN requires more than 16 digits it is indicated by the Extension1 record number being unequal to 'FF'. The remainder is stored in the EF_{EXT1} with the remaining length of the overflow data being coded in the appropriate overflow record itself (see subclause 10.3.27).

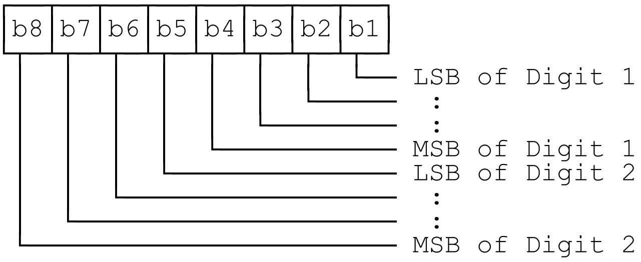
  Coding: binary. NULL ('00') value indicates a free record.

- PSTN or PABX number:

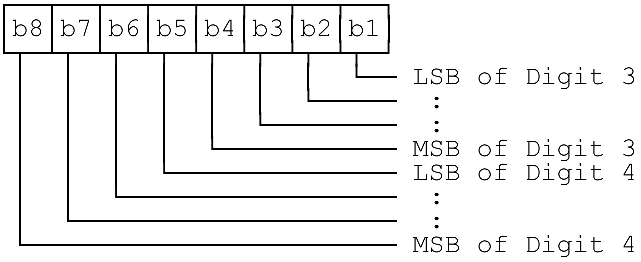
  Contents: up to 16 digits of the telephone number.

  Coding: according to ETS 300 392-2 [11], subclause 14.8.20. If the telephone number is longer than 16 digits, the first 16 digits are stored in this data item and the overflow data is stored in an associated record in the EF_{EXT1}. The record is identified by the Extension1 record number. If ADN requires less than 16 digits, excess nibbles at the end of the data item shall be ignored.

Byte X+2



Byte X+3:



etc.

- Gateway address record number:

  Contents: This byte identifies the number of a record in the EF_{GTW} containing an associated gateway address. The use of this byte is optional. If it is not used it shall be set to 'FF'.

  Coding: binary.

- Call modifier record number:

  Contents: This byte identifies the number of a record in the EF_{CMT} containing an associated call modifier information. The use of this byte is optional. If it is not used it shall be set to 'FF'.

  Coding: binary.

- Extension1 record number:

  Contents: This byte identifies the number of a record in the EF_{EXT1} containing an associated ADN overflow. The use of this byte is optional. If it is not used it shall be set to 'FF'.

  Coding: binary.

10.3.27 EF_{EXT1} (Extension1)

This EF contains extension data of an ADN or Last Number Dialled (LND). Extension data is caused by an ADN or LND which is greater than the 16 digit capacity of the ADN or LND EF. The remainder is stored in this EF as a record, which is identified by a specified identification byte inside the ADN or LND EF.

Identifier: '6F1B '	Structure: linear fixed		Optional
Record length: 13 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	Length of extension data	M	1
2 to 12	Extension data	M	11
13	Next record number	M	1

- Length of extension data:

  Contents: This field gives the number of digits of the following "Extension data" -field containing an actual BCD number.

  Coding: Binary. NULL ('00') value indicates a free record.

- Extension data:

  Contents: Up to 22 digits of the telephone number.

  Coding: According to ETS 300 392-2 [11], subclause 14.8.20.

- Next record number:

  Contents: Record number of the next extension record to enable storage of information longer than 11 bytes.

  Coding: Record number of next record. 'FF' identifies the end of the chain.

10.3.28 EF$_{ADNTETRA}$ (Abbreviated dialling numbers for TETRA network)

EF contains the phone numbers that are used when calling to a TETRA phone. The access strings for Supplementary services are stored in the same file.

Identifier: '6F1C'	Structure: linear fixed	Optional
Record length: X + 6 bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	CHV2
REHABILITATE	CHV2

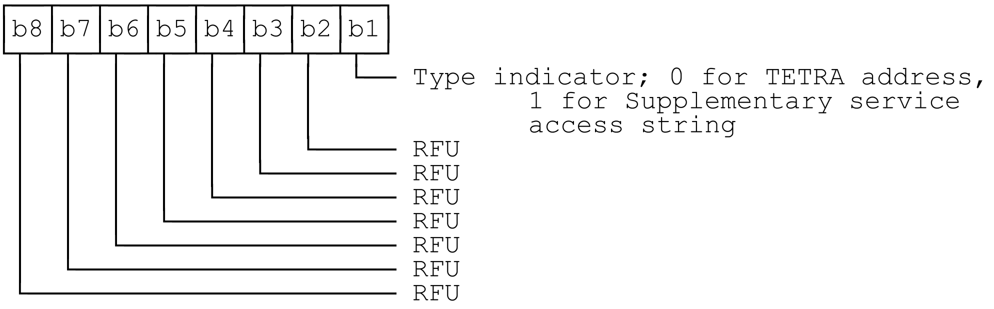
Bytes	Description	M/O	Length
1	Type	M	1
2 to X	Name	M	X
X + 1	Network address record number	M	1
X + 2 to X + 4	SSI of TETRA address or Supplementary service access string	M	3
X + 5	Call modifier record number	M	1
X + 6	Extension A record number	M	1

- Type:

Contents: One byte indicator to identify the entry type in SSI of TETRA address or Supplementary service access string -field.

Coding:

Byte 1:


b8
b7
b6
b5
b4
b3
b2
b1
Type indicator; 0 for TETRA address,
1 for Supplementary service
access string
RFU
RFU
RFU
RFU
RFU
RFU
RFU

- Name:

Contents: The alphanumeric name the user has assigned for corresponding phone number or Supplementary services access string.

Coding: According to the default 8-bit alphabet ISO 8859-1 [22].

- Network address record number:

Contents: Record number of the corresponding network address. Network addresses are stored in EF$_{NWT}$.

Coding: Binary. NULL ('00') value indicates a free record. When storing the Supplementary service *access* strings to the SSI of TETRA address, this field is set to 'FF'.

- Call modifier record number:

  Contents: This byte identifies the number of a record in the EF_{CMT} containing an associated call modifier information. The use of this byte is optional. If it is not used it shall be set to 'FF'.

  Coding: Binary.

- SSI of TETRA address or Supplementary service access string:

  Contents: The short subscriber identity that is used when calling to a TETRA phone or Supplementary service strings to be stored.

  Coding: Length of the SSI is 24 bits. When the field contains a SSI the field is binary-coded. When storing Supplementary service strings on this field, the digits and characters are BCD-coded according to ETS 300 392-2 [11], subclause 14.8.19.

- Extension A record number:

  Contents: This byte identifies the number of a record in the EF_{EXTA} containing an associated supplementary services access string overflow. The use of this byte is optional. If it is not used, it shall be set to 'FF'.

  Coding: Binary.

10.3.29 EF_{EXTA} (Extension A)

This EF contains the overflow of a Supplementary service access string.

<table>
<tr><td colspan="2">Identifier: '6F1D'</td><td>Structure: linear fixed</td><td colspan="2">Optional</td></tr>
<tr><td colspan="3">Record length: 20 bytes</td><td colspan="2">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ CHV1
UPDATE CHV1
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td colspan="2">Length of extension data</td><td>M</td><td>1</td></tr>
<tr><td>2 to 19</td><td colspan="2">Overflow data</td><td>M</td><td>18</td></tr>
<tr><td>20</td><td colspan="2">Next record number</td><td>M</td><td>1</td></tr>
</table>

- Length of extension data:

  Contents: This field gives the number of digits of the following "Overflow data" -field containing an actual BCD number.

  Coding: Binary. NULL ('00') value indicates a free record.

- Overflow data:

  Contents: Overflow data of a Supplementary services access string.

  Coding: BCD according to ETS 300 392-2 [11], subclause 14.8.19.

- Next record number:

  Contents: record number of the next extension record to enable storage of information longer than 18 bytes.

  Coding: record number of next record. 'FF' identifies the end of the chain.

10.3.30 EF_{FDN} (Fixed dialling numbers)

This EF contains FDN. In addition it contains record numbers of associated gateway, call modifier and extension records.

NOTE 1: When calling to phone numbers contained in this EF from within a TETRA network, the PSTN/PABX gateway address is added in front of the phone number. This implementation requires that there is one universally acknowledged TETRA address for PSTN gateways in all different networks.

NOTE 2: Fixed dialling numbers are used for example in a situation when a supervisor in an organization fixes the numbers on a SIM card so that a worker of the organization may only call to work related numbers.

Identifier: '6F1E'	Structure: linear fixed	Optional
Record length: X + 12 bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	CHV2
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 to X	Name	O	X
X + 1	Length of PSTN or PABX number contents	M	1
X + 2 to X + 9	PSTN or PABX number	M	8
X + 10	Gateway address record number	M	1
X + 11	Call modifier record number	M	1
X + 12	Extension2 record number	M	1

For contents and coding of all data items see the respective data items of the EF_{ADN}, with the exception that extension records are stored in the EF_{EXT2}.

10.3.31 EF$_{EXT2}$ (Extension2)

This EF contains extension data of an FDN (see Extension2 record number in 10229). Extension data is caused by an FDN which is greater than the 16 digit capacity of the EF$_{FDN}$. The remainder is stored in this EF as a record, which is identified by a specified identification byte inside the EF$_{FDN}$.

Identifier: '6F1F'	Structure: linear fixed		Optional
Record length: 13 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV2 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	Length of extension data	M	1
2 to 12	Extension data	M	11
13	Next record number	M	1

For contents and coding see subclause 10.3.27.

10.3.32 EF$_{FDNTETRA}$ (Fixed dialling numbers for TETRA network)

EF contains the Fixed Dialling Numbers (FDN) to be used within TETRA network.

Identifier: '6F20'	Structure: linear fixed		Optional
Record length: X + 6 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV2 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1 to X	Name	M	X
X + 1	Network address record number	M	1
X + 2 to X + 4	SSI of TETRA address	M	3
X + 5	Call modifier record number	M	1
X + 6	Extension B record number	M	1

For contents and coding of all data items see the respective data items of the EF$_{ADNTETRA}$.

10.3.33 EF$_{EXTB}$ (Extension B)

This EF contains the overflow of a Supplementary service access string.

Identifier: '6F21'	Structure: linear fixed		Optional
Record length: 20 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV2 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	Length of extension data	M	1
2 to 19	Overflow data	M	18
20	Next record number	M	1

For contents and coding of all data items see the respective data items of the EF$_{EXTA}$.

10.3.34 EF$_{LND}$ (Last number dialled)

This EF contains the last numbers dialled (LND). In addition it contains record numbers of associated gateway, call modifier and extension records.

NOTE: When calling to phone numbers contained in this EF from within a TETRA network, the PSTN/PABX gateway address is added in front of the phone number. This implementation requires that there is one universally acknowledged TETRA address for PSTN gateways in all different networks.

Identifier: '6F22'	Structure: cyclic		Optional
Record length: X + 12 bytes		Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1 to X	Name	O	X
X + 1	Length of PSTN or PABX number contents	M	1
X + 2 to X + 9	PSTN or PABX number	M	8
X + 10	Gateway address record number	M	1
X + 11	Call modifier record number	M	1
X + 12	Extension1 record number	M	1

Contents and coding: see EF$_{ADN}$.

10.3.35 EF$_{LNDTETRA}$ (Last numbers dialled for TETRA network)

EF contains the last numbers dialled to TETRA phones within TETRA network.

Identifier: '6F23'	Structure: cyclic		Optional
Record length: X + 6 bytes		Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/ O	Length
1 to X	Name	M	X
X + 1	Network address record number	M	1
X + 2 to X + 4	SSI of TETRA address or Supplementary service access string	M	3
X + 5	Call modifier record number	M	1
X + 6	Extension A record number	M	1

For contents and coding of all data items see the respective data items of the EF$_{ADNTETRA}$.

10.3.36 EF$_{SDN}$ (Service Dialling Numbers)

This EF contains the special user-non-modifiable Service Dialling Numbers (SDN) that are used when calling to a phone outside the TETRA network. In addition it contains record numbers of associated gateway, call modifier and extension records.

NOTE: When calling to telephones contained in this EF from within a TETRA network, the PSTN/PABX gateway address is added in front of the phone number. This implementation requires that there is one universally acknowledged TETRA address for PSTN gateways in all different networks.

Identifier: '6F24'	Structure: linear fixed		Optional
Record length: X + 12 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/ O	Length
1 to X	Name	O	X
X + 1	Length of PSTN or PABX number contents	M	1
X + 2 to X + 9	PSTN or PABX number	M	8
X + 10	Gateway address record number	M	1
X + 11	Call modifier record number	M	1
X + 12	Extension3 record number	M	1

For contents and coding of all data items see the respective data items of the EF$_{ADN}$ (see subclause 10.3.25), with the exception that extension records are stored in the EF$_{EXT3}$.

10.3.37 EF$_{EXT3}$ (Extension3)

This EF contains extension data of an SDN (see Extension3 record number in subclause 10.3.36). Extension data is caused by an SDN which is greater than the 16 digit capacity of the SDN EF. The remainder is stored in this EF as a record, which is identified by a specified identification byte inside the SDN EF.

Identifier: '6F25'	Structure: linear fixed		Optional
Record length: 13 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	Length of extension data	M	1
2 to 12	Extension data	M	11
13	Next record number	M	1

For contents and coding see subclause 10.3.27.

10.3.38 EF$_{SDNTETRA}$ (Service Dialling Numbers for TETRA network)

EF contains the user-non-modifiable phone numbers that are used when calling to a TETRA phone.

Identifier: '6F26'	Structure: linear fixed		Optional
Record length: X + 5 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1 to X	Name	M	X
X + 1	Network address record number	M	1
X + 2 to X + 4	SSI of TETRA address	M	3
X + 5	Call modifier record number	M	1

For contents and coding of all data items see the respective data items of the EF$_{ADNTETRA}$.

10.3.39 EF$_{STXT}$ (Status message texts)

This EF contains text strings to be displayed upon receipt of precoded status message.

Identifier: '6F27'	Structure: linear fixed	Optional
Record length: X + 2 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-2	Message value	M	2
3 to X + 2	Message text	M	X

- Message value:

Contents: The message value identifies the actual message.

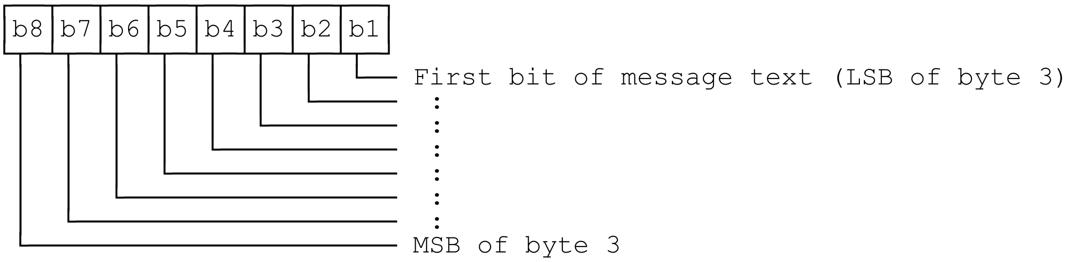
Coding: The message value is coded with two bytes as defined in ETS 300 392-2 [11], subclause 14.8.34. A reserved ('0001'-'7FFF') value indicates an empty record.

- Message text:

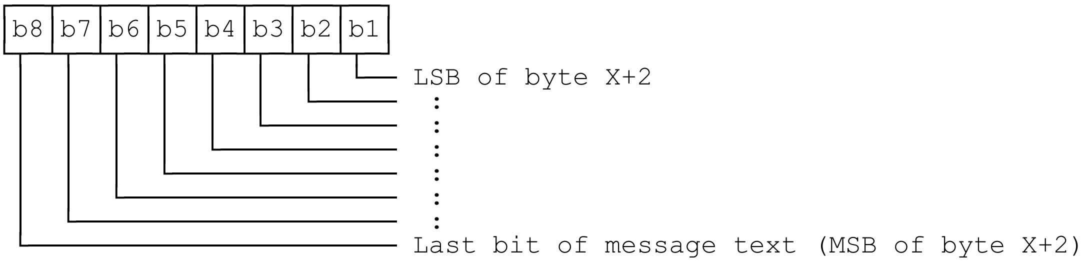
Contents: The message text contains the text string corresponding the message value and it is shown to the user instead of or with the message value.

Coding: The string shall use the default 8-bit alphabet ISO 8859-1 [22]. The message text is coded with X bytes. If the text is shorter than X bytes, the remaining bytes shall be filled with FF.

Byte 3:



Byte X+2:



NOTE: Of the precoded status messages only messages above and including the value of 32 768 are stored in this EF.

10.3.40 $\text{EF}_{\text{MSGTXT}}$ (SDS-1 message texts)

This EF contains text strings to be displayed upon receipt of an SDS-1 (user defined data 1) message.

Identifier: '6F28'	Structure: linear fixed		Optional
Record length: X + 2 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1-2	Message value	M	2
3 to X + 2	Message text	M	X

- Message value:

  Contents: The message value identifies the actual message.

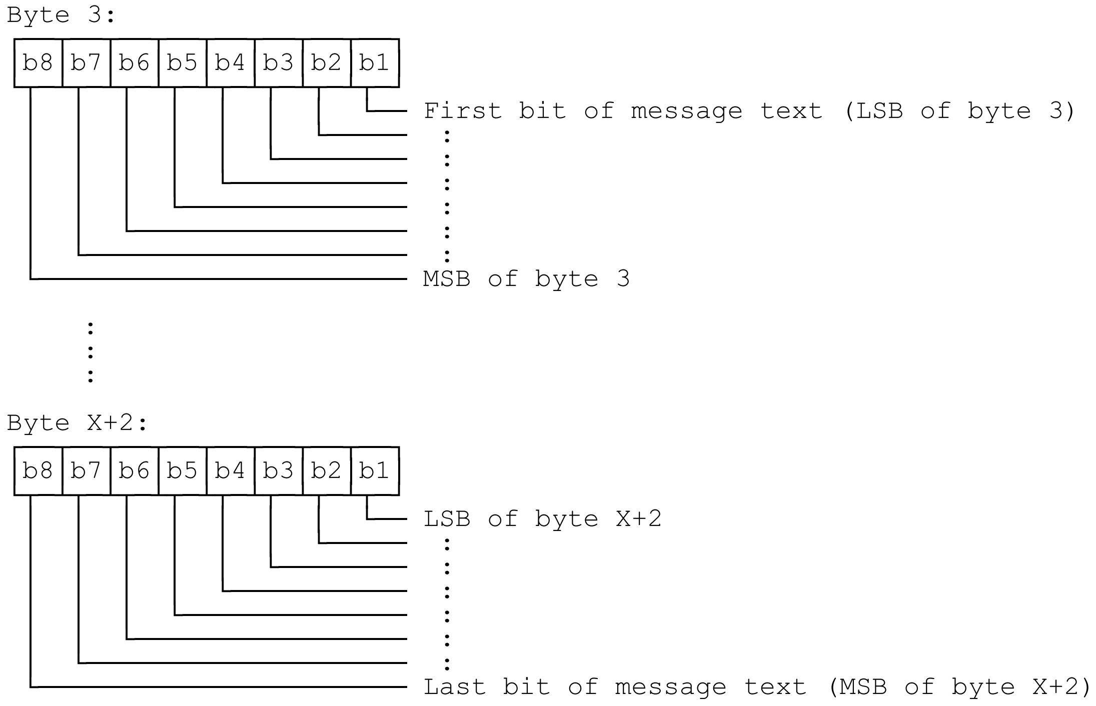
  Coding: The message value is coded with two bytes as defined in ETS 300 392-2 [11], subclause 14.8.49.

NOTE: User application knows which Message values are valid, because all values have been reserved for user application. Therefore the user application also knows which records contain valid data.

- Message text:

  Contents: The message text contains the text string corresponding the message value and it is shown to the user instead of or with the message value.

  Coding: The string shall use the default 8-bit alphabet ISO 8859-1 [22]. The message text is coded with X bytes. If the text is shorter than X bytes, the remaining bytes shall be filled with FF.



NOTE: The SDS-1 messages are applicable to the user's home network only.

10.3.41 EF$_{SDS123}$ (Status and SDS type 1, 2 and 3 message storage)

This EF contains the numerical values of Status messages and SDS type 1, 2 or 3 messages (and associated parameters) which have either been received by the MS from the network, or are to be used as MS originated messages.

Identifier: '6F29'	Structure: linear fixed	Optional
Record length: 21 bytes	Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Message status	M	1
2-5	Message destination/source identifier (note 1)	M	4
6-21	Message header and message (note 2)	M	16

NOTE 1: Contains called party address for received messages; calling party address for transmit messages.

NOTE 2: Contains calling party address for received messages; called party address for transmit messages.

- Message status:

Contents: Status of the message stored.

Coding: byte 1.

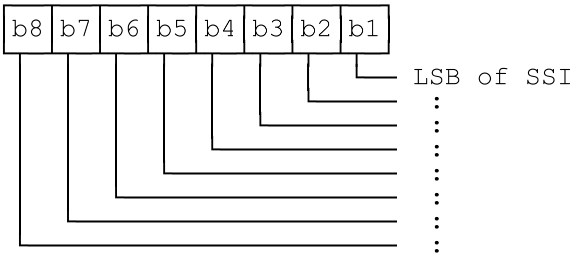
```
b8 b7 b6 b5 b4 b3 b2 b1
               0  0  0   Record not used
               0  1  0   RFU
               1  0  0   RFU
               1  1  0   RFU
               x  x  1   used space
               0  0  1   message received by MS from network;
                                message read
               0  1  1   message received by MS from network;
                                message to be read
               1  0  1   MS originating message;
                                message sent to the network
               1  1  1   originating message; message to be
                                sent

                         RFU
```

- Message destination/source identifier:

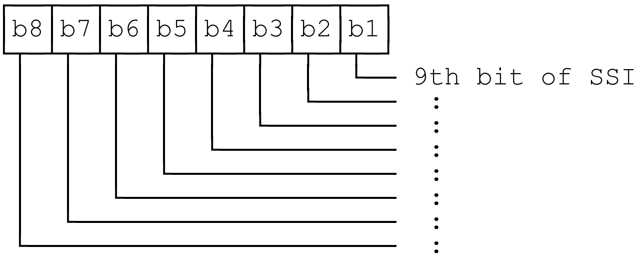
Contents: This data item contains the called party address for received messages or the calling party address for transmitted messages. Received messages could have been directed to any of the associated GTSIs so the destination address needs to be clearly identified. MS originated messages are always identified with an individual address, usually the ITSI but in some circumstances it could be an ATSI or a VTSI.

Coding:

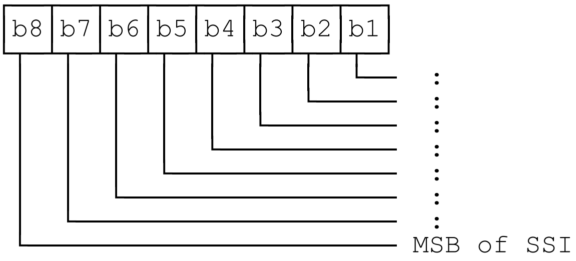
Byte 2:



Byte 3:



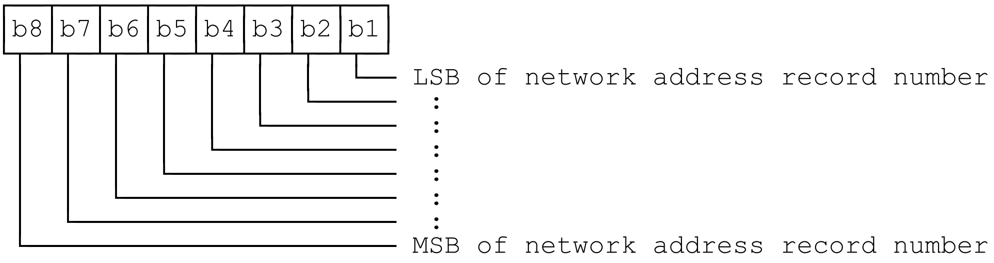
Byte 4:



- Network address record number:

Contents: One byte index pointing to the record number of the corresponding network address. Network addresses are stored in EF_{NWT}.

Coding: byte 5


b8 b7 b6 b5 b4 b3 b2 b1
LSB of network address record number
MSB of network address record number

- Message header and message: Contains information on transmitted or received messages.

Contents: In the case of received status and SDS123 messages this data item contains the downlink Status message header and status code (as defined in ETS 300 392-2 [11], subclause 14.7.1.11) or the downlink SDS types 1, 2 or 3 message header plus message (as defined in ETS 300 392-2 [11], subclause 14.7.1.10).

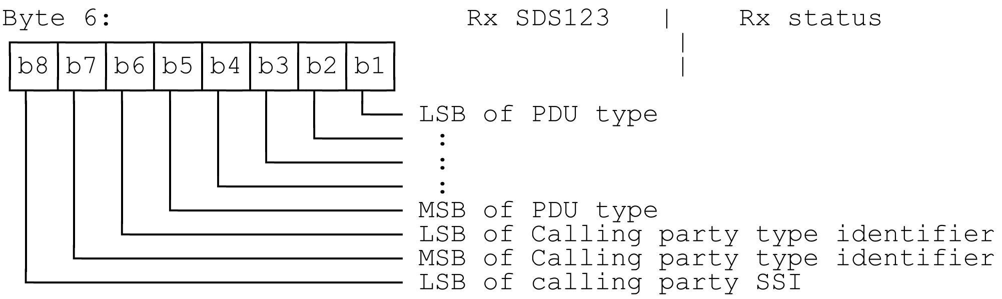
Coding: According to ETS 300 392-2 [11], subclauses 14.7.1.10, 14.7.1.11, 14.7.2.7 and 14.7.2.8.

NOTE 1: The longest conditional message header fields are shown below to indicate the maximum expected EF storage requirements. The message header and message will be stored by the SIM in the format received over the air interface.

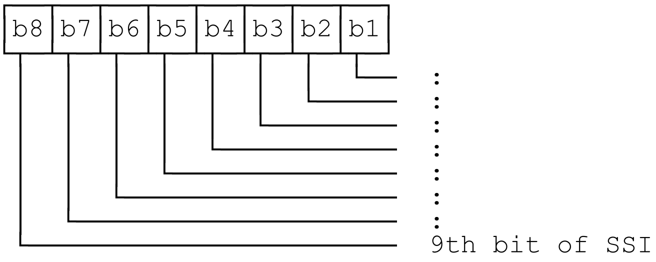
NOTE 2: The separation of coding between Receive (Rx) SDS123 and Rx Status is shown from byte 12 onward.

NOTE 3: All unused bytes following the PDUs shall be filled with 'FF'.

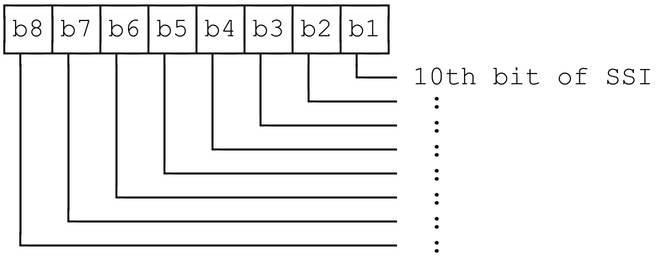
Byte 6:



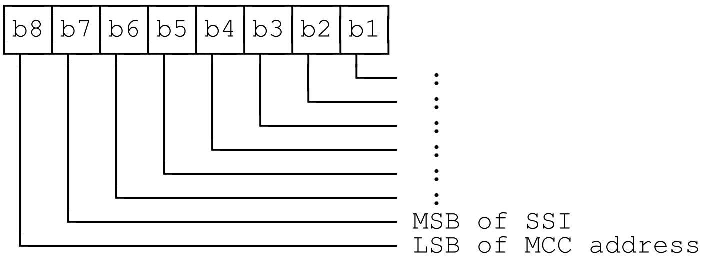
Byte 7:



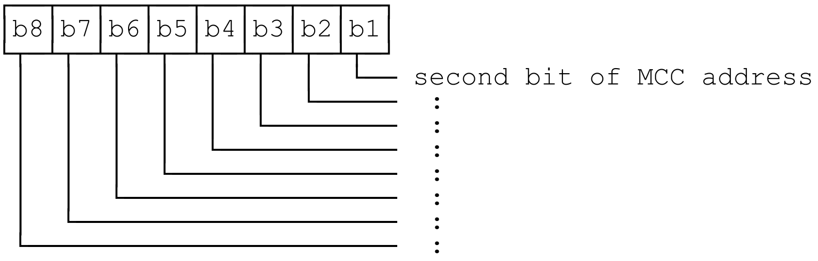
Byte 8:



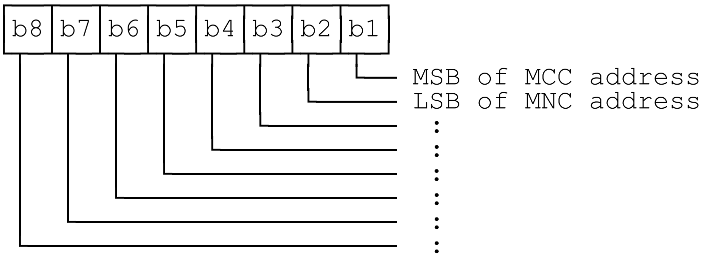
Byte 9:



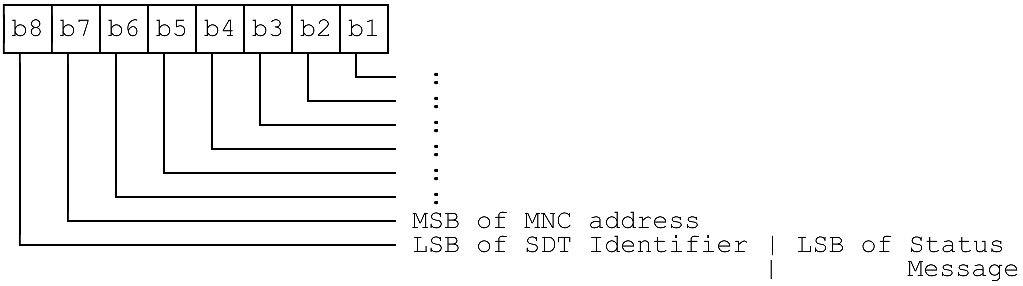
Byte 10:



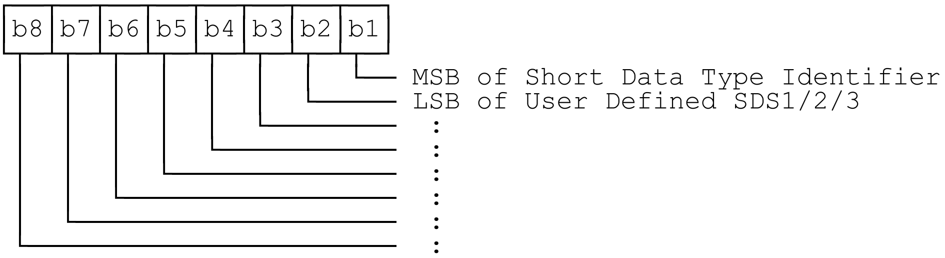
Byte 11:



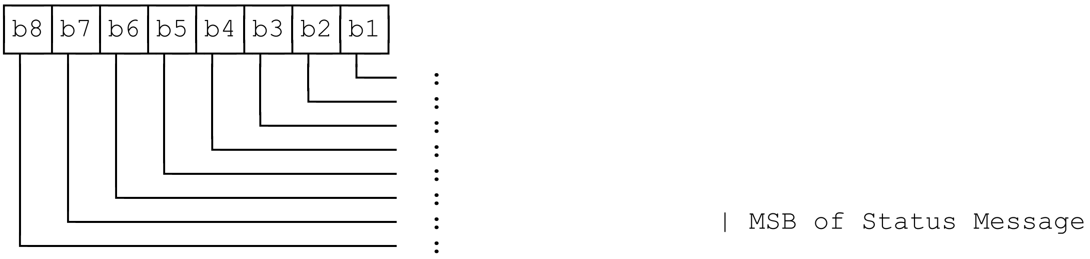
Byte 12:



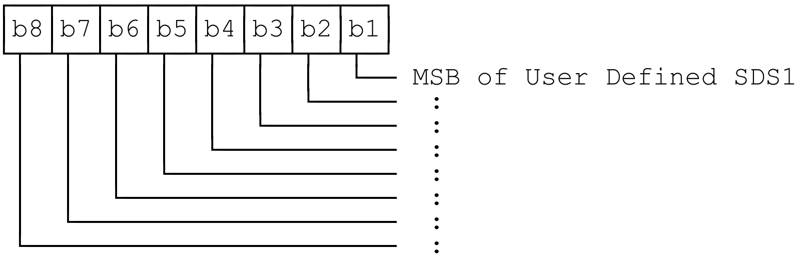
Byte 13:


b8
b7
b6
b5
b4
b3
b2
b1
MSB of Short Data Type Identifier
LSB of User Defined SDS1/2/3

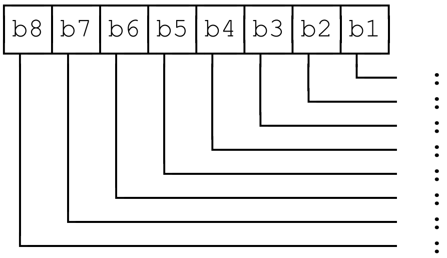
Byte 14:



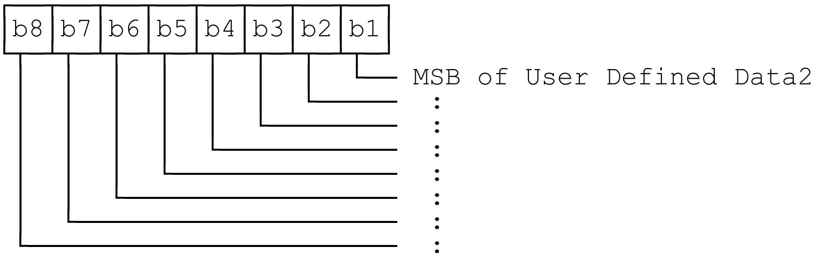
Byte 15:



Byte 16:

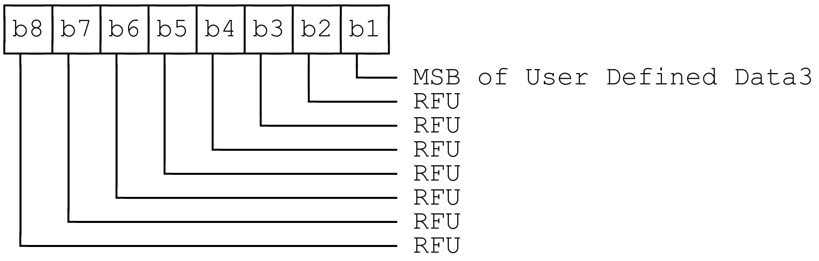


Byte 17:



:
:
:

Byte 21:



Message header and message:

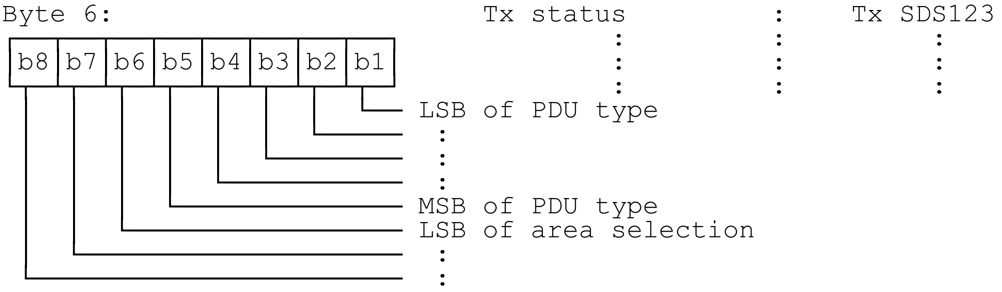
Contents: In the case of transmit SDS123 and Status messages this data item contains the uplink Status message header and code (as defined in ETS 300 392-2 [11], subclause 14.7.2.7) or the uplink SDS types 1, 2 or 3 message header plus message (as defined in ETS 300 392-2 [11], subclause 14.7.2.8).

NOTE 1: The longest conditional message header fields are shown below to indicate the maximum expected EF storage requirements. The message header and message will be stored by the SIM in the format to be sent over the air interface.

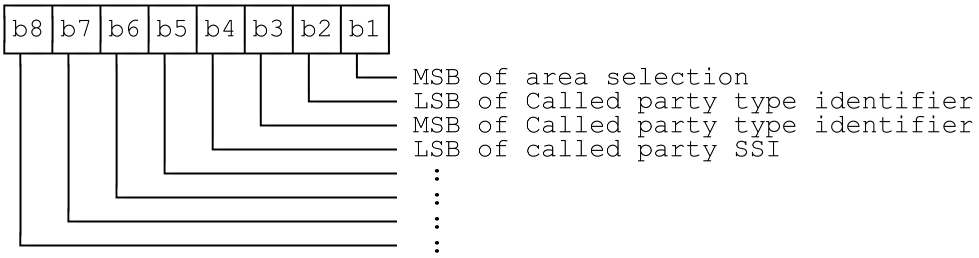
NOTE 2: The separation of coding between Transmit (Tx) SDS123 and TX Status is shown from byte 13 onward.

Coding: According to ETS 300 392-2 [11], subclauses 14.7.1.10, 14.7.1.11, 14.7.2.7 and 14.7.2.8. All unused bytes following the PDUs shall be filled with 'FF'.

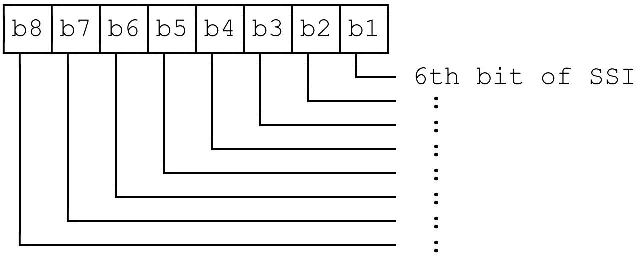
Byte 6:



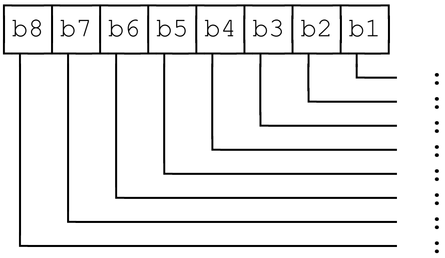
Byte 7:


b8
b7
b6
b5
b4
b3
b2
b1
MSB of area selection
LSB of Called party type identifier
MSB of Called party type identifier
LSB of called party SSI
:
:
:
:

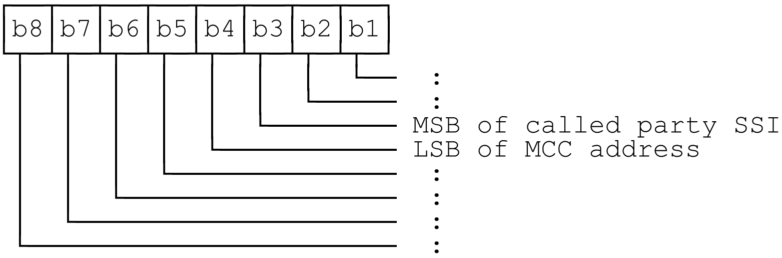
Byte 8:



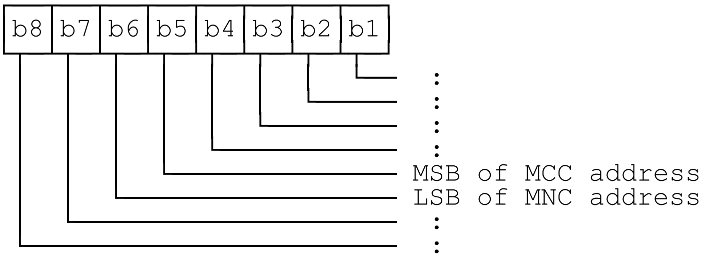
Byte 9:



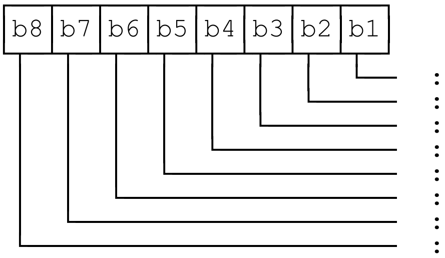
Byte 10:



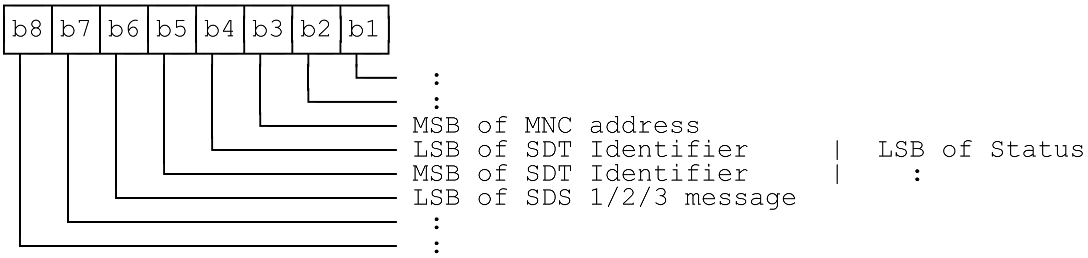
Byte 11:



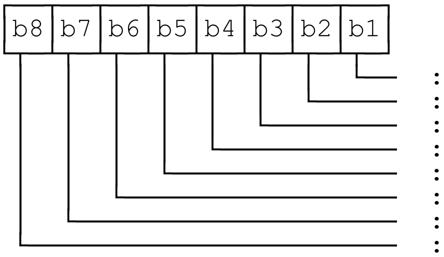
Byte 12:



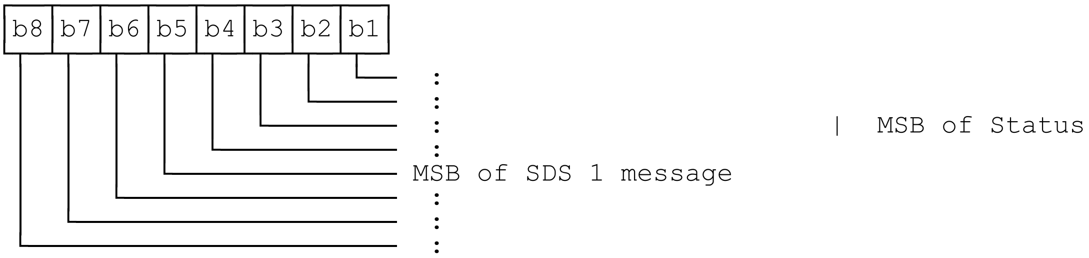
Byte 13:



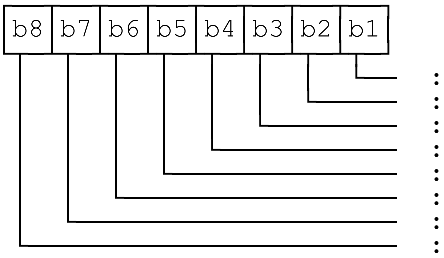
Byte 14:



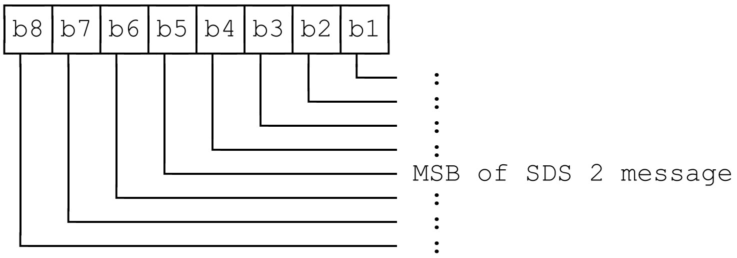
Byte 15:



Byte 16:

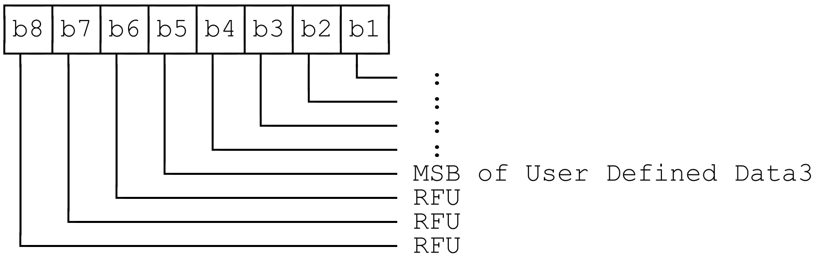


Byte 17:



:
:
:

Byte 21:



10.3.42 EF_{SDS4} (SDS type 4 message storage)

This EF contains text strings (and associated parameters) which have either been received by the MS from the network, or are to be used as an MS originated message.

Identifier: '6F2A'	Structure: linear fixed	Optional
Record length: 255 bytes	Update activity: high	

Access Conditions:

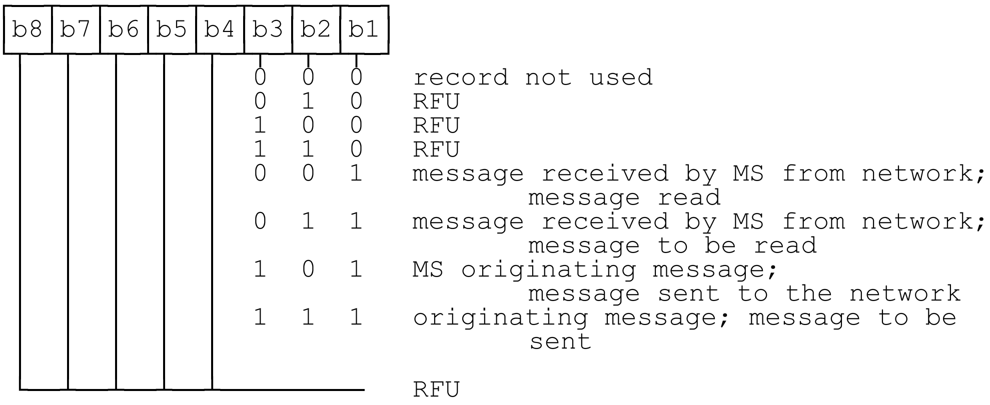
READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Message status	M	1
2-5	Message destination/source	M	4
6-254	Message header and message	M	249
255	Message extension record number	O	1

- Message status:

Contents: Status of the message stored.

Coding:



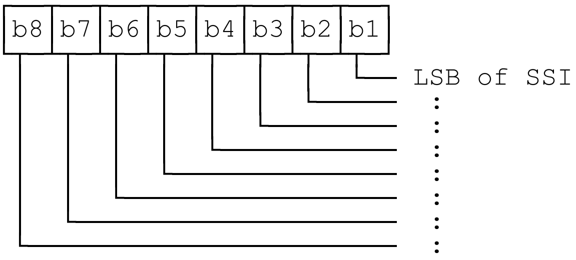
```
b8 b7 b6 b5 b4 b3 b2 b1
               0  0  0   record not used
               0  1  0   RFU
               1  0  0   RFU
               1  1  0   RFU
               0  0  1   message received by MS from network;
                                message read
               0  1  1   message received by MS from network;
                                message to be read
               1  0  1   MS originating message;
                                message sent to the network
               1  1  1   originating message; message to be
                                sent

                         RFU
```

- Message destination/source identifier:

Contents: This data item contains the called party address for received messages or the calling party address for transmitted messages. Received messages could have been directed to any of the associated GTSIs so the destination address needs to be clearly identified. MS originated messages are always identified with an individual address, usually the ITSI, but in some circumstances it could be an ATSI or a VTSI.

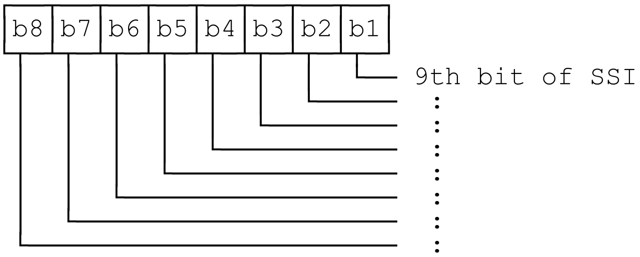
Coding:

Byte 2:

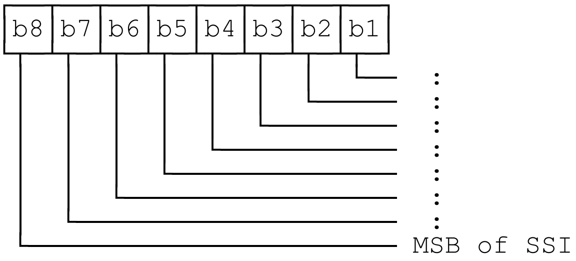


```
b8 b7 b6 b5 b4 b3 b2 b1
                        LSB of SSI
                          :
                          :
                          :
                          :
                          :
                          :
                          :
```

Byte 3:



Byte 4:

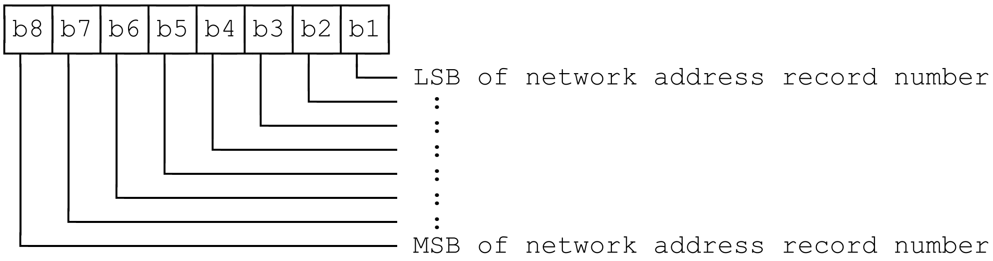


- Network address record number:

Contents: One byte index pointing to the record number of the corresponding network address. Network addresses are stored in EF_{NWT}.

Coding:

Byte 5:


b8
b7
b6
b5
b4
b3
b2
b1
LSB of network address record number
MSB of network address record number

- Message header and message: Received or transmit SDS4 messages.

Contents: This data item contains the downlink or uplink SDS type 4 message header plus message (as defined in ETS 300 392-2 [11], subclauses 14.7.1.10 and 14.7.2.8).

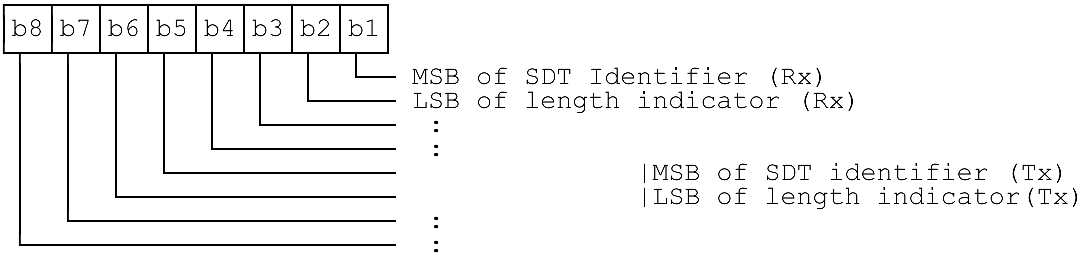
Coding: According to ETS 300 392-2 [11], subclauses 14.7.1.10 and 14.7.2.8.

NOTE 1: The longest conditional message header fields are shown below to indicate the maximum expected EF storage requirements. The message header and message is stored by the SIM in the format received/to be sent over the air interface.

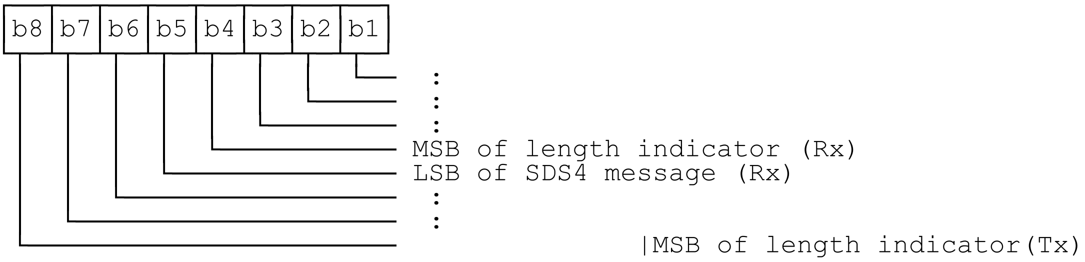
NOTE 2: All unused bytes following the PDUs shall be filled with 'FF'.

Bytes 6 to 12 as in subclause 10.3.41

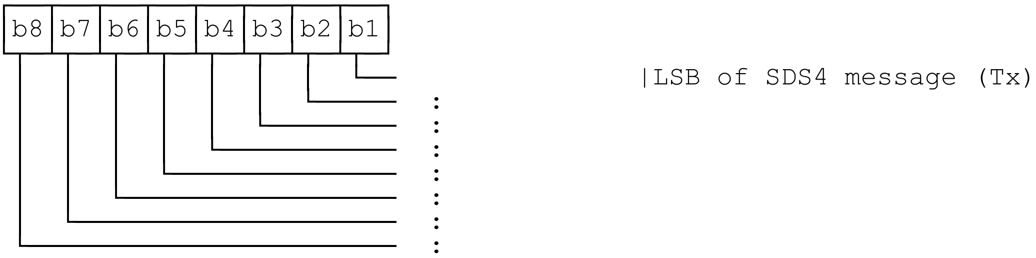
Byte 13:


b8 b7 b6 b5 b4 b3 b2 b1
MSB of SDT Identifier (Rx)
LSB of length indicator (Rx)
|MSB of SDT identifier (Tx)
|LSB of length indicator(Tx)

Byte 14:

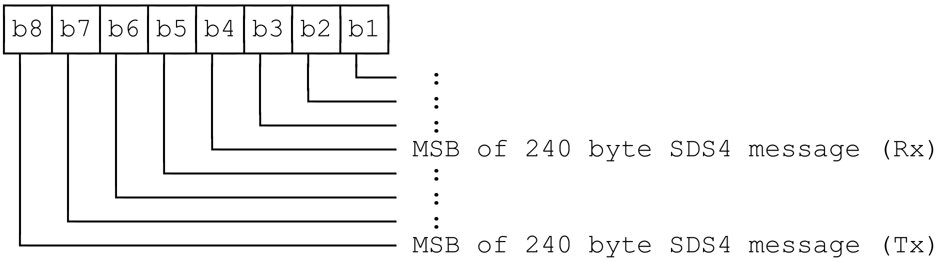


Byte 15:



:
:
:

Byte 254:


b8 b7 b6 b5 b4 b3 b2 b1
MSB of 240 byte SDS4 message (Rx)
MSB of 240 byte SDS4 message (Tx)

- Byte 255 Message Extension record number:

Contents: This byte identifies the number of a record in the EF_{MSGEXT} containing an associated message overflow. The use of this byte is optional. If it is not used, it shall be set to 'FF'.

Coding: Binary.

10.3.43 EF_{MSGEXT} (Message Extension)

This EF contains the overflow of an SDS-4 message which is longer than the space reserved for it in EF_{SDS4}.

Identifier: '6F2B'	Structure: linear fixed	Optional
Record length: 16 bytes	Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-16	Overflow message	M	16

- Overflow message:

  Contents: Overflow data of an SDS-4 message exceeding the length reserved for it in $EF_{SDS4.}$

  Coding: As defined in ETS 300 392-2 [11], subclauses 14.7.1.10. and 14.7.2.8. All bytes following the PDUs shall be filled with 'FF'.

NOTE: A free record is not pointed to by any record in EF_{SDS4}.

10.3.44 EF_{EADDR} (Emergency addresses)

The user (or the organization) can determine the address to which an emergency call is initiated; to a predetermined address or to the group last used by the user. The selection is controlled by the addresses stored in EF_{EADDR}.

Where a data call type is selected, the ESource field indicates the preferred source of the data to be included in the message for status, SDS-1, SDS-2, SDS-3 and SDS-4 messages. In each case the data content can be a pre-defined value stored in EF_{SDS123} or EF_{SDS4} (or a data field obtained from an application running in the terminal).

Identifier: '6F2C'	Structure: linear fixed	Mandatory
Record size: 17 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1/CHV2 (see note) INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Emergency call definition	M	1
2-17	Emergency address	M	16
NOTE: Card issuer will choose between CHV1 or CHV2 protection.			

- Emergency call definition:

  Contents: One byte indicating the call type and the emergency address type coded on the Emergency address field, and the source of the message content for status and data calls.

  Coding:

  b1-b4: Emergency address type
  b5: Source of the data to be transmitted in the emergency data message
  b6-b7: Emergency call type

NOTE: An empty record is indicated by NULL ('F') value in bits b1-b4.

b8	b7	b6	b5	b4	b3	b2	b1	
				0	0	0	0	TETRA address
				0	0	0	1	DMO address
				0	0	1	0	PABX address (gateway and External subscriber number)
				0	0	1	1	PSTN number (gateway and External subscriber number)
				0	1	0	0	Last active group address
				0	1	0	1	RFU
				0	1	1	0	RFU
				0	1	1	1	RFU
				1	0	0	0	Status/SDS123 msg record number
				1	0	0	1	SDS4 message record number
				1	0	1	0	RFU
				1	0	1	1	RFU
				1	1	0	0	RFU
				1	1	0	1	RFU
				1	1	1	0	RFU
				1	1	1	1	Record contains no valid data
			0					Predefined and stored in EF_{EADDR}
			1					From an application in the terminal
	0	0						Point-to-Point
	0	1						Point-to-Multipoint
	1	0						Point-to-Multipoint acknowledged
	1	1						Broadcast
								RFU

- Emergency address:

  Contents: The address that can be used when the user initiates an emergency call. The type of call is determined by byte 1.

  In the case of a TETRA address the emergency address consists of the ITSI (or GTSI) of the called party.

  In the case of a DMO address the emergency address consists of the ITSI (or GTSI) of the called party and the DMO channel number.

  In the case of a PABX address the emergency address consists of the full PABX Gateway ITSI and the External Subscriber number.

  In the case of a PSTN address the emergency address consists of the telephone number as defined in EF_{ADN}. The call is handled as a normal call to the PSTN network, i.e. the PSTN gateway address is used.

  In the case of the last active group address, the address field in EF_{EADDR} is unused - the address for the emergency call should be obtained from EF_{GINFO}.

  In the case of status, SDS-1, SDS-2, SDS-3 and SDS-4 messages the content of this data item consists of the message record number in SDS123 or SDS4 as appropriate.

Coding:

In the case of a TETRA address, according to EF_{ITSI}.

In the case of a DMO address, according to EF_{ITSI} followed by the 24 bit DMO channel number, coded according to EF_{DMOCh}

In the case of a PABX number, the Gateway ITSI is coded according to EF_{ITSI} and the External Subscriber number is BCD coded as defined in ETS 300 392-2 [11], subclause 14.8.20.

In the case of a PSTN number, the Gateway ITSI is coded according to EF_{ITSI} and the external PSTN address is BCD coded according to ETS 300 392-2 [11], subclause 14.8.20.

In the case of the last used group address, this field is unused - the address for the call to be obtained from EF_{GINFO}.

NOTE: The emergency addresses are stored in order of precedence.

10.3.45 EF_{EINFO} (Emergency call information)

This EF contains information about setting up and continuing an emergency call.

Identifier: '6F2D'	Structure: transparent	Mandatory
File size: 2 bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Emergency call continuation	M	1
2	Current emergency call record number	M	1

- Emergency call continuation:

Contents: A flag indicating whether an interrupted emergency call should continue at power-on.

Coding:

```
Byte 1:
┌──┬──┬──┬──┬──┬──┬──┬──┐
│b8│b7│b6│b5│b4│b3│b2│b1│
└──┴──┴──┴──┴──┴──┴──┴──┘
 │  │  │  │  │  │  │  0   Emergency call should continue
 │  │  │  │  │  │  │  1   Emergency call should not continue
 └──┴──┴──┴──┴──┴──┴────── RFU
```

- Current emergency call record number:

Contents: One byte field available to the emergency application to store on the SIM information pertaining to an emergency call in progress, typically to cater for the possibility of unexpected power-down. It may be the record number of the record in EF_{EADDR} used to set up the emergency call currently in progress. A zero value indicates that no call is in progress.

Coding: Binary.

10.3.46 EF_{DMOCh} (DMO channel information)

This EF contains a selection of DMO channels. One or more of the channels may be designated as emergency channel(s) to be used for emergency calls within DMO operation.

Identifier: '6F2E'	Structure: linear fixed		Optional
Record size: 4 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	**Description**	**M/O**	**Length**
1	DMO channel type	M	1
2-4	DMO channel number	M	3

- DMO channel type:

  Contents: This field contains the DMO channel type information.

  Coding: The type is coded in the first bit of the first byte: emergency='01'; regular='00'. NULL ('FF') value indicates an empty record. All other values are reserved.

- DMO channel number:

  Contents: This field contains the DMO channel definition.

  Coding: As defined in 300 396-3 [13].

10.3.47 EF_{MSCh} (MS allocation of DMO channels)

This EF contains a bitmap which allocates a subset of the DMO channels in EF_{DMOCh}. There shall be one bit corresponding to each record in EF_{DMOCh}.

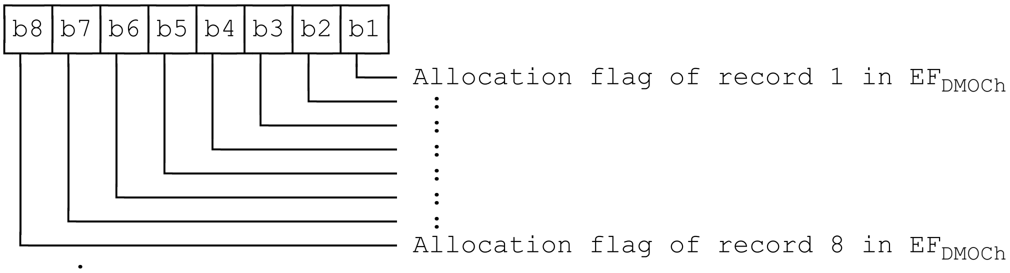
Identifier: '6F2F '	Structure: transparent		Optional
File size: X bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	**Description**	**M/O**	**Length**
1	Allocation flag 1 to 8	M	1
X	Allocation flag 8*X-7 to 8*X	M	1

NOTE: The value of X should be sufficiently large to accommodate all the records in EF_{DMOCh}.

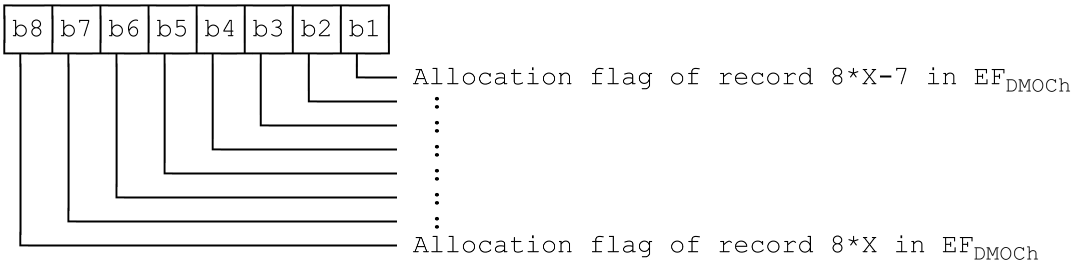
- Allocation flag:

Coding: Channel is allocated=1, channel is not allocated=0.

Byte 1:


b8
b7
b6
b5
b4
b3
b2
b1
Allocation flag of record 1 in EF_{DMOCh}
Allocation flag of record 8 in EF_{DMOCh}

Byte X:


b8
b7
b6
b5
b4
b3
b2
b1
Allocation flag of record 8*X-7 in EF_{DMOCh}
Allocation flag of record 8*X in EF_{DMOCh}

10.3.48 EF_{KH} (List of Key Holders)

This EF contains a list of those ITSI numbers that can act as a key holder for this subscriber's ITSI.

Identifier: '6F30 '	Structure: transparent		Optional
Record size: 6 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1 - 6	Key holder ITSI	M	6

- Key holder ITSI;

Contents: Key holder ITSI consists of MCC, MNC and ISSI.

Coding: As in EF_{ITSI}. Record filled with NULL ('FF') value indicates no ITSI is stored.

10.3.49 EF$_{REPGATE}$ (DMO repeater and gateway list)

This EF contains a list of those DMO repeaters, gateways and REP/GATEs that this subscriber is allowed to use. Each address is 10 bits long. DMO equipment type is also identified.

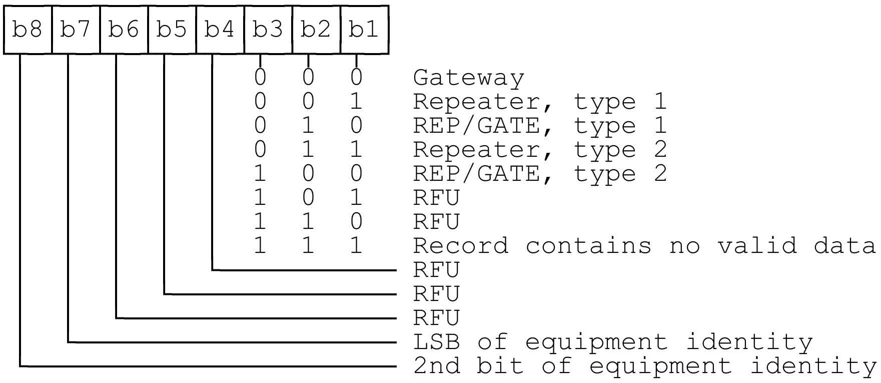
Identifier: '6F31'	Structure: linear fixed		Optional
Record size: 2 bytes	Update activity: low		
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	**Description**	**M/O**	**Length**
1-2	DMO equipment type and identity	M	2

- DMO equipment type and identity:

Contents: This field contains the DMO equipment type and the first part of its identity.

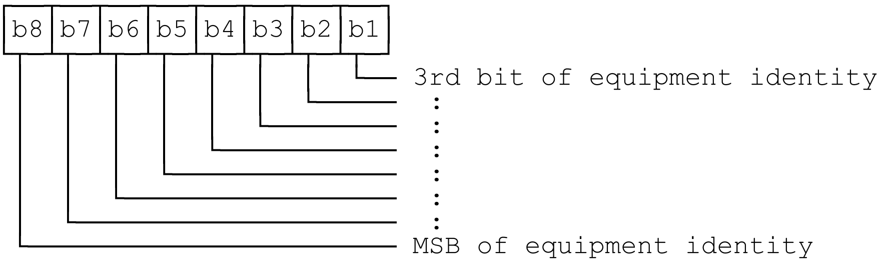
Coding:

Byte 1:



```
b8 b7 b6 b5 b4 b3 b2 b1
               0  0  0   Gateway
               0  0  1   Repeater, type 1
               0  1  0   REP/GATE, type 1
               0  1  1   Repeater, type 2
               1  0  0   REP/GATE, type 2
               1  0  1   RFU
               1  1  0   RFU
               1  1  1   Record contains no valid data
                         RFU
                         RFU
                         RFU
                         LSB of equipment identity
                         2nd bit of equipment identity
```

Byte 2:



```
b8 b7 b6 b5 b4 b3 b2 b1
                         3rd bit of equipment identity
                          :
                          :
                          :
                          :
                          :
                          :
                         MSB of equipment identity
```

10.3.50 EF$_{AD}$ (Administrative data)

This EF contains information concerning the mode of operation according to the type of SIM, such as normal operation, type approval (to allow specific use of the ME during type approval procedures of e.g. the radio equipment) or others.

Identifier: '6F32'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	
Access Conditions: READ ALW UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

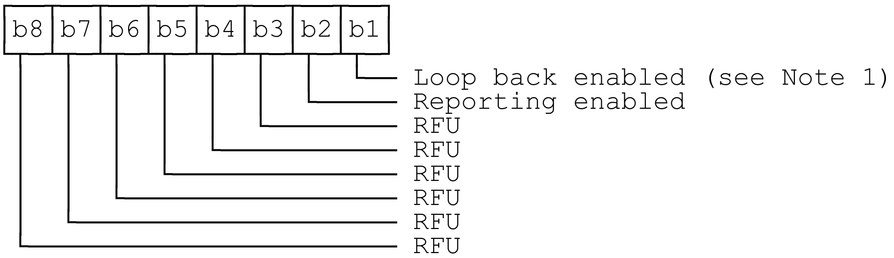
Bytes	Description	M/O	Length
1	MS operation mode	M	1 byte

- MS operation mode:

Contents: mode of operation for the MS.

Coding:

Byte 1:



NOTE 1: Loop back enabled and security/authentication disabled (see ETS 300 394-2 [23]).

NOTE 2: The coding '00' means normal operation.

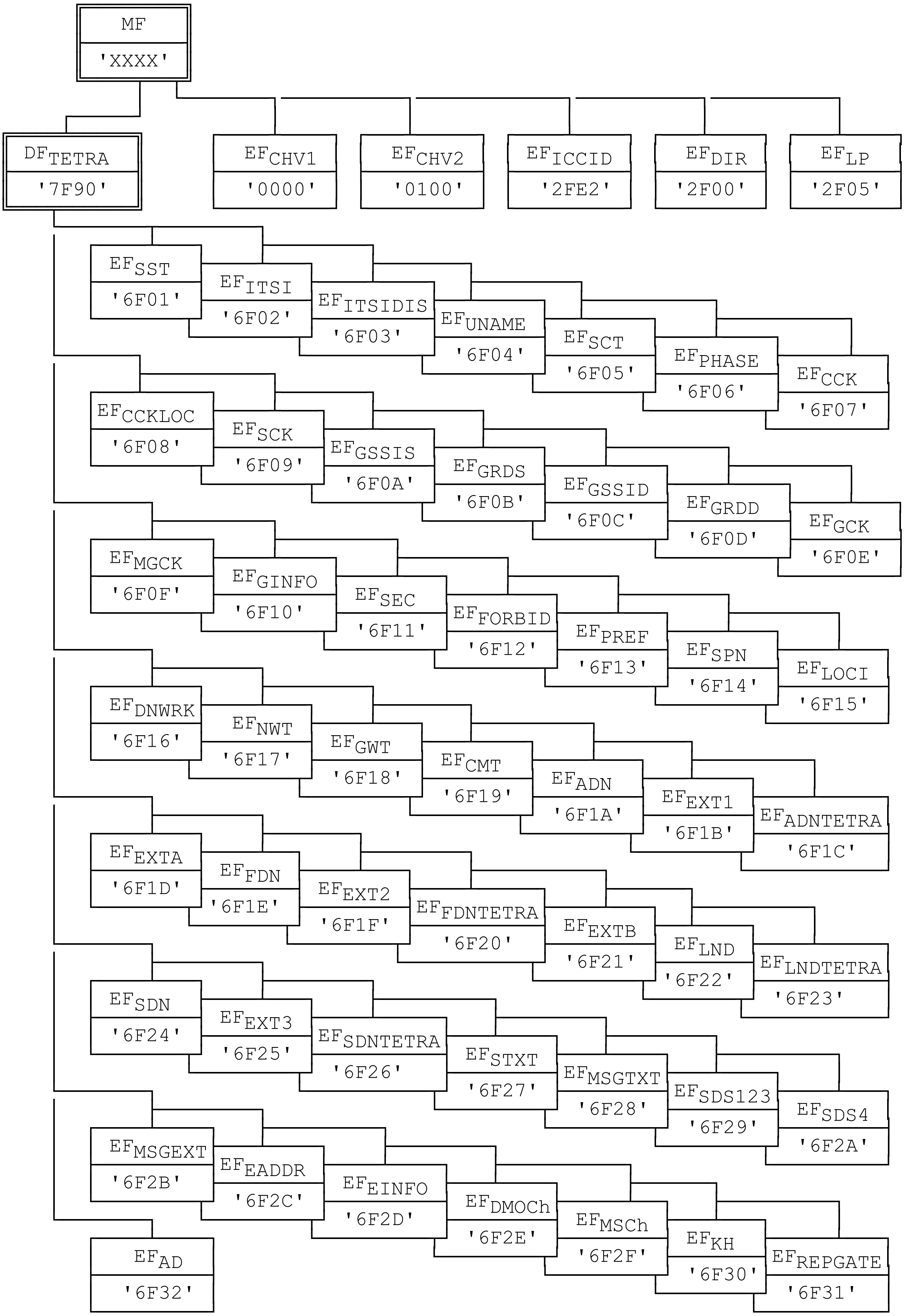


Figure 7: File identifiers and directory structures of TETRA

11 Application protocol

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

When involved in TETRA network operations the SIM interfaces with an ME with which messages are exchanged. A message can be a command or a response as follows:

- a TETRA command/response pair is a sequence consisting of a command and the associated response;

- a TETRA procedure consists of one or more TETRA 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 ME shall ensure that, when operated according to the manufacturer's manual, any unspecified interruption of the sequence of command/response pairs which realize the procedure, leads to the abortion of the procedure itself;

- a TETRA session of the SIM in the TETRA application is the interval of time starting at the completion of the SIM initialization procedure and ending either with the start of the TETRA session termination procedure, or at the first instant the link between the SIM and the ME is interrupted.

During the TETRA network operation phase, the ME plays the role of the master and the SIM plays the role of the slave.

Some procedures at the SIM/ME interface require 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 MS and the network. Such procedures are marked NETwork "(NET)" in the list given below.

Some procedures are automatically initiated by the ME. They are marked "ME" in the list given below.

The list of procedures at the SIM/ME interface in TETRA network operation is as follows:

General Procedures:

-	Reading an EF	ME;
-	Updating an EF	ME.

SIM management procedures:

-	SIM initialization	ME;
-	TETRA session initialization	ME;
-	TETRA session termination	ME;
-	Language preference request	ME;
-	Administrative information request	ME;
-	SIM service table request	ME;
-	SIM phase request	ME;
-	SIM presence detection	ME.

CHV related procedures:

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

TETRA security related procedures:

- TETRA algorithms computation NET;
- TETRA key computation (SCK, DCK, MGCK, GCK) NET;

ITSI request NET;

- ITSI disabling NET;
- Location Information NET;
- Broadcast network information NET;
- Forbidden networks information NET.

Subscription related procedures:

- Username MMI;
- Subscriber class request ME;
- Group information MMI/NET;
- User's group information ME/NET;
- Call modifiers NET/ME;
- Network information ME;
- Dialling Numbers (AND, ADNTETRA, ADNPABX, FDN, FDNTETRA, FDNPABX, LND, LNDTETRA, LNDPABX, SDN, SDNTETRA, SDNPABX) MMI/ME;
- SDS messages (Message texts, SDS123 and SDS4) MMI;
- Preferred networks MMI;
- Service Provider Name (SPN) ME;
- ICCID ME;
- Emergency addresses ME/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. The procedures listed in subclauses 11.5, 11.6, 11.7 and 11.8 are only executable if the associated services, which are optional, are provided in the SIM. However, if the procedures are implemented, it shall be in accordance with subclauses 11.4, 11.6, 11.7 and 11.8.

If a procedure is related to a specific service indicated in the SIM service table, it shall only be executed if the corresponding bit denoting this service as "available" (see EF_{SST}). In all other cases this procedure shall not start.

11.1 General procedures

11.1.1 Reading an EF

The ME 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 SIM sends the requested data contained in the EF to the ME. 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 ME 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 SIM updates the selected EF by 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 Invalidating an EF

The ME selects the EF and sends an INVALIDATE command. If the access conditions of INVALIDATE are fulfilled the EF is invalidated.

11.2 SIM management procedures

11.2.1 SIM initialization

The ME runs the language request procedure. If none of the indicated languages are available, a default language (e.g. English) is selected by the ME. The ME checks the presence of a EF_{CHV1} at master file level.

The ME selects EF_{DIR} and obtains the read access condition. If the read access condition is CHV, the ME runs the CHV verification procedure for CHV1.

11.2.2 TETRA session initialization

Following the SIM initialization, the ME selects DF_{TETRA} by using the identifier or by the path given in EF_{DIR}. After that, the ME selects the EF_{ITSI} to obtain its INVALIDATION status. If the ITSI is invalidated the ME informs the user and the TETRA session initialization fails.

The ME checks the presence of an EF_{CHV1} at TETRA application level. If present, it runs the CHV verification procedure for CHV1. If no EF_{CHV1} is present at TETRA application level, and the ME has not yet verified the CHV at master file level (see subclause 11.2.1) then this action is performed now. If the CHV verification is unsuccessful, the TETRA session initialization fails.

NOTE: If there is no EF_{CHV1} present at the application level, there has to be one at the master file level. For convenience of the user, implementations having both an EF_{CHV} at application and at master file level should be avoided.

If the CHV verification procedure is performed successfully, the ME then runs the following procedures:

- Administrative information request;
- SIM Phase request;
- SIM Service Table request;
- ITSI request;
- ITSI temporarily disabled enquiry;
- Subscriber class request;
- Preferred networks request;
- Location Information request;
- Mutual authentication requirement request;
- Forbidden networks request;
- Interrupted emergency call request.

After the SIM initialization has been completed successfully, the MS is ready for a TETRA session.

NOTE: If the ITSI is "Temporary disabled by SwMI", the ME enters a TETRA session with a restricted mode of operation. The restricted TETRA session usually consists of the MS simply listening to the SwMI to eventually detect a re-enabling of the ITSI by the network (see ETS 300 392-7 [12]).

11.2.3 TETRA session termination

NOTE 1: This procedure is not to be confused with the deactivation procedure in subclause 4.3.2.

The TETRA session is terminated by the ME as follows:

The ME runs all the procedures which are necessary to transfer the following subscriber related information to the SIM:

- Location Information update;
- Forbidden networks update.

As soon as the SIM indicates that these procedures are completed, the ME/SIM link may be deactivated.

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

NOTE 2: If the ME has already updated any of the subscriber related information during the TETRA Session, and the value has not changed until TETRA session termination, the ME may omit the respective update procedure.

11.2.4 Language preference request

Request: The ME performs the reading procedure with EF_{LP}.

Update: The ME performs the updating procedure with EF_{LP}.

11.2.5 Administrative information request

Request: The ME performs the reading procedure with EF_{AD}.

Update: The ME performs the updating procedure with EF_{AD}.

11.2.6 SIM service table request

The ME performs the reading procedure with EF_{SST}.

11.2.7 SIM phase request

The ME performs the reading procedure with EF_{PHASE}.

11.2.8 SIM presence detection

As an additional mechanism, to ensure that the SIM has not been removed during a card session, the ME 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. This procedure shall be used in addition to a mechanical or other device used to detect the removal of a SIM.

11.2.9 SIM card number request

The ME performs the reading procedure with EF_{ICCID}.

11.2.10 Common Cipher Key request

The ME performs the read procedure with EF_{CCK} to obtain the current record in this EF.

11.3 CHV related procedures

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

After a third consecutive presentation of a wrong CHV to the SIM, not necessarily in the same TETRA session, the 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.

11.3.1 CHV verification

The ME 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 ME 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 is set "enabled", the ME uses the VERIFY CHV function. If the CHV presented by the ME is equal to the corresponding CHV stored in the SIM, the procedure is finished successfully. If the CHV presented by the ME is not equal to the corresponding CHV stored in the SIM, the procedure ends and is finished unsuccessfully.

After an unsuccessful termination of the CHV verification procedure, the ME repeats the procedure until the procedure either ends successfully, or until the CHV becomes blocked.

11.3.2 CHV value substitution

The ME 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 "enabled", the ME uses the CHANGE CHV function. If the old CHV presented by the ME is equal to the corresponding CHV stored in the SIM, the new CHV presented by the ME is stored in the SIM 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 no.1 "available".

This requirement applies for the CHV1 at the TETRA application level. For the CHV1 at the master file level, it only applies in the case of a mono-application TETRA card.

The ME checks the CHV1 status. If the CHV1 status is "blocked", the procedure ends and is finished unsuccessfully.

If the CHV1 status is not "blocked", the ME reads the CHV1 enabled/disabled indicator. If this is set "disabled", the procedure ends and is finished unsuccessfully.

If the CHV1 status is not "blocked" and the enabled/disabled indicator is set "enabled", the ME uses the DISABLE CHV function. If the CHV1 presented by the ME is equal to the CHV1 stored in the SIM, the status of CHV1 is set "disabled" and the procedure is finished successfully. If the CHV1 presented by the ME is not equal to the CHV1 stored in the SIM, the procedure ends and is finished unsuccessfully.

11.3.4 CHV enabling

The ME checks the CHV1 status. If the CHV1 status is "blocked", the procedure ends and is finished unsuccessfully.

If the CHV1 status is not "blocked", the ME reads the CHV1 enabled/disabled indicator. If this is set "enabled", the procedure ends and is finished unsuccessfully.

If the CHV1 status is not "blocked" and the enabled/disabled indicator is set "disabled", the ME uses the ENABLE CHV function. If the CHV1 presented by the ME is equal to the CHV1 stored in the SIM, the status of CHV1 is set "enabled" and the procedure is finished successfully. If the CHV presented by the ME is not equal to the CHV1 stored in the SIM, 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 ME checks the UNBLOCK CHV status. If the UNBLOCK CHV status is "blocked", the procedure ends and is finished unsuccessfully.

If the UNBLOCK CHV status is not "blocked", the ME uses the UNBLOCK CHV function. If the UNBLOCK CHV presented by the ME is equal to the corresponding UNBLOCK CHV stored in the SIM, the relevant CHV status becomes "unblocked" and the procedure is finished successfully. If the UNBLOCK CHV presented by the ME is not equal to the corresponding UNBLOCK CHV stored in the SIM, the procedure ends and is finished unsuccessfully.

11.4 TETRA security related procedures

The SIM security procedures are associated with the air interface message exchange protocol procedures for authenticating the SIM to a TETRA network and the TETRA network to the SIM. During these SIM security procedures the card runs the specified algorithms TA11/12 and TA21/22 to calculate respectively the expected response from the SIM, (X)RES1 with its associated derived cipher key DCK1 and the expected response from the SwMI, (X)RES2 with its associated derived cipher key DCK2.

On successful authentication the derived cipher key DCK, used for encrypting air interface signalling and traffic channels, shall be derived from its two parts DCK1 and DCK2 by running the TB4 algorithm.

None of the algorithms shall not be executable unless DF_{TETRA} has been selected as the Current Directory and a successful CHV verification procedure has been performed (see 1120).

The procedures are either initiated by the ME (internal applications or MMI) or interfaced from the SwMI via the ME. In the latter case the ME provides only a delivery service with no other functionality than to interpret the PDUs if necessary.

11.4.1 Authentication procedures and generation of DCK

11.4.1.1 Mutual authentication requirement request

The SIM performs the read procedure with EF_{SEC} to determine whether a mutual authentication is requested by the SIM in case of a SIM authentication request from the SwMI.

11.4.1.2 SIM authentication

The ME runs the TA11/12 ALGORITHM, followed by a GET RESPONSE to obtain the RES1. If and only if the SIM requests a mutual authentication (see subclause 11.4.1.1), the ME runs then the GET RANDOM, followed by the TA21/22 ALGORITHM. If the authentication was successful, it finally runs the TB4 ALGORITHM to obtain DCK.

11.4.1.3 SwMI authentication

The ME runs the GET RANDOM function, followed by the TA21/22 ALGORITHM. If and only if the SwMI requests a mutual authentication, the ME runs the TA11/12 ALGORITHM, followed by a GET RESPONSE to obtain the RES1. If the authentication was successful, it finally runs the TB4 ALGORITHM to obtain DCK.

11.4.2 TETRA OTAR key computation (CCK, GCK, SCK)

The CCK, GCK and SCK cipher keys can be updated by OTAR. They are sent over the air interface in sealed format and need to be unsealed on receipt by algorithms on the SIM.

SCK and CCK are accessible from the SIM-ME interface but GCK is accessible only in modified format (MGCK).

11.4.2.1 CCK distribution

On receipt of a new SCCK from the SwMI, the ME checks the validity of the CCK-ID then runs the TA32 ALGORITHM to update EF_{CCK}. The record to be updated in EF_{CCK} is identified as follows: The ME checks whether the CCK-ID being broadcast by the SwMI is identical to the CCK-ID stored in record 1 of EF_{CCK}. If no, record 1 is updated; otherwise, record 2 is updated.

11.4.2.2 CCK changeover

When the ME detects a new CCK-ID in use it determines the record number in EF_{CCK} which contains the new CCK-ID. After verifying that the new CCK-ID is valid, the ME runs the TA71 ALGORITHM to update all records in EF_{MGCK} using the CCK record in EF_{CCK} identified by the CCK-ID.

11.4.2.3 GCK distribution

The ME analyses EF_{GSSIS} and EF_{GSSID} to locate the required GTSI. If the GTSI is not already present, the ME allocates a free record number in the EF_{GSSID} and there places the new GTSI.

The ME checks whether there is a GCK (and MGCK) associated with the GTSI by investigating the appropriate GCK record number data element in EF_{GRDS} or EF_{GRDD}. If there is no such associated GCK, then a free record in EF_{GCK} is allocated (see note below), and the corresponding target record number in EF_{GRDS} or EF_{GRDD} is updated accordingly.

In the case where there was already a GCK (and MGCK) present, the ME identifies whether the new GCK-VN is valid by comparing it to the GCK-VN being stored currently in the appropriate record of EF_{MGCK}. If it is not valid the procedure is aborted.

The ME then runs the TA82 ALGORITHM to update the respective GCK. After this, the ME runs the TA71 ALGORITHM on this particular GCK to obtain the corresponding MGCK. For this operation, the current CCK (the one being indicated on the broadcast channel) is used.

NOTE: To allocate a free record in EF_{GCK} the ME reads EF_{GRDS} and EF_{GRDD} and works out if there is a record in EF_{GCK} which is not presently pointed to by any GCK record pointer.

11.4.2.4 SCK distribution

On receipt of a new SSCK from the SwMI, the ME identifies whether the new SCK-VN is valid by comparing it to the one being stored currently. If it is not valid the procedure is aborted. Then the ME runs the TA41/52 ALGORITHM in order to unseal the SCK and store it in that record of EF_{SCK} which is indicated by the SCKN.

11.4.3 ITSI request

The ME performs the reading procedure with EF_{ITSI}.

11.4.4 ITSI disabling/re-enabling

Permanent disabling:

On receiving the ITSI permanent disable command the ME runs the SwMI authentication procedure defined in 11302. If the SwMI is successfully authenticated then the invalidate procedure is performed on $EF_{ITSI.}$ The TETRA session is immediately terminated. (see note)

Temporary disabling:

On receiving the ITSI temporary disable command the ME runs the SwMI authentication procedure defined in 11302. If the SwMI is successfully authenticated then the ME performs the update procedure with $EF_{ITSIDIS}$ to set the flag to "temporarily disabled". (see note)

Re-enabling:

On receiving the ITSI enable command the ME runs the SwMI authentication procedure defined in 11302. If the SwMI is successfully authenticated then the updating procedure is performed on $EF_{ITSIDIS}$ to set the flag to "not disabled".

NOTE: It is an implementation issue for the SIM to deny access to further sensitive EFs (such as group identities and air interface encryption keys) if the ITSI is temporarily or permanently disabled.

11.5 Subscription related procedures

11.5.1 Username request

Requirement: Service no.16 "available".

Request: The ME performs the reading procedure with EF_{UNAMF}.

Update: The ME performs the updating procedure with EF_{UNAMF}.

11.5.2 ITSI temporarily disabled enquiry

Request: The ME performs the reading procedure with $EF_{ITSIDIS}$.

Update: The ME performs the updating procedure with $EF_{ITSIDIS}$.

11.5.3 Subscriber class request

Request: The ME performs the reading procedure with EF_{SCT}.

Update: The ME performs the updating procedure with EF_{SCT}.

11.5.4 Location information

Request: The ME performs the reading procedure with EF_{LOCI}.

Update: The ME performs the updating procedure with EF_{LOCI}.

This procedure/file uses also the network table EF_{NWT}.

11.5.5 Group identity information

The following procedures apply to both static (EF_{GSSIS}) and dynamic (EF_{GSSID}) groups with the exceptions mentioned in the following paragraphs.

Request: The ME performs the reading procedure with EF_{GSSIS} or EF_{GSSID}.

Update: The ME performs the updating procedure with EF_{GSSID}.

Erasure: The ME identifies the record in EF_{GSSID} containing the GSSID to be erased and marks it as free.

The update and erasure of EF_{GSSID} requires the updating of the network table. The handling procedures of the network table (EF_{NWT}) are defined under subclause 11.6.

11.5.6 Group related data

The following procedures apply to both static and dynamic group related data (EF_{GRDS} and EF_{GRDD}).

Request: The ME performs the reading procedure with EF_{GRDS} or EF_{GRDD}

Update: The ME performs the updating procedure with EF_{GRDS} or EF_{GRDD}

NOTE: A record in EF_{GRDX} is free when the associated record in EF_{GSSIX} is marked free.

11.5.7 User's group information

Request: The ME performs the reading procedure with EF_{GINFO}

Update: The ME performs the updating procedure with EF_{GINFO}.
The update of the file is performed in the beginning of a group call.

The update of this file requires the updating of the network table. The handling procedures of the network table (EF_{NWT}) is defined under subclause 11.6.

11.5.8 Call modifiers

Requirement: Service no.26 "available".

Request: The ME performs the reading procedure with EF_{CMT}

Update: The ME performs the updating procedure with EF_{CMT}.

11.5.9 Service Provider Name

Requirement: Service no.14 "available".

Request: The ME performs the reading procedure with EF_{SPN}.

11.5.10 DMO channel procedures

Requirement: Service no.27 "available".

Request: The ME performs the reading procedure with EF_{DMOCh}

Update: The ME performs the updating procedure with EF_{DMOCh}.

Erasure: The ME erases the contents of the record in EF_{DMOCh} by filling the record with 'FF'.

11.5.11 Emergency addresses

Request: The ME performs the reading procedure with EF_{EADDR}

Update: The ME performs the updating procedure with EF_{EADDR}.

Erasure: The ME erases the contents of the record in EF_{EADDR} by filling the record with 'FF'.

11.5.12 Interrupted emergency call request

Request: The ME performs the reading procedure with EF_{EINFO}.

Update: The ME performs the update procedure with EF_{EINFO}

NOTE: If an emergency call was in progress when the ME was powered down the current emergency call record number, if non-zero, indicates that an emergency call procedure was in progress when the ME was powered down. The ME should recognize the non-zero value as an indication to take action as necessary to restart the emergency call after authentication.

11.6 Network related procedures

Request: The ME performs the reading procedure with EF_{NWT}.

Update: The ME checks whether the network address to be stored is already present. If so, the record pointer counter of the found network address record is increased by one.

If the address is not found on the network table, a new record is added to the network table and the corresponding record pointer counter is set to one.

Erasure: The ME checks the record pointer counter of the network address to be deleted. If the value of the counter is 2 or more, the counter is decreased by one. If the Record pointer counter is 1, the record on the network table is deleted (indicated as free by filling it with 'FF's)

11.6.1 Forbidden networks

Request: The ME performs the reading procedure with EF_{FORBID}.

Update: The ME performs the updating procedure with EF_{FORBID}.

Erasure: The ME can erase the whole contents of the Forbidden networks. The action can either be initiated by the ME or the MMI. In case of erasure, the whole table of Forbidden addresses will be erased i.e. marked free by filling them with 'FF's.

11.6.2 Preferred networks

Requirement: Service no.15 "available".

Request: The ME performs the reading procedure with EF_{PREF}.

Update: The ME performs the updating procedure with EF_{PREF}.

11.7 Phonebook related procedures

11.7.1 Dialling numbers

The following procedures may be applied to EF_{ADN} and its associated extension file EF_{EXT1} as described in the procedures below, and also to EF_{EDN}, EF_{LDN}, EF_{SDN}, $EF_{ADNTETRA}$, $EF_{FDNTETRA}$, $EF_{LDNTETRA}$ and $EF_{SDNTETRA}$ and their associated extension files. If these files are not allocated and activated, as denoted in the SIM service table, the current procedure shall be aborted and the appropriate EFs shall remain unchanged.

As an example, the following procedures are described as applied to ADN.

Requirement: Service no.3 "available".
(Service no.2 for ADNTETRA;
Service no.4 for FDNTETRA;
Service no.5 for FDN;
Service no.6 for SDNTETRA;
Service no.7 for SDN;
Service no.8 for LNDTETRA;
Service no.9 for LND.)

Request: The ME sends the identification of the information to be read. The ME shall analyse the data of EF_{ADN} (see subclause 10.3.26) to ascertain whether additional data is associated in EF_{EXT1}. If necessary, the ME performs the reading procedure on EF_{EXT1} and EF_{GWT} to assemble the complete ADN.

Update: The ME analyses and assembles the information to be stored as follows:

i) the ME identifies the record containing the Name to be updated;

ii) the dialling number (and/or Supplementary service access string in case of ADNTETRA) shall be allocated to the bytes of the EF as follows:

- If the dialling number contains 16 or less "digits", it shall be stored in "PSTN or PABX number".

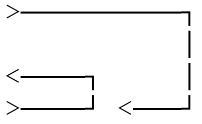
- If the dialling number contains more than 16 "digits", the procedure shall be as follows:

  The ME seeks for a free record in EF_{EXT1}. If no Extension1 record is marked as "free", the procedure is aborted.

  When a free Extension1 record is found, the first 16 "digits" are stored in the "PSTN or PABX number". The value of the "Length of PSTN or PABX number contents" is set to the maximum value, which is 16. The Extension1 record number in EF_{ADN} is coded with the associated record number in the EF_{EXT1}. The remaining digits are stored in the selected Extension1 record. The first byte of the Extension1 record is set with the number of digits of the remaining data. Further extension records can be added up to the full length of the dialling string by chaining records in Extension1. The total number of digits is the sum of the "Length of PSTN or PABX number contents" of EF_{ADN} and byte 1 of all associated chained Extension1 records containing data;

Example of a chain of extension records being associated to an ADN or LND. The Extension1 record number of ADN or LND is set to 3.

No of Record	Extension Data	Next	Record
.	.	.	
.	.	.	
Record 3	xx xx	'06'	
Record 4	xx xx	'xx'	
Record 5	xx xx	'FF'	
Record 6	xx xx	'05'	
.	.	.	
.	.	.	



iii) the ME seeks the gateway address in EF_{GWT}. If it is not already in the table a new entry is created. If a new entry can not be created, the procedure is aborted. When the entry is available the ME updates the Gateway address record number in EF_{ADN} to the associated record in EF_{GWT};

iv) the ME chooses a proper call modifier in EF_{CMT}.

When i), ii), iii) and iv) have been successfully executed the ME performs the updating procedure with EF_{ADN}.

NOTE: If the SIM does not have available empty space to store the received ADN, or if the procedure has been aborted, the ME advises the user.

Erasure: The ME sends the identification of the information to be erased. The content of the identified record in EF_{ADN} is marked as "free". Furthermore, the associated records in EF_{GWT} and EF_{EXT1} are updated accordingly.

11.7.2 FDN specific procedures

Requirement: Service no. 5 "available"

If FDN is enabled (i.e. EF_{ADN} is invalidated or not present) the ME shall operate in a restricted mode where only those phone numbers contained in EF_{FDN} are used.

If FDN-TETRA is enabled (i.e. $EF_{ADNTETRA}$ is invalidated or not present) the ME shall operate in a restricted mode where only those phone numbers contained in $EF_{FDNTETRA}$ are used.

Both modes FDN and FDN-TETRA can be enabled independently of each other.

ADN and FDN are mutually exclusive of each other and independent of the state of ADNTETRA and FDNTETRA. Likewise, ADNTETRA and FDNTETRA are mutually exclusive of each other and independent of the state of ADN and FDN. This means that there may be restricted ADN phonebook operation or restricted TETRA phonebook operation and these are independent of each other.

The following three procedures are only applicable to service no.4 (FDNTETRA) and no.5 (FDN). As an example, the following procedures are described as applied to FDN.

11.7.2.1 FDN capability request

To ascertain the state of FDN, the ME checks in EF_{SST} whether or not ADN is activated. If ADN is not activated, service no.5 is enabled. If ADN is activated, the ME checks the response data of EF_{ADN}. If EF_{ADN} is invalidated , service no.5 is enabled. In all other cases service no.5 is disabled.

11.7.2.2 FDN disabling

The FDN disabling procedure requires that CHV2 verification procedure has been performed successfully and that ADN is activated. If not, FDN disabling procedure will not be executed successfully. To disable FDN capability, the ME rehabilitates EF_{ADN}. The invalidate/rehabilitate flag of EF_{ADN}, which is set by the REHABILITATE command, is at the same time the indicator for the state of the service no.5. If ADN is not activated, disabling of FDN is not possible and thus service no.5 is always enabled (see FDN capability request).

11.7.2.3 FDN enabling

The FDN enabling procedure requires that CHV2 verification procedure has been performed successfully. If not, FDN enabling procedure will not be executed successfully. To enable FDN capability, the ME invalidates EF_{ADN}. The invalidate/rehabilitate flag of EF_{ADN}, which is set by the INVALIDATE command, is at the same time the indicator for the state of the service no.5 (see FDN capability request). If ADN is not activated, service no.5 is always enabled.

Invalidated ADNs may optionally still be readable and updatable depending on the file status (see subclause 9.4).

11.8 Status and short data message procedures

11.8.1 Display of status message texts

Requirement: Service no.22 "available".

Request: The SIM selects EF_{STXT} and seeks for the identified status message value. If the message value is found it performs the reading procedure with EF_{STXT}.

11.8.2 Display of SDS1 message texts

Requirement: Service no.23 "available".

Request: The SIM selects EF_{MSGTXT} and seeks for the identified status message value. If the message value is found it performs the reading procedure with EF_{MSGTXT}.

11.8.3 Storage of status and SDS messages types 1, 2 and 3

Requirement: Service no.24 "available".

Request: The SIM selects EF_{SDS123} and seeks for the identified status or SDS message. If this message is found, the ME performs the reading procedure with EF_{SDS123}

Update: The ME looks for the next available area to store the status or SDS message in EFsds123. If such an area is available, it performs the updating procedure with EF_{SDS123}.

If there is no available empty space in the SIM to store the received short message, a specific MMI takes place in order not to lose the message.

Erasure: The ME selects EF_{SDS123} and identifies the records to be erased. Then it performs the update procedure to mark them as free.

NOTE: Depending on the MMI, the message may be read before the record is marked as "free". After performing the updating procedure with EF_{SDS123}, the memory allocated to this short message in the SIM is made available for a new incoming message. The memory of the SIM may still contain the old message until a new message is stored in that area.

11.8.4 Storage of SDS messages type 4

Requirement: Service no.25 "available".

Request: The SIM selects EF_{SDS4} and seeks for the identified short message . If this message is found, the ME performs the reading procedure.

Update: The ME looks for the next available area to store the short message in EF_{SDS4}. If such an area is available, it performs the updating procedure with EF_{SDS4}.

If there is no available empty space in the SIM to store the received short message, a specific MMI procedure takes place in order not to lose the message.

Erasure: The ME selects EF_{SDS4} and identifies the records to be erased. Then it performs the update procedure to mark them as free.

NOTE: Depending on the MMI, the message may be read before the record is marked as "free". After performing the updating procedure with EF_{SDS123}, the memory allocated to this short message in the SIM is made available for a new incoming message. The memory of the SIM may still contain the old message until a new message is stored in that area.

Annex A (normative): Plug-in SIM

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

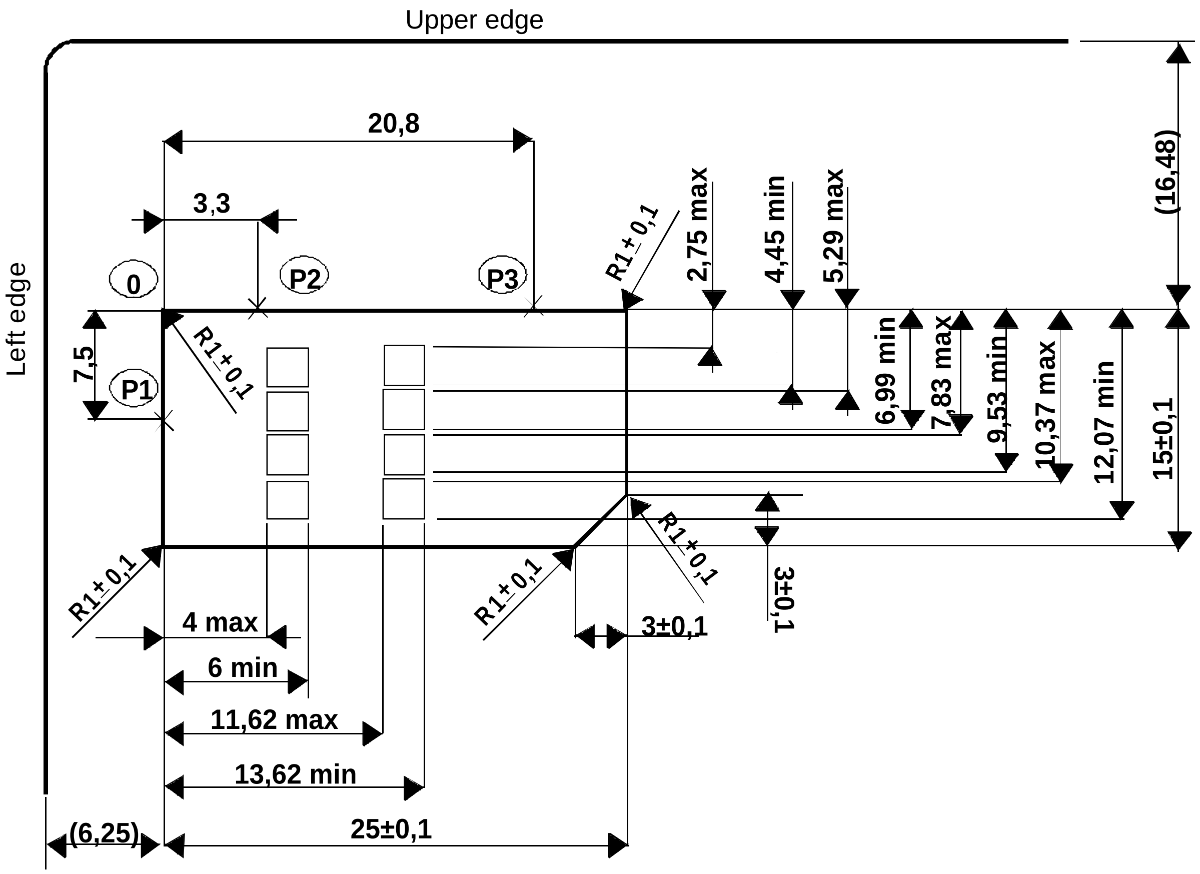


Figure A.1: Plug-in SIM

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

Annex B (informative): FDN Procedures

The FDN facility allows operation of the TETRA terminal in a restricted state whereby it can only initiate calls to a pre-determined destination or list of destinations.

A TETRA SIM may be personalized so that the terminal can be operated in only the restricted state, only the unrestricted state or to allow the operation mode to be switched between states through the MMI.

FDN services

Two FDN services are provided for the TETRA SIM. Service number 4 allows fixed dialling to other TETRA addresses while service number 5 allows fixed dialling to destinations on a PABX or the PSTN. These services may be individually or jointly enabled as indicated in the SIM service table.

The SIM service table provides an enable/disable indicator for each of the two FDN services to indicate to the ME the capabilities of the SIM. Where the SIM service table indicates that the SIM is capable of both ADN and FDN services, the operating state can be switched as described below.

FDN operation

When the ME is operating in the restricted FDN state, the user may only call destinations listed in the FDN directories EF_{FDN} (service no 5) and/or $EF_{FDNTETRA}$ (service no 4). Attempts to call other destinations shall be rejected by the ME, other than those initiated by activation of the emergency call procedures.

FDN initialization

When a TETRA session is initialized, the ME should check the SIM service table for the state of the FDN services. If neither service is enabled, the ME should enter the unrestricted operation state, offering facilities as otherwise indicated in the SIM service table.

If either of the FDN services are enabled in the SIM service table, the ME should further check the entries for ADN (service no 2) and ADNTETRA (service no 3). If neither ADN service is enabled the ME should enter the restricted FDN operation state.

If both ADN and FDN services are enabled in the SIM service table, the operation mode may be determined by the validity of EF_{ADN}. If EF_{ADN} is invalidated, the ME should enter the restricted FDN operation state. If EF_{ADN} is not invalidated, the ME should enter the unrestricted state.

Change of FDN operation mode.

Where the SIM Service Table indicates that a SIM supports both FDN and unrestricted modes of operation, the validity of the file EF_{ADN} provides the indicator as to the current operating state as described above.

The ME may provide an MMI operation to allow toggling of the operation state by performing invalidation or rehabilitation of EF_{ADN}. This procedure can only be performed after successful completion of the CHV2 verification procedure to satisfy the access rights for EF_{ADN}.

Change of FDN access details

The ME may provide a method on the MMI to change entries in the FDN directories, thereby changing the list of call destination when the ME is operating in the restricted state. This procedure can only be performed after successful completion of the CHV2 verification procedure to satisfy the access rights for update to EF_{FDN}.

Annex C (informative): Suggested contents of EFs at pre-personalization

If EFs have an unassigned value, it may not be clear from the main text what this value should be. This annex suggests values in these cases.

File Identification	Description	Value
EF_{ICCID}	Card identification	operator dependant (see subclause 10.2.1)
EF_{DIR}	Application directory	
EF_{ITSI}	ITSI	operator dependant (see 10.3.2)
$EF_{ITSIDIS}$	ITSI Disabled	'00'
EF_{UNAME}	Username	'FF...FF'
EF_{SCT}	Subscriber class table	operator dependant (see 10.3.5)
EF_{PHASE}	Phase identification	'00'
EF_{SCK}	Static Cipher Key	operator dependant (see 10.3.9)
EF_{GSSIS}	Pre-programmed GSSIs	operator dependant (see 10.2.10)
EF_{GSSID}	Dynamic GSSIs	'FF...FF00'
EF_{GCK}	Group Cipher Keys	operator dependant (see subclause 10.2.14)
EF_{GINFO}	User's group information	'00FF...FF'
EF_{SST}	SIM Service Table	'00...00'
EF_{CMT}	Call modifier table	'FF...FF'
EF_{FORBID}	Forbidden networks table	'FF...FF'
EF_{PREF}	Preferred networks table	'FF...FF'
EF_{SPN}	Service provider name	'00FF...FF
EF_{LP}	Language preference	'FF'
EF_{LOCI}	Location information	'FF...FF'
EF_{NWT}	Network table	'FF...FF'
EF_{ADN}	Abbreviated dialling numbers	'FF...FF'
EF_{EXT1}	Extension 1	'FF...FF'
$EF_{ADNTETRA}$	Abbreviated dialling numbers for TETRA network	'FF...FF'
EF_{EXTA}	Extension A	'FF...FF'
EF_{FDN}	Fixed dialling numbers	'FF...FF'
EF_{EXT2}	Extension 2	'FF...FF'
$EF_{FDNTETRA}$	Fixed dialling numbers for TETRA network	'FF...FF'
EF_{LND}	Last number dialled	'FF...FF'
$EF_{LNDTETRA}$	Last number dialled for TETRA network	'FF...FF'
EF_{SDN}	Service dialling numbers	'FF...FF'
EF_{EXT3}	Extension 3	'FF...FF'
$EF_{SDNTETRA}$	Service dialling numbers for TETRA network	'FF...FF'
$EF_{SDNPABX}$	Service dialling numbers for PABX	'FF...FF'
EF_{MSGTXT}	SDS message texts	'FF...FF'
EF_{STXT}	Status message texts	Operator dependent
EF_{SDS4}	SDS type 4 message storage	'00FF...FF'
EF_{MSGEXT}	Message Extension	'FF...FF'
EF_{EADDR}	Emergency address	'FF...FF'
EF_{EINFO}	Emergency call information	'00'

Annex D (normative): Database structure for group IDs and phone books

Use of the network table

Relational database mechanisms are used to save a significant amount of memory. Several EFs (e.g. EF_{GSSIS} and EF_{GSSID}) refer to the Network table for network address instead of saving it with each group short subscriber identity. However, since a network address can be referenced from more than one place, a record pointer counter is needed to keep track of how many times a network address is referenced. When the record pointer counter of a network address is one, it is referenced from only one place. When that address is removed, the corresponding network address can be removed also, since it was the only one using it. This housekeeping method is used to remove unnecessary network addresses from the network table.

The network table is thus handled using the following procedures:

> When a network address needs to be stored with a record, the network table (EF_{NWT} see subclause 10.3.23) needs to be read. If the address (MCC and MNC) is already found on the network table, the Record pointer counter of the found network address record needs to be increased by one. Only the record number of the network address on the network table is stored with the record that needs the network address.
>
> If the address is not found on the network table, a new record needs to be added to the network table. On the network table the new network address (MCC and MNC) is stored along with a record pointer counter, which is set to one. Only the record number of the network address on the network table is stored with the record that needs the network address.
>
> If the desired network address is not found in the network table, and it cannot be added because of the file being full, the new network address cannot be stored on the SIM.
>
> If a record that uses a network address in the network table needs to be deleted, the network table also needs to be updated. The record that needs to be updated can be found using the record number. The record number is stored with the record that is to be deleted. When the record in the network table is found, the record pointer counter is read. If the value of the counter is 2 or higher, the counter is decreased by one and the record that referenced it can be deleted.
>
> If the record pointer counter is 1, the whole record on the network table can be deleted (indicated as free by filling it with 'FF's) along with the record that pointed to that record.

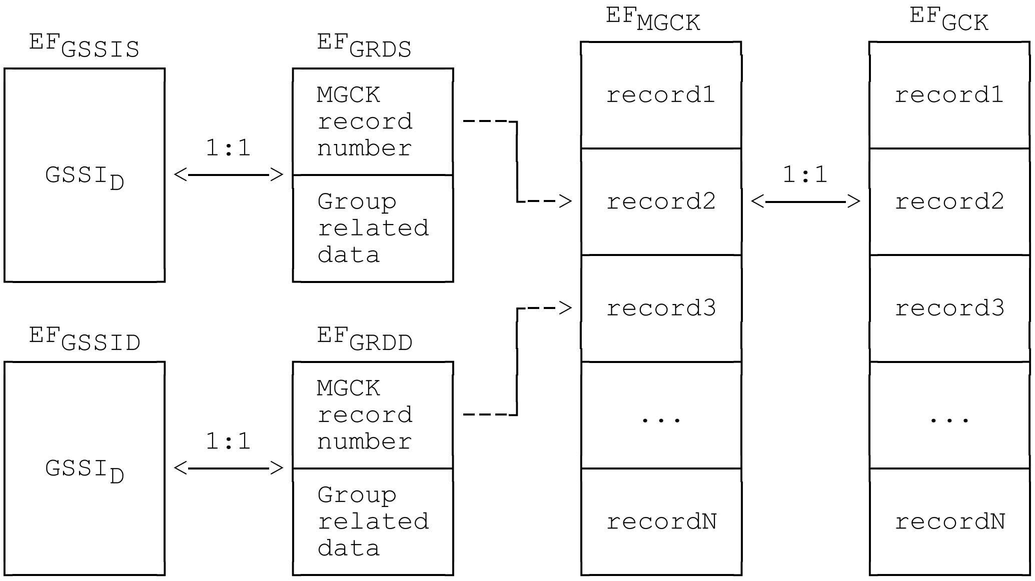


Figure D.1: Graphical presentation of group data related EF structures

Figure D.2 shows how records in phonebook related EFs can point to records in other phonebook related EFs.

NOTE: Each of the 8 phonebooks (ADN, LND, FDN, SDN, ADNTETRA, LNDTETRA, FDNTETRA, SDNTETRA) may point to EF_{CMT}, which is not shown on the diagram.

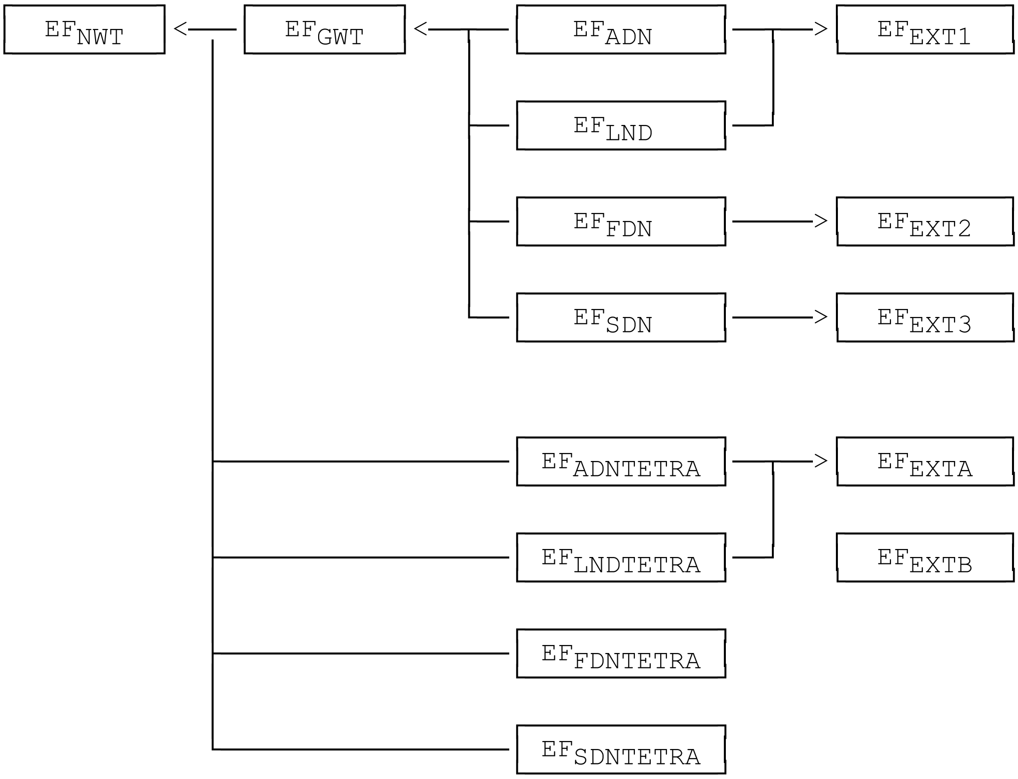


Figure D.2: Graphical presentation of phonebook related EF structures

Annex E (informative): Emergency call facilities and procedures.

The TETRA standards provide a wide variety of call types and facilities which may be used in an emergency situation. The activation of an emergency facility is implementation-specific and so the file content defined for the TETRA SIM card is intended to offer flexibility in handling emergency situations. This annex offers further explanation of the information available to the ME in handling an emergency situation.

Emergency call control

The EF_{EINFO} contains a control flag to indicate to the whether or not emergency calls are enabled for this particular card.

Emergency call addresses

The EF_{EADDR} contains a list of call destinations for use in an emergency call. Entries in the file can require that the call be placed to either the last group in which the ME took part or to a pre-defined destination. When the file contains more than one address, it is suggested that the order of the records in the file should indicate the order of preference for the call, starting with the highest preference.

Each record in EF_{EADDR} also contains a number of flags providing an indication as to the type of the call address, allowing a mix of call types to be indicated. The call type can be one of a selection of 10 variants, including all of the common speech calls and short data transactions. For circuit mode calls, a data field indicates the nature of the required call i.e. individual, group, acknowledged group or broadcast.

When the emergency call type is a status or short data transaction, an additional option is selected by a flag which may be used to indicate a preference as to the source of the data to be transferred in an emergency message. When the pre-defined value stored in the card is selected, a record number pointer indicates EF_{SDS123} or EF_{SDS4} which contain both the destination and message content. When the "application" source is selected, it is suggested that the contents of the data field would be obtained by an application running in the ME.

Protection for interrupted emergency calls

The EF EF_{EINFO} contains a flag indicating the action to be taken on power-on after an interrupted emergency call - to optionally resume the emergency call without further operator intervention.

Where EF_{EINFO} indicates that an interrupted emergency call should be continued next time the ME is powered up, the ME should maintain the current emergency call index in EF_{EINFO} during any emergency call procedure. In particular, the index should be set by the ME to a value to be understood by the restarting ME as the call is initiated and zeroed on normal termination. The index allows the restarting ME to establish that an emergency transaction was in progress and, from the index, which of the available call options to restart. The coding of the index is implementation-dependant but is dimensioned so that it can be used as a pointer to a record number in EF_{EADDR} if required.

Successful connection of an emergency call

It is suggested above that the ME should attempt to set up the emergency call to each of the destinations prescribed in EF_{EADDR} until a successful connection is achieved.

It should, however, be noted that not all call types provide a definite indication of success. An unacknowledged group call, for example, may succeed in establishing a 'call' but it is possible that no other member of the group could be available and so the result would be no exchange of useful information. For PABX or PSTN voice calls, call routing beyond the TETRA infrastructure may not be able to return a definite indication of a successful exchange to the originating terminal and so a call to an unanswered or engaged number could result. The implementation of the emergency facility may take account of this possibility in controlling the emergency call.

Emergency calls in Direct Mode

When an emergency call record in EF_{EADDR} requires the use of direct mode, the implementation may handle the possibility of the required party being on one of a multiplicity of DMO channels. The record in EF_{EADDR} includes a field to indicate a channel number explicitly. It is suggested that a zero channel number could cause the ME to use the flags provided in EF_{DMOCh} which designate a channel for emergency use in attempting to set up the call.

Emergency calls when the SIM card is not fitted

Where the ME is not equipped with a SIM interface, or the SIM is absent, it must still be possible, for some applications, to make an emergency call. This can be achieved using the Virtual SIM concept to duplicate the operation of the emergency procedures and files.

Whilst the actual data storage and access mechanisms are not standardized for a Virtual SIM implementation, the facilities provided in the ME could usefully provide corresponding data elements to allow the same operational procedure to be effective. In particular, the file EF_{EINFO} includes the flag enabling the emergency call facility.

For a virtual SIM implementation to be able to operate on the TETRA network when the SIM is not present, the ME will need to store many of the data elements defined as mandatory within this ETS. If, however, only emergency calls are to be permitted in the no-SIM condition, the ME will need a minimum data set to enable authentication to the network and knowledge of the intended destination(s) of the emergency call(s).

Annex F (informative): Bibliography

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

- EN 726-4: "Terminal Equipment (TE); Requirements for IC cards and terminals for telecommunication use - Part 4: Application independent card related terminal requirements".

History

Document history			
February 1998	Public Enquiry	PE 9826:	1998-02-27 to 1998-06-26
September 1998	Vote	V 9847:	1998-09-22 to 1998-11-20
November 1998	First Edition		

ISBN 2-7437-2712-8
Dépôt légal : Novembre 1998