3GPP2 C.S0023-D

Version 2.05

July, 2013

3RD GENERATION PARTNERSHIP PROJECT 2 "3GPP2"

Removable User Identity Module for Spread Spectrum Systems

Index to Changes Accepted for Inclusion in C.S0023-D v3.0

§	Name	Changes	Source	Ver.
1	**General**			
1.4	**Terms**	Add ECC	20130311-011R2	2.02
2	**Physical, Electrical and Logical Interfaces**			
2.7	**Content of EFs**	• Reword the sentence to be intelligible	Editor	2.01
3				
3.4	**Coding of EFs**	• Section 1.2 is now 1.5	Editor	2.01
3.4.18	**EF$_{CST}$**	• Add new MO SMS Control service	20130311-036	2.02
3.4.20	**EF$_{OTAPASPC}$**	• Missing underscore in SPC_Change_Enable	Editor	2.01
		• Insert name of field “SPC_Change_Enable” to remove ambiguity	Editor	2.02
3.4.31	**EF$_{SPN}$**	• Add a description of how 7 bit coding should be used.	201307011-004r1	2.05
3.4.32	**EF$_{USGIND}$**	• Remove first two lines of coding, which are confusing relative to the byte description	20130709-004	2.05
3.4.37	**EF$_{ECC}$**	• “B2” should be “b2”	Editor	2.01
		• Add acronym “ECC” • Add sentence indicating that dialing an ECC means that the ME shall treat the call as an emergency call.	20130311-011R2	2.02
3.4.38	**EF$_{ME3GPDOPC}$**	• Acronym should follow EF name, not the name of another EF. Use of ‘mode’ in text should be replaced by ‘capability’ as there is also a 3GPD operating mode EF.	Editor	2.01
3.4.39	**EF$_{3GPDOPM}$**	• Note that the position of the two bit field differs from the referenced specification (C.S0016).	Editor	2.01
3.4.51	**EF$_{MECRP}$**	• Fix formatting problems in tables (note that change is invisible).	Editor	2.01
3.4.60	**EF$_{BCSMSP}$**	• “B2” should be “b2”	Editor	2.01
3.4.67	**EF$_{MMSN}$**	• Add space before EF table. • “Reserved for future use” -> “RFU” (two times) • Change table borders to get Octet 3 definitions on one line.	Editor	2.01
3.4.73	**EF$_{3GCIK}$**	• Note that this is just for storing the CK/IK produced by the “3G ACCESS AKA” command	20121015-006	2.01
		• Space between introductory paragraph and table.	Editor	2.02
3.4.77	**EF$_{CDMACNL}$**	• “B6” should be “b6” (twice)	Editor	2.01
3.4.84	**EF$_{AppLabels}$**	• Add a description of how 7 bit coding should be used.	201307011-004r1	2.05
3.4.89	**EF$_{3GPDUPPExt}$**	• Table headings bold, not underlined • Add “keep with next” to some paragraphs.	Editor	2.01
3.4.91	**EF$_{IPV6CAP}$**	• Put border on IPv6 flags table	Editor	2.01
		• Change “Reserved for Future Use” to “RFU”	Editor	2.02
3.5	**Coding of Packet Data Security-Related Parameters**	• Use SW1/2 codes of 69/82 to indicate Secure Mode Not Active	20121015-015r2	2.01
4				
4.2.1	**Managing Shared Secret Data**	• Restore damaged Figures 2-6 (not change marked)	Editor	2.02

4.2.2	**Performing Authentication Calculations and Generating Encryption Keys**	• Replace reference to generic "Get Response" command by "response to Run CAVE"	Editor	2.01
4.3.2.15	**3GPD Configuration Request/Response Messages**	• Change already made	20121015-015r2	2.01
4.3.2.16	**3GPD Download Request/Response Messages**	• Change already made	20121015-015r2	2.01
4.4.4	**AUTHENTICATE**	• Add sentence to both P1='01' and '02' that IK, CK and UAK, if calculated, are stored in temporary memory until the CONFIRM_KEYS command is received. • Change title of P1='02' to align with subcommand name.	20121015-006	2.01
4.4.4.1	**Advisory note on the use of RUN CAVE**	• Replace reference to generic "Get Response" by reference to response to Run CAVE	Editor	2.01
4.4.4.2	**Use of Cipher Key Generation Command**	• Replace reference to "Get Response" by reference to response to GENERATE KEY/VPM	Editor	2.01
4.5.12	**3GPD Configuration Request**	• Explain interaction of Result Code and Status Words. • Suggested new cross reference has already been added.	20121015-015r2	2.01
4.5.13	**3GPD Download Request**	• Explain interaction of Result Code and Status Words • Suggested new cross reference has already been added.	20121015-015r2	2.01
4.7	**Description of Packet Data Security-Related Functions**	• Replace damaged figures 7-10. • Center all figures	Editor	2.02
4.7.3	**Performing Mobile IP Authentication**	• Redraw Figure 9 • Change "UIM" to "R-UIM"	Editor	2.04
4.8.1	**COMPUTE IP AUTHENTICATION**	• Change unspecified value for P2 to '00'	Editor	2.01
		• Back out this change as P2 is not always '00'	Editor	2.02
4.11	**Description of AKA-related functions**	• Redraw Figure 11 to fix arrow with wrong orientation	Editor	2.01
		• Fix placement of "No" arrows in decision boxes in Figure 11	Editor	2.04
4.11.5	**Restoration of 3G Keys**	• Replace paragraph with clarifications notably to indicate that CK/IK restoral just relates to the circuit switched value (3G Access AKA AUTHENTICATE).	20121015-006	2.01
4.11.6	**CONFIRM_KEYS Command Description**	• Note that this applies to 3G Access AKA and EAP AKA AUTHENTICATE commands • Only if previous command was 3G Access AKA should the values be stored in EF_{3GCIK}	20121015-006	2.01
		• Memory for keys is semi-permanent	Editor	2.02

Revision History

Revision	**Description**	**Date**
C.S0023-0 v2.0	Initial Release, Version 2	July 2000
C.S0023-0 v4.0	Initial Release, Version 4	June 2001
C.S0023-A v1.0	Release A	September 2002
C.S0023-A v2.0	Release A Version 2	February 2004
C.S0023-B v1.0	Release B	May 2004
C.S0023-A v3.0	Release A Version 3	January 2005
C.S0023-C v1.0	Release C	June 2006
C.S0023-C v2.0	Release C Version 2	October 2008
C.S0023-D v1.0	Release D	June 2009
C.S0023-D v2.0	Release D Version 2	December 2011
C.S0023-D v3.0	Release D Version 3	<month> 2013

CONTENTS

FIGURES

TABLES

FOREWORD

(This foreword is not part of this specification)

This document contains the requirements for the Removable User Identity Module (R-UIM). It is an extension of Subscriber Identity Module (SIM), per latest [17][1] capabilities, to enable operation in a [5][14][15] radiotelephone environment. Examples of this environment include, but are not limited to, analog, [14]-based CDMA and the [1-5] family of standards.

These requirements are expressed as additions to the current specification of the SIM. The composite R-UIM is comprised of the current SIM specification and this ancillary or "delta" document. The SIM specification is included as a reference. It is intended that all upgrades to the SIM specification will also apply to the R-UIM.

The current SIM specifications (see references) address the physical and electrical characteristics of the removable module, along with the user-to-card interface and terminal-to-card signaling protocol. Operation in a [5][14][15] environment requires that additional commands and responses be developed within the context of this document. This document also defines new Elementary Files (EFs) for storage of parameters that are added for operation in a [5][14][15] environment.

This standard specifies security-related procedures and commands, along with data and information storage items that permit basic operation in the [5][14][15] environment. Later versions are expected to also address the delivery of [5][14][15] user features and services via the R-UIM.

[1] [ ] indicates the corresponding document to be cross referenced.

1 GENERAL

1.1 Scope

This document contains the requirements for use of a Removable User Identity Module (R-UIM) card in a cdma2000®[2] wireless communications device operating in a [5][14][15] radiotelephone environment.

1.2 Requirements Language

"Shall" and "shall not" identify requirements to be followed strictly to conform to this document and from which no deviation is permitted. "Should" and "should not" indicate that one of several possibilities is recommended as particularly suitable, without mentioning or excluding others, that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited. "May" and "need not" indicate a course of action permissible within the limits of the document. "Can" and "cannot" are used for statements of possibility and capability, whether material, physical or causal.

1.3 References

The following standards are referenced in this text. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based upon this document are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. ANSI and TIA maintain registers of currently valid national standards published by them.

1.3.1 Normative References

1. Reserved.
2. 3GPP2 C.S0002-E v3.0, *Physical Layer Standard for cdma2000 Spread Spectrum Systems*, June 2011.
3. Reserved.
4. Reserved.
5. 3GPP2 C.S0005-E v3.0, *Upper Layer (Layer 3) Signaling Standard for cdma2000 Spread Spectrum Systems*, June 2011.
6. Reserved.

---

[2] cdma2000® is the trademark for the technical nomenclature for certain specifications and standards of the Organizational Partners (OPs) of 3GPP2. Geographically (and as of the date of publication), cdma2000® is a registered trademark of the Telecommunications Industry Association (TIA-USA) in the United States.

7. 3GPP2 C.S0016-D v2.0, *Over-the-Air Service Provisioning of Mobile Stations in Spread Spectrum Systems*.
Editor's Note: The above document is a work in progress and should not be referenced unless and until it is approved and published. Until such time as this Editor's Note is removed, the inclusion of the above document is for informational purposes only.
8. C.S0015-B v2.0, *Short Message Service for Spread Spectrum Systems*, October 2005.
9. ITU-T Recommendation E.212, *Identification Plan for Land Mobile Stations*, November 1998.
10. Reserved.
11. Reserved.
12. Reserved
13. Reserved
14. TIA-95-B, *Mobile Station - Base Station Compatibility Standard for Wideband Spread Spectrum Cellular Systems*, October 2004.
15. 3GPP2 X.S0004-E v10.0, *Mobile Application Part*, January 2010.
16. Reserved.
17. 3GPP TS 51.011 V4.15.0 *Specification of the Subscriber Identity Module-Mobile Equipment (SIM-ME) Interface*, June 2005.
18. ETSI TS 102 221 V9.2.0, *Smart cards; UICC-Terminal Interface; Physical and logical Characteristics*, October 2010.
19. Reserved.
20. 3GPP2 S.S0053-0 v2.0, *Common Cryptographic Algorithms*, May 2009.
21. Reserved.
22. Reserved.
23. 3GPP2 X.S0011-D v2.0, *cdma2000 Wireless IP Network Standard*, November 2008.
24. IETF RFC 3344, *IP Mobility Support*, August 2002.
25. IETF RFC 2794, *Mobile IP Network Access Identifier Extension for IPv4*, March 2000.
26. IETF RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*, June 2000.
27. IETF RFC 4721, *Mobile IPv4 Challenge/Response Extensions*, January 2007.
28. 3GPP2 C.S0024-B v2.0, *cdma2000 High Rate Packet Data Air Interface Specification*, April 2007.
29. 3GPP2 A.S0008-C v2.0, *Interoperability Specification (IOS) for High Rate Packet Data (HRPD) Network Access Interfaces*, January 2009.
30. 3GPP TS 31.102 V8.6.0, *Characteristics of the Universal Subscriber Identity Module (USIM) application*, June 2009.
31. 3GPP TS 31.103 V8.1.0, *Characteristics of the IP Multimedia Services Identity Module (ISIM) Application*, June 2009.

32. Reserved.
33. IETF RFC 3261, *SIP: Session Initialization Protocol*, June 2002.
34. IETF RFC 4282, *The Network Access Identifier*, December 2005.
35. Reserved
36. 3GPP2 S.S0083-A v1.0, *Broadcast-Multicast Service Security Framework*, September 2004.
37. 3GPP2 X.S0016-200-A v1.0, *MMS Stage-2, Functional Description*, February 2006.
38. 3GPP TS 23.038 V8.2.0, *Alphabets and language-specific information,* September 2008.
39. 3GPP2 X.S0016-310-0 v2.0, *MMS MM1 Stage-3 Using OMA/WAP*, July 2004.
40. 3GPP2 X.S0016-311-0 v1.0, *MMS MM1 Stage-3 Using M-IMAP for message submission and retrieval*, May 2003.
41. 3GPP2 X.S0016-312-0 v1.0, *MMS MM1 Stage-3 Using SIP*, July 2004.
42. 3GPP2 S.S0055-A v3.0, *Enhanced Cryptographic Algorithms*, September 2005.
43. Reserved.
44. 3GPP2 C.S0068-0 v1.0, *ME Personalization for cdma2000 Spread Spectrum Systems*, June 2006.
45. 3GPP2 S.S0086-B, *IMS Security Framework*, December 2005.
46. IETF RFC 3629, *UTF-8, a transformation format of ISO 10646*. November 2003.
47. ITU E.118, *The international telecommunication charge card*, February 2001.
48. ITU X.509, *Public-key and attribute certificate frameworks*, August 2005.
49. ITU X.690, *ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*, November 2008.
50. IETF RFC 2315, *PKCS #7: Cryptographic Message Syntax Version 1.5*, March 1998.
51. RSA PKCS #12 v1.0, *Personal Information Exchange Syntax*, June 1999.
52. Reserved.
53. IETF RFC 1738, *Uniform Resource Locators (URL)*, December 1994.
54. 3GPP2 C.S0017-012-A v1.0, *Data Service Options for Spread Spectrum Systems: Service Options 33 and 66*, July 2004.
55. ISO/IEC 7816-4. *Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange*. 2005.
56. 3GPP2 C.S0035-A v2.0, *CDMA Card Application Toolkit (CCAT)*, August 2007.
57. 3GPP2 C.S0057-D, *Band Class Specification for cdma2000 Spread Spectrum Systems*, September 2009.
58. 3GPP2 X.S0022-A v1.0, *Broadcast and Multicast Service in cdma2000 Wireless IP Network,* April 2007.
59. 3GPP2 S.S0078-B v1.0, *Common Security Algorithms*, February, 2008.

60. 3GPP TS 31.101 v10.0.1. UICC-terminal interface; Physical and logical characteristics. June 2011.

61. IETF RFC 2195. *IMAP/POP AUTHorize Extension for Simple Challenge/Response*. September 1997.

62. IETF RFC 2617. *HTTP Authentication: Basic and Digest Access Authentication*. June 1999.

63. IETF RFC 3310. *Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)*. September 2002.

64. IETF RFC 2831. *Using Digest Authentication as a SASL Mechanism*. May 2000.

65. IETF RFC 2444. *The One-Time-Password SASL Mechanism*. October 1998.

66. IETF RFC 2222. *Simple Authentication and Security Layer (SASL)*. October 1997.

1.3.2 Informative References

1. C.R1001-H, "*Administration of Parameter Value Assignments for cdma2000 Spread Spectrum Standards*", July, 2011.

1.4 Terms

3GPD. Third Generation Packet Data.

Access Network (AN). The network equipment providing data connectivity between a packet switched data network (typically the Internet) and the access terminals. An access network is equivalent to a base station in [2].

Access Terminal (AT). A device providing data connectivity to a user. An access terminal may be connected to a computing device such as a laptop personal computer or it may be a self-contained data device such as a personal digital assistant. An access terminal is equivalent to a mobile station in [2].

A-key. A secret, 64-bit pattern stored in the mobile station and HLR/AC. It is used to generate or update the mobile station's Shared Secret Data.

Authentication. A procedure used by a base station to validate a mobile station's identity.

Authentication Center (AC). An entity that manages the authentication information related to the mobile station.

BAK. BCMCS related parameter. See [36].

BAK_Expire. BCMCS related parameter. See [36].

BAK_ID. BCMCS related parameter. See [36].

Base Station. A fixed station used for communicating with mobile stations. Depending upon the context, the term base station may refer to a cell, a sector within a cell, an MSC, an OTAF or other part of the wireless system. See also MSC and OTAF.

BCMCS. Broadcast Multicast Service.

BCMCS_Flow_ID. BCMCS related parameter. See [36].

BCMCS Root Key. A secret 128-bit pattern used for BCMCS (Broadcast Multicast Service). Defined as 'Registration Key' in [36].

BIP. Bearer Independent Protocol. See [56].

Card Session. See [17].

CDMA Session. That part of the *Card Session* dedicated to the CDMA operation.

CAVE. The algorithm currently used in [15] for Authentication and Key Generation.

Cyclic Redundancy Code (CRC). A class of linear error detecting codes which generate parity check bits by finding the remainder of a polynomial division.

DF. Dedicated File.

Diffie/Hellman. The key exchange mechanism used by [7].

ECC. Emergency Call Code, a number, that when dialed by the user, is to be treated as an emergency call.

ECMEA. Enhanced Cellular Message Encryption Algorithm

ECMEA_NF. Enhanced Cellular Message Encryption Algorithm (Non Financial)

EF. Elementary File.

Electronic Serial Number (ESN). An identifier that is either an ESN_ME or a UIMID.

ESN_ME. A 32-bit number that is either a pESN or a unique value assigned to a mobile station.

EUIMID. Expanded R-UIM Identifier.

Home Location Register (HLR). The location register to which a MIN/IMSI is assigned for record purposes such as subscriber information.

Home System. The cellular system in which the mobile station subscribes for service.

ICC. Integrated Circuit(s) Card.

ICCID. ICC Identification.

IMS. IP Multimedia Subsystem.

IMSI_M. MIN-based IMSI using the lower 10-digits to store the MIN.

IMSI_T. True IMSI not associated with MIN. This could be 15 digits or fewer.

IMS Root Key. A secret 128-bit pattern used for IMS (IP Multimedia Subsystem).

International Mobile Subscriber Identity (IMSI). A method of identifying subscribers in the land mobile service as specified in [9].

IRM. International Roaming MIN.

Long Code Mask. A 42-bit binary number that creates the unique identity of the long code. See also Public Long Code, Private Long Code, Public Long Code Mask, and Private Long Code Mask.

LF_EUIMID. Long form of EUIMID, which is ICCID based. In this document this term refers to the entire 20 digit/10 octet contents of EF_{ICCID} even though this will include a check digit and a padding digit.

LSB. Least significant bit.

M/O. Mandatory/Optional.

MAC. Message authentication code

MAC-A. MAC used for authentication and key agreement

MAC-I. Message Authentication Code for message integrity. The 32-bit output of the message integrity algorithm that allows the receiver to authenticate the message

ME. Mobile Equipment.

MEID_ME. A 56-bit number assigned by the mobile station manufacturer, uniquely identifying the mobile station equipment.

MF. Master File.

Mobile Country Code (MCC). A part of the E.212 IMSI identifying the home country. See [9].

Mobile Directory Number (MDN). A dialable directory number which is not necessarily the same as the mobile station's air interface identification, i.e., MIN, IMSI_M or IMSI_T.

Mobile Equipment (ME). An R-UIM capable mobile station without an R-UIM inserted.

Mobile Equipment Identifier (MEID). An identifier that is either an MEID_ME or an SF_EUIMID.

Mobile Identification Number (MIN). The 34-bit number that is a digital representation of the 10-digit number assigned to a mobile station.

Mobile Network Code (MNC). A part of the E.212 IMSI identifying the home network within the home country. See [9].

Mobile Station (MS). A station, fixed or mobile, which serves as the end user's wireless communication link with the base station. Mobile stations include portable units (e.g., hand-held personal units) and units installed in vehicles.

Mobile Station Originated Call. A call originating from a mobile station.

Mobile Station Terminated Call. A call received by a mobile station (not to be confused with a disconnect or call release).

MSB. Most significant bit.

Network. A network is a subset of a wireless system, such as an area-wide wireless network, a private group of base stations, or a group of base stations set up to handle a special requirement. A network can be as small or as large as needed, as long as it is fully contained within a system. See also System.

Network Identification (NID). A number that uniquely identifies a network within a wireless system. See also System Identification.

Number Assignment Module (NAM). A set of MIN/IMSI-related parameters stored in the mobile station.

Over-the-Air Service Provisioning Function (OTAF). A configuration of network equipment that controls OTASP functionality and messaging protocol.

Over-the-Air Parameter Administration (OTAPA). Network initiated OTASP process of provisioning mobile station operational parameters over the air interface.

Over-the-Air Service Provisioning (OTASP). A process of provisioning mobile station operational parameters over the air interface.

Parity Check Bits. Bits added to a sequence of information bits to provide error detection, correction or both.

P-CSCF. Proxy Call Session Control Function

Preferred Roaming List (PRL). See SSPR.

Private Long Code. The long code characterized by the private long code mask.

Private Long Code Mask. The long code mask used to form the private long code.

pseudo-ESN (pESN). A non-unique 32-bit number hashed from MEID and used in place of ESN.

pseudo-UIMID (pUIMID). A 32-bit number hashed from EUIMID and used in place of UIMID.

Release. A process that the mobile station and base station use to inform each other of call disconnect.

RFU. Reserved for future use.

Roamer. A mobile station operating in a wireless system (or network) other than the one from which service was subscribed.

Root Key. A secret 128-bit pattern permanently stored in the R-UIM.

R-UIM. Removable UIM.

SF_EUIMID. Short form of EUIMID. An EUIMID selected from MEID numbering resources.

Secure Mode. Network initiated mode of communicating operational parameters between a mobile station and network based provisioning entity in an encrypted form.

Service Option. A service capability of the system. Service options may be applications such as voice, data or facsimile. See [Informative 1].

Service Programming Lock (SPL). A protection provided for preventing the over-the-air provisioning of certain parameters by an unauthorized network entity by way of verifying the Service Programming Code (SPC).

Shared Secret Data (SSD). A 128-bit pattern stored in the mobile station (in semi-permanent memory) and known by the base station. SSD is a concatenation of two 64-bit subsets: SSD_A, which is used to support the authentication procedures, and SSD_B,

which serves as one of the inputs to the process generating the encryption mask and private long code.

SIP. Session Initialization Protocol

SIM. Subscriber Identity Module.

SK. BCMCS related parameter. See [36].

SK_RAND. BCMCS related parameter. See [36].

SMCK. Secure Mode Ciphering Key**.**

SP_LOCK_STATE - A locking state of the OTASP/OTAPA programmable parameters in the R-UIM. If SP_LOCK_STATE = '1', the parameters cannot be programmed.

SPASM. See Subscriber Parameter Administration Security Mechanism.

SPC. Service Programming Code.

SRTP. Secure Real Time Transport Protocol. See [36].

Subscriber Parameter Administration Security Mechanism (SPASM). Security mechanism protecting parameters and indicators of active NAM from programming by an unauthorized network entity during the OTAPA session.

SW1/SW2. Status Word 1/Status Word 2.

System. A system is a wireless telephone service that covers a geographic area such as a city, metropolitan region, county or group of counties. See also Network.

System Identification (SID). A number uniquely identifying a wireless system.

System Selection Code. A part of the Activation Code that specifies the user selection of a Band and a Block operated by the selected service provider.

System Selection for Preferred Roaming (SSPR). A feature that enhances the mobile station system acquisition process based on the set of additional parameters stored in the mobile station in the form of a Preferred Roaming List ($PR\_LIST_{S\text{-}P}$).

TK. BCMCS related parameter. See [36].

TK_RAND. BCMCS related parameter. See [36].

TMSI. Temporary Mobile Station Identity.

UAK. UIM Authentication Key. A 128-bit pattern produced by AKA that is used for R-UIM authentication.

UMAC. UIM-Present MAC. A 32-bit output of the UMAC algorithm computed by R-UIM based on MAC-I, which provides a means for the mobile station to prove that the R-UIM was present at the time the message is formed.

UCS2. Universal Multiple-Octet Coded Character Set.

UIM. User Identity Module.

UIMID. A 32-bit identifier that is either a number unique to the R-UIM or a non-unique pUIMID.

URI. Universal Resource Identifier.

VPM. Voice Privacy Mask.

WLAN Root Key. A secret 128-bit pattern used for WLAN services.

1.5 Parameters Stored Temporarily in the R-UIM

The following parameters with subscript "s" indicate a value stored temporarily in the R-UIM:

NAM_LOCK$_s$ – A network controlled status of the SPASM protection of the active NAM for the subsequent OTAPA session – temporarily stored in the R-UIM.

SPC$_s$ – Service Programming Code temporarily stored in the R-UIM if the Service Programming Lock feature is supported by the R-UIM.

SSD$_s$ – A secret 128-bit pattern for the Shared Secret Data temporarily stored in the R-UIM.

2 PHYSICAL, ELECTRICAL AND LOGICAL INTERFACES

2.1 Physical Interface

The physical interface of the R-UIM shall follow the definitions specified in section 4 of [60]. For the requirements in section 4A of [60], which are referenced by the present specification, the usage of the term “USIM” and “UICC” shall be equivalent to the term "R-UIM".

2.2 Electrical Interface

The electrical characteristics of the R-UIM shall follow the definitions specified in the sections of [17] shown in the following table.

Table 1. Electronic Signals and Transmission Protocols

Section of [17]	Title
5	Electronic Signals and Transmission Protocols
5.1	Electrical specifications
5.2	Initial communication establishment procedures
5.2.1	Error handling for speed enhancement
5.3	Transmission protocols
5.4	Clock

Terminals and R-UIM supporting other voltage technologies than Class A (see section 5.1 of 18) shall support at least 2 consecutive voltage classes, i.e. classes A and B, or classes B and C.

2.3 Logical Interface

The logical interface of the R-UIM shall follow the definitions specified in the sections of [17] shown in the following table. The Dedicated file ID for CDMA (used for EFs in section 3.4) is '7F25'.

Table 2. Logical Model

Section of [17]	Title
6	Application and File structure
6.1	SIM application structure
6.4	File types
6.4.1	Dedicated files
6.4.2	Elementary files
6.4.2.1	Cyclic EF
6.5	Methods for selecting a file

2.4 Security Features

Security-Related procedures and protocols are defined in section 4.

2.4.1 2G Authentication and Key Generation Procedure

See section 4.1 and 4.2.

2.4.2 Algorithms and Processes

The algorithm used by the R-UIM for authentication and key generation is CAVE (see section 4.1 and 4.2).

2.4.3 File Access Conditions

The file access conditions of the R-UIM shall follow the definitions specified in the section of [17] shown in the following table.

Table 3. File Access Conditions

Section of [17]	Title
7.3	File Access Conditions

2.4.4 3G AKA (Authentication and Key Agreement) Procedure and Function

See section 4.11 and 4.12.

2.5 Function Description

VOID

2.6 Command Description

The commands which are applicable for R-UIM are shown in the following sections.

2.6.1 R-UIM Supply Voltage Identification

R-UIM supporting Class B or C operating conditions (as specified in [17]) shall support the supply voltage indication as specified in section 9.2.1 of [17]. The table below shows the CDMA equivalent command for the listed GSM command.

GSM command	CDMA Equivalent command
SELECT DF_{GSM}	SELECT DF_{CDMA}

2.6.2 cdma2000 Specific Commands

These commands shall not be executed unless DF_{CDMA} or any sub-directory under DF_{CDMA} has been selected as the current directory and successful CHV1 verification procedure has been performed.

Table 4. Description of the cdma2000 Specific Commands

Section	Command Title	CLA	INS
4.4	R-UIM Security-Related Commands		
4.4.1	UPDATE SSD	'A0'	'84'
4.4.2	BASE STATION CHALLENGE	'A0'	'8A'
4.4.3	CONFIRM SSD	'A0'	'82'
4.4.4	AUTHENTICATE • Run CAVE • 3G Access AKA • EAP AKA	'A0'	'88'
4.4.5	GENERATE KEY / VPM	'A0'	'8E'
4.5	OTAPA / OTASP Commands		
4.5.1	MS KEY REQUEST	'A0'	'50'
4.5.2	KEY GENERATION REQUEST	'A0'	'52'
4.5.3	COMMIT	'A0'	'CC'
4.5.4	VALIDATE	'A0'	'CE'
4.5.5	CONFIGURATION REQUEST	'A0'	'54'
4.5.6	DOWNLOAD REQUEST	'A0'	'56'
4.5.7	SSPR CONFIGURATION REQUEST	'A0'	'EA'

Section	Command Title	CLA	INS
4.5.8	SSPR DOWNLOAD REQUEST	'A0'	'EC'
4.5.9	OTAPA REQUEST	'A0'	'EE'
4.5.10	PUZL CONFIGURATION REQUEST	'A0'	'F4'
4.5.11	PUZL DOWNLOAD REQUEST	'A0'	'F6'
4.5.12	3GPD CONFIGURATION REQUEST	'A0'	'FC'
4.5.13	3GPD DOWNLOAD REQUEST	'A0'	'48'
4.5.14	SECURE MODE	'A0'	'4A'
4.5.15	FRESH	'A0'	'4C'
4.5.16	SERVICE KEY GENERATION REQUEST	'A0'	'4E'
4.5.17	MMD CONFIGURATION REQUEST	'A0'	'C4'
4.5.18	MMD DOWNLOAD REQUEST	'A0'	'C6'
4.5.19	MMS CONFIGURATION REQUEST	'A0'	'42'
4.5.20	MMS DOWNLOAD REQUEST	'A0'	'46'
4.5.21	SYSTEM TAG CONFIGURATION REQUEST	'A0'	'C8'
4.5.22	SYSTEM TAG DOWNLOAD REQUEST	'A0'	'CA'
4.6	ESN and MEID Management Command		
4.6.1	STORE ESN_MEID_ME	'A0'	'DE'
4.8	Packet Data Security Related Commands		
4.8.1	COMPUTE IP AUTHENTICATION	'80'	'80'
4.9	BCMCS Sub-commands		
4.9	BCMCS • Retrieve SK • Update BAK • Delete BAK • Retrieve SRTP SK • Generate Authorization Signature • BCMCS Authentication	'A0'	'58'
4.10	Application Authentication Commands		
4.10.1	Application Authentication	'A0'	'5A'
4.12	AKA Commands		
4.12.1	UMAC GENERATION	'A0'	'5E'
4.12.2	CONFIRM_KEYS	'A0'	'5C'

2.6.3 Inherited Commands

The commands used with the R-UIM shall also follow the definitions specified in the sections of [17] shown in the following table.

Table 5. Description of the Inherited Commands [17]

Section of [17]	Title
9	Description of the Commands
9.1	Mapping Principles
9.2	Coding of the Commands
9.2.1	SELECT*
9.2.2	STATUS
9.2.3	READ BINARY
9.2.4	UPDATE BINARY
9.2.5	READ RECORD
9.2.6	UPDATE RECORD
9.2.7	SEEK
9.2.8	INCREASE
9.2.9	VERIFY CHV
9.2.10	CHANGE CHV
9.2.11	DISABLE CHV
9.2.12	ENABLE CHV
9.2.13	UNBLOCK CHV
9.2.14	INVALIDATE
9.2.15	REHABILITATE
9.2.17	SLEEP
9.2.18	GET RESPONSE
9.2.19	TERMINAL PROFILE
9.2.20	ENVELOPE
9.2.21	FETCH
9.2.22	TERMINAL RESPONSE
9.3	Definition and coding
9.4	Status conditions returned by the card (NOTE 1)
9.4.1	Responses to commands which are correctly executed
9.4.2	Responses to commands which are postponed

Section of [17]	Title
9.4.3	Memory management
9.4.4	Referencing management
9.4.5	Security management
9.4.6	Application independent errors
9.4.7	Commands versus possible status responses

NOTE 1: See section 2.6.2 for the summary of new and modified R-UIM status words.

The INCREASE command is coded as specified in [18] with the following limitations:

- Class = 'A0'
- P1, P2 = '00'
- P3 = 'Record length of selected cyclic file'

The response is according to the command parameters, as defined in [18]

*Response parameters/data in case of DF_{CDMA}:

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	CDMA specific data	21

CDMA 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	0 ≤ lgth ≤ 11

Bytes 1 - 22 are mandatory and shall be returned by the R-UIM. Bytes 23 and following are optional and may not be returned by the R-UIM.

NOTE 1: Byte 35 and following are RFU.

For the above bytes R-UIM shall follow definitions in section 9.2.1 of [17].

2.6.4 R-UIM Status Conditions

In response to commands sent by the ME to the R-UIM, the R-UIM returns status conditions contained in SW1 and SW2 to the ME. The status conditions defined in [17] and [55] apply to R-UIM, with the following status conditions and error descriptions taking precedence over [17]:

Table 6. R-UIM Status Conditions

SW1	SW2	Error Description
'94'	'02'	"Invalid BAK ID", in addition to "out of range (invalid address)" in [17]
'94'	'04'	"Invalid BCMCS Flow ID", in addition to "- file ID not found - pattern not found" in [17]
'98'	'34'	"Error, out of sequence", instead of "Error, Update SSD order sequence not respected" in [17]

2.7 Content of EFs

The ~~content of the EFs of the~~ R-UIM shall include contents described in the sections of [17] shown in the following table.

Table 7. Content of EFs

Section of [17]	Title
10.1	Contents of the EFs at the MF level
10.1.1	EF_{ICCID} (ICC Identification) (3)
10.1.2	EF_{LP} (Language Preference)
10.2	DFs at the GSM application level (4)
10.5	Contents of files at the telecom level
10.5.1	EF_{ADN} (Abbreviated dialing numbers)(1)
10.5.2	EF_{FDN} (Fixed dialing numbers)(1) / (2)
10.5.8	EF_{LND} (Last number dialed)(1)
10.5.9	EF_{SDN} (Service Dialing Numbers)(1)
10.5.10	EF_{EXT1} (Extension1)(1)
10.5.11	EF_{EXT2} (Extension2)(1)
10.5.12	EF_{EXT3} (Extension3)(1)
10.6	DFs at the telecom level
10.6.1	Contents of files at the telecom graphics level
10.6.1.1	EF_{IMG} (Image)
10.6.1.2	Image Instance Data Files

Notes:

(1) The numbers are stored in the same format as [17].
(2) See FDN procedures in [17] Annex C. The table below shows the CDMA equivalent of GSM files that are specially handled in FDN mode:

GSM File	CDMA Equivalent File
DF_{GSM}	DF_{CDMA}
EF_{LOCI}	EF_{TMSI}
EF_{IMSI}	EF_{IMSI_M}, EF_{IMSI_T}

(3) See section 3.4.83 for some additional restrictions on the contents of EF_{ICCID}.
(4) DFs at the GSM application level can be included in a multi-mode R-UIM (See Fig. 1 and Section 3).

In addition, the R-UIM may optionally provide an enhanced phonebook in a $DF_{PHONEBOOK}$ (File ID '5F3A') under $DF_{TELECOM}$ as defined in [30]. In this case, the content of $DF_{PHONEBOOK}$ on the R-UIM may include the sections of [30] shown in Table 8. , with the following restrictions:

- PIN shall be interpreted as CHV1 and PIN2 shall be interpreted as CHV2.
- SFIs (Short File Identifiers) shall not apply to the R-UIM.

EF_{ADN} and EF_{PBR} shall always be present if the $DF_{PHONEBOOK}$ is present.

To ensure proper inter-working in all terminals, the first EFs ADN and EXT1 files, if under $DF_{PHONEBOOK}$, are linked to the corresponding files under $DF_{TELECOM}$, i.e. EF_{ADN} = '6F3A' and EF_{EXT1} = '6F4A', respectively. This means that the contents of EFs ADN and EXT1 files under $DF_{PHONEBOOK}$ shall remain synchronized with those under $DF_{TELECOM}$

In addition, the Phonebook Restrictions defined in chapter 4.4.2.14 of [30] apply to the R-UIM.

Table 8. Content of EFs for R-UIM supporting the enhanced phonebook

Section of [30]	Title
4.4.2.1	EF_{PBR} (Phone Book Reference file) (1)
4.4.2.2	EF_{IAP} (Index Administration Phone book)
4.4.2.3	EF_{ADN} (Abbreviated dialing numbers) (1)
4.4.2.4	EF_{EXT1} (Extension1)
4.4.2.6	EF_{GRP} (Grouping file)
4.4.2.7	EF_{AAS} (Additional number Alpha String)
4.4.2.8	EF_{GAS} (Grouping Information Alpha String)
4.4.2.9	EF_{ANR} (Additional Number) (2)
4.4.2.10	EF_{SNE} (Second Name Entry) (2)
4.4.2.13	EF_{EMAIL} (e-mail address) (2)

Notes:

(1) The files EF_{PBC} (Phone Book Control), EF_{UID} (Unique Identifier), and EF_{CCP1} (Capability Configuration Parameters 1), EF_{PSC} (Phone Book Synchronisation Counter), EF_{CC} (Change Counter) and EF_{PUID} (Previous Unique Identifier) are not applicable to the R-UIM.

(2) "ADN File SFI" should be interpreted as "Last byte of ADN File Identifier" whenever a one-byte field is used to refer to an ADN file.

2.8 Application Protocol

The application protocol of the R-UIM shall follow the definitions specified in the sections of [17] shown in the following table.

Table 9. Application Protocol

Section of [17]	Title
11	Application protocol
11.1	General procedures
11.2.5	Administrative information request
11.2.6 (1)	SIM service table request
11.2.7 (2)	SIM phase request
11.2.8	SIM Presence Detection and Proactive Polling

(1) To CDMA mode, ME should read EF_{CST}.

(2) To CDMA mode, ME should read $EF_{REVISION}$.

2.9 CDMA Card Application Toolkit

The CDMA Card Application Toolkit of the R-UIM shall follow the definitions specified in [56]..

2.10 Coding of Alpha Fields in the R-UIM for UCS2

Reserved.

3 MULTI-MODE R-UIM DEDICATED FILE (DF) AND ELEMENTARY FILE (EF) STRUCTURE

The figure below depicts the multi-mode R-UIM file structure.

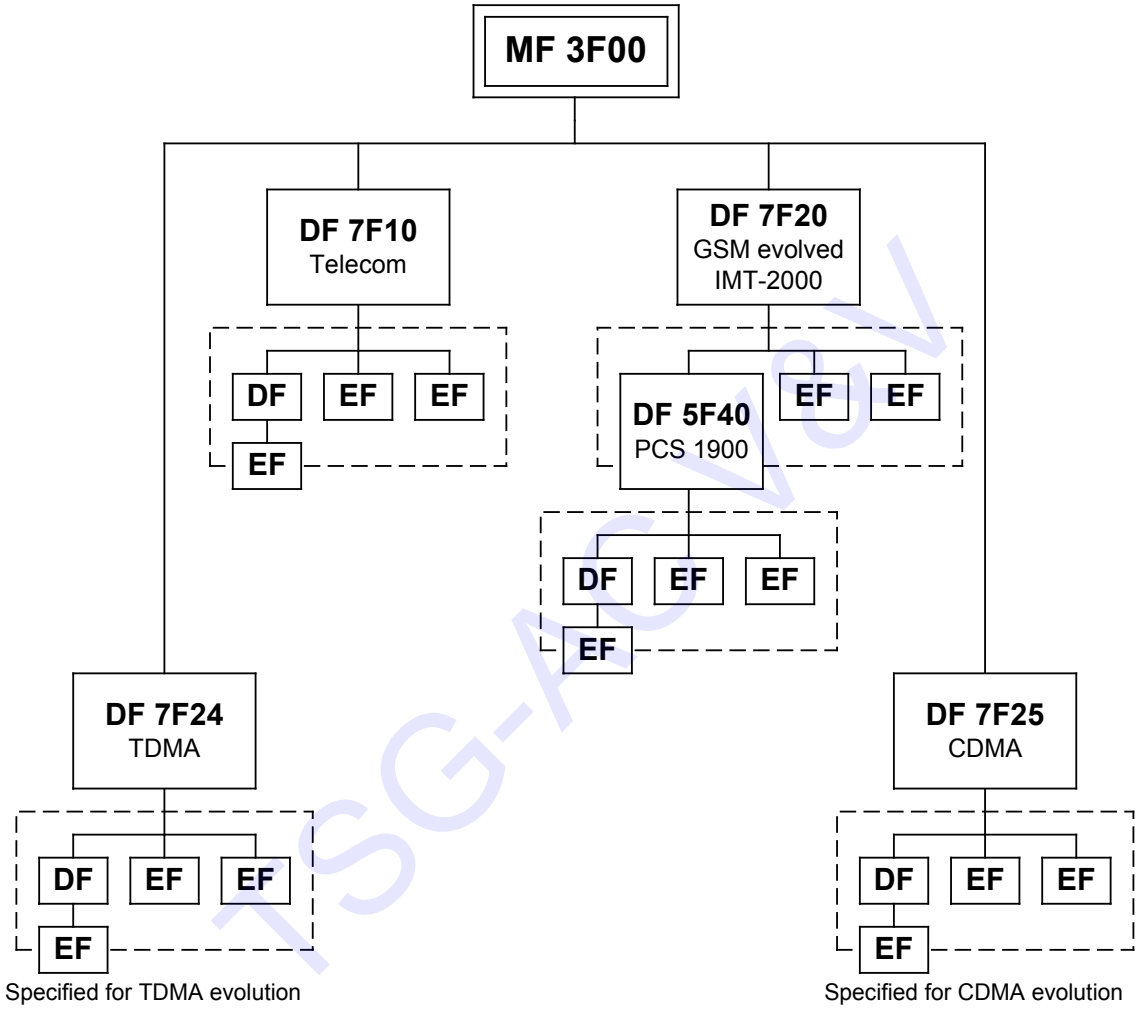


Figure 1. Dedicated File Structure

3.1 DF and EFs for ANSI-41 Based Applications

EFs assigned under DF '7F25' for storage of Number Assignment Module (NAM) parameters and operational parameters that are required for Analog/CDMA operation are based on [14] and the [2], [5], [28] family of standards as shown in [Informative 1].

Section 3.4 shows the detailed coding of these EFs. In this document, only single-NAM operation for CDMA is supported and therefore, each parameter is included once.

3.2 File Identifier (ID)

A file ID is used to address or identify each specific file. The file ID consists of two bytes and shall be coded in hexadecimal notation. File IDs are specified in section 0.

The first byte identifies the type of file. The numbering scheme for DFs and EFs is inherited from [17] as:

- '3F': Master File;
- '7F': First level Dedicated File;
- '5F': Second level Dedicated File;
- '2F': Elementary File under the Master File;
- '6F': Elementary File under the first level Dedicated File;
- '4F': Elementary File under the second level Dedicated File.

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.

3.3 Reservation of File IDs

In addition to the identifiers used for the files specified in the present document, the following file IDs are reserved for use by GSM and CDMA.

Dedicated Files:

- administrative use:
  '7F 4X', '5F 1X', '5F 2X'
- operational use:
  '7F 10' ($DF_{TELECOM}$), '7F 20' (DF_{GSM}), '7F 21' ($DF_{DCS1800}$), '7F 22' (DF_{IS-41}), '7F 23' (DF_{FP-CTS}), '7F 24' ($DF_{TIA/EIA-136}$), '7F 25' ($DF_{TIA/EIA-95}$), and '7F 2X', where X ranges from '6' to 'F'.
- reserved under '7F10':
  '5F 50' ($DF_{GRAPHICS}$)
- reserved under '7F20':
  '5F 30' ($DF_{IRIDIUM}$), '5F 31' ($DF_{Globalstar}$), '5F 32' (DF_{ICO}), '5F 33' (DF_{ACeS}), '5F 3X', where X ranges from '4' to 'F' for other MSS.
  '5F 40'($DF_{PCS-1900}$), '5F 4Y' where Y ranges from '1' to 'F';
  '5F 5X' where X ranges from '0' to 'F';
  '5F 60'(DF_{CTS}), '5F 6Y' where Y ranges from '1' to 'F';
  '5F 70' (DF_{SoLSA}), '5F 7Y' where Y ranges from '1' to 'F';
  '5F YX' where Y ranges from '8' to 'F' and X from '0' to 'F'.

Elementary files:

- administrative use:
  '6F XX' in the DFs '7F 4X'; '4F XX' in the DFs '5F 1X', '5F 2X'
  '6F 1X' in the DFs '7F 10', '7F 20', '7F 21', '7F 25';
  '4F 1X' in all second level DFs
  '2F 01', '2F EX' in the MF '3F 00';
- operational use:
  '6F 2X', '6F 3X', '6F 4X' in '7F 10' and '7F 2X';

'4F YX', where Y ranges from '2' to 'F' in all second level DFs.
'2F 1X' in the MF '3F 00'.

- reserved under '7F25' (DF_{CDMA}):
  '6F80': Reserved.
  From '6F81'to '6F89': Reserved for CDG.

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

3.4 Coding of EFs for NAM Parameters and Operational Parameters

All quantities shown in the EF descriptions are represented in binary format, unless otherwise specified. All unused, allocated bytes of memory are set to '00' unless otherwise specified. RFU bytes are also set to '00' unless otherwise specified. Some bits are marked as RFU. Some or all of these RFU bits may be used in the future for additional parameters. Therefore, all RFU bits shall be set to '0' (zero). The ME shall ignore the state of all RFU bits.

The dedicated file ID used for EFs in this section is '7F25' (CDMA).

References [5], [14] and [Informative 1] store parameters in several different types of memory:

- Variables stored in permanent memory which use the subscript p.
- Variables stored in semi-permanent memory which use the subscript s-p.
- Variables temporarily stored (including those parameters defined in section 1.5~~Sec. 1.2~~ which use the subscript s).

When an R-UIM is used, some of these variables are maintained in the R-UIM while other variables are maintained in the ME.

3.4.1 EF_{COUNT} (Call Count)

This EF stores the value of Call Count, $COUNT_{s-p}$.

Identifier: '6F21'	Structure: cyclic	Mandatory
Record Length: 2 bytes	Update activity: high	

Access Conditions:

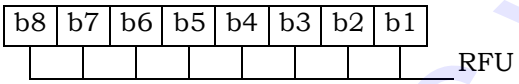
READ	CHV1
UPDATE	CHV1
INCREASE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 – 2	$COUNT_{s-p}$	M	2 bytes

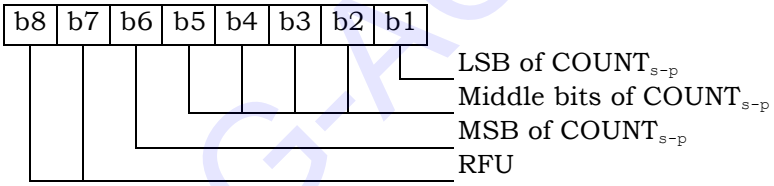
$COUNT_{s-p}$ is contained in the least significant 6 bits of the two-byte field.

Coding:

Byte 1:



Byte 2:



3.4.2 EF_{IMSI_M} (IMSI_M)

This EF stores the five components of IMSI_M.

Identifier: '6F22'	Structure: transparent	Mandatory
File size: 10 bytes	Update activity: low	

Access Conditions:

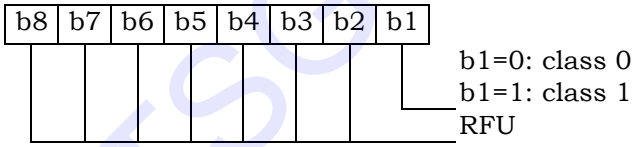
READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	CHV1

Bytes	Description	M/O	Length
1	$IMSI\_M\_CLASS_p$	M	1 byte
2 – 3	IMSI_M_S2 from $IMSI\_M\_S_p$	M	2 bytes
4 – 6	IMSI_M_S1 from $IMSI\_M\_S_p$	M	3 bytes
7	$IMSI\_M\_11\_12_p$	M	1 byte
8	IMSI_M_PROGRAMMED/ $IMSI\_M\_ADDR\_NUM_p$	M	1 byte
9 –10	MCC_M_p	M	2 bytes

$IMSI\_M\_CLASS_p$ - Class assignment of the IMSI_M.
$IMSI\_M\_ADDR\_NUM_p$ - Number of IMSI_M address digits.
MCC_M_p - Mobile country code.
$IMSI\_M\_11\_12_p$ - 11th and 12th digits of the IMSI_M.
$IMSI\_M\_S_p$ - The least significant 10 digits of the IMSI_M.

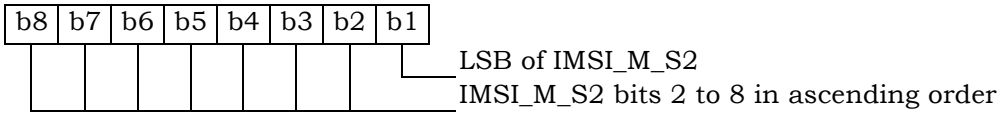
Coding:

Byte 1:

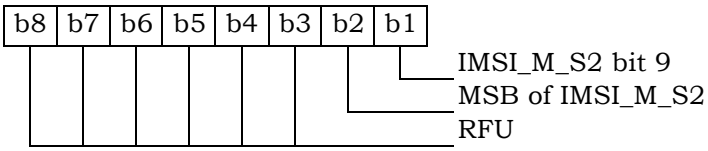


Byte 2, byte 3, byte 4, byte 5 and byte 6 are encoded as described in Section 2.3.1.1 of [5] and Section 6.3.1.1 of [14], "Encoding of IMSI_M_S and IMSI_T_S". IMSI_M_S2 contains the most significant digits of IMSI_M_S and IMSI_M_S1 contains the least significant digits of IMSI_M_S as described in Figure 2.3.1.-2 of [5] and Figure 6.3.1-2 of [14].

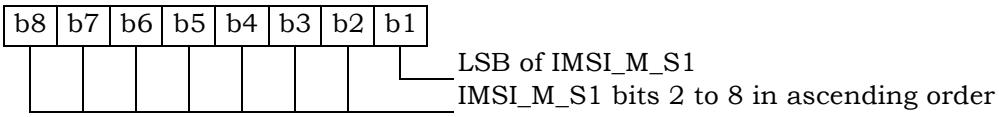
Byte 2:



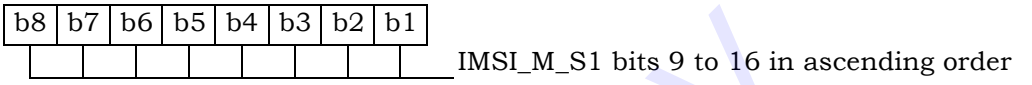
Byte 3:



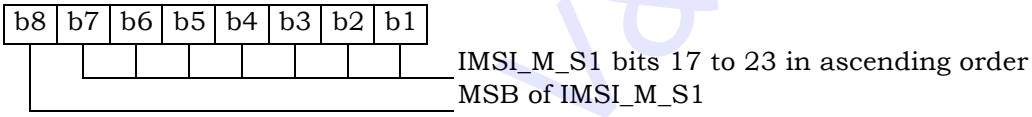
Byte 4:



Byte 5:

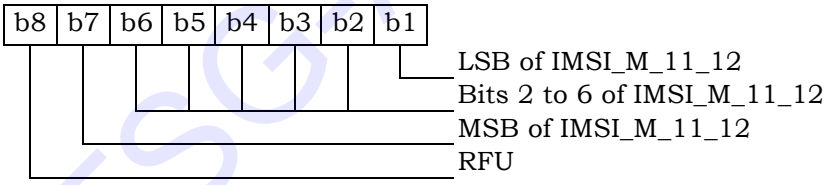


Byte 6:



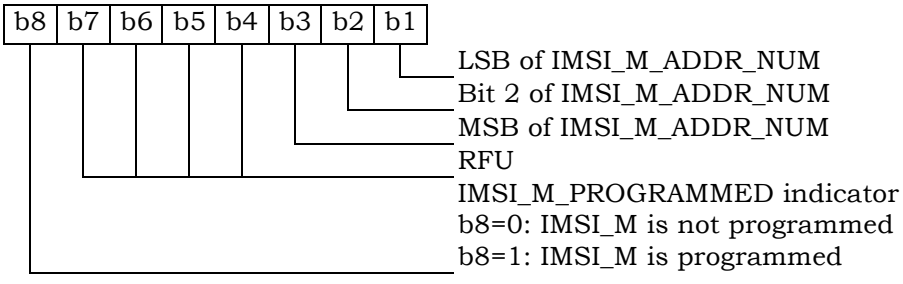
Byte 7 is encoded as described in Section 2.3.1.2 of [5] and Section 6.3.1.2 of [14], "Encoding of IMSI_M_11_12 and IMSI_T_11_12".

Byte 7:



Byte 8 is the binary equivalent of the IMSI_M_ADDR_NUM, as described in Section 2.3.1 of [5] and Section 6.3.1 of [14], "Mobile Station Identification Number".

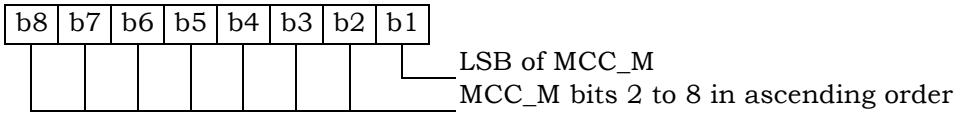
Byte 8:



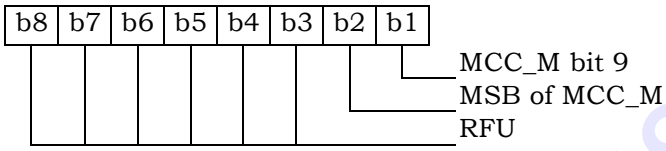
IMSI_M_PROGRAMMED shall be set to '1' if an IMSI_M is programmed otherwise it shall be set to '0'. If OTASP is used to update this EF, see section 4.5.3 COMMIT. See [5] or [14] for details on IMSI_M programming.

Byte 9 and byte 10 are encoded as described in Section 2.3.1.3 of [5] and Section 6.3.1.3 of [14], "Encoding of the MCC_M and MCC_T".

Byte 9:



Byte 10:



For R-UIM applications in systems that comply with [5] or [14], the parameter "MIN" is stored in EF_{IMSI_M}. For these instances, the 10 bits of "MIN2" are stored in bytes 2 and 3, with the coding shown above, while the 24 bits of "MIN1" are stored in bytes 4, 5, and 6.

The selection of IMSI_M or IMSI_T for use in the authentication process shall be in accordance with [14] Section 6.3.12.1 and [5] Section 2.3.12.1, which stipulate that the "MIN" portion of IMSI_M shall be used as an input parameter of the authentication calculation if IMSI_M is programmed and that a 32-bit subset of IMSI_T shall be used if only IMSI_T has been programmed.

3.4.3 EF_{IMSI_T} (IMSI_T)

This EF stores the five components of IMSI_T.

Identifier: '6F23'	Structure: transparent	Mandatory
File size: 10 bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	CHV1

Bytes	Description	M/O	Length
1	$IMSI\_T\_CLASS_p$	M	1 byte
2 – 3	IMSI_T_S2 from $IMSI\_T\_S_p$	M	2 bytes
4 – 6	IMSI_T_S1 from $IMSI\_T\_S_p$	M	3 bytes
7	$IMSI\_T\_11\_12_p$	M	1 byte
8	IMSI_T_PROGRAMMED/ $IMSI\_T\_ADDR\_NUM_p$	M	1 byte
9 –10	MCC_T_p	M	2 bytes

All byte descriptions, encodings and reference sections in [5] and [14] are identical to those described in Section 3.4.2, except that all references to "IMSI_M" shall apply to "IMSI_T".

EF_{IMSI_T} is not used to store a MIN.

3.4.4 EF_{TMSI} (TMSI)

This EF stores the Temporary Mobile Station Identity (TMSI). TMSI is assigned by the serving network and consists of 4 components, $ASSIGNING\_TMSI\_ZONE\_LEN_{s-p}$, $ASSIGNING\_TMSI\_ZONE_{s-p}$, $TMSI\_CODE_{s-p}$, and $TMSI\_EXP\_TIME_{s-p}$.

Identifier: '6F24'	Structure: transparent	Mandatory
File size: 16 bytes	Update activity: high	

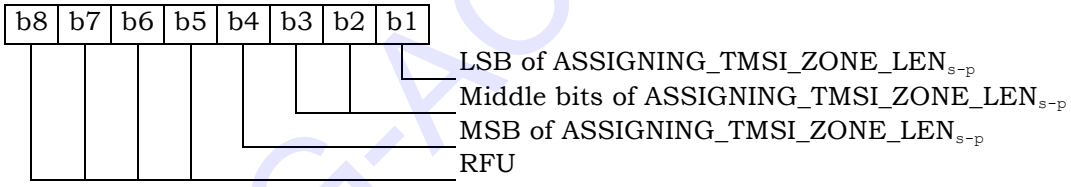
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	CHV1

Bytes	Description	M/O	Length
1	$ASSIGNING\_TMSI\_ZONE\_LEN_{s-p}$	M	1 byte
2 – 9	$ASSIGNING\_TMSI\_ZONE_{s-p}$	M	8 bytes
10 – 13	$TMSI\_CODE_{s-p}$	M	4 bytes
14 – 16	$TMSI\_EXP\_TIME_{s-p}$	M	3 bytes

Coding:

Byte 1:



Bytes 2 through 9 store the (up to) 8-octet TMSI Zone as described in Section 2.3.15 of [5] and Section 6.3.15 of [14]. These sections are entitled "Temporary Mobile Station Identity", "Overview" and "TMSI Assignment Memory" respectively. In each case the lowest-order octet shall be stored in the lowest-order byte (i.e., byte 2) of each set of contiguous 8 bytes, and successively higher octets stored in the next highest order bytes. Unused bytes shall be set to '00'.

Bytes 10 through 13 store the (2 to 4 octet) TMSI Code as described in the sections of [5] and [14] referenced above. In each case the lowest-order octet shall be stored in the lowest-order byte (i.e., byte 10) of each set of contiguous 4 bytes, and successively higher octets stored in the next highest order bytes. Unused bytes shall be set to '00'.

Bytes 14 through 16 store the TMSI Expiration Time as described in the sections of [5] and [14] referenced above. In each case the lowest-order octet shall be stored in the lowest-order byte (i.e., byte 14) of each set of contiguous 3 bytes, and successively higher octets stored in the next highest order bytes.

3.4.5 EF_{AH} (Analog Home SID)

This EF identifies the home SID when the mobile station is operating in the analog mode.

Identifier: '6F25'	Structure: transparent	Optional
File size: 2 bytes	Update activity: low	

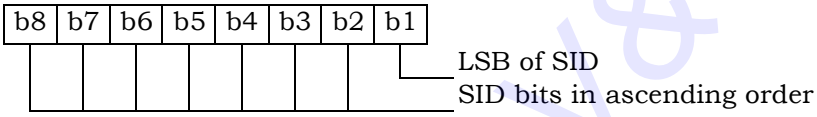
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

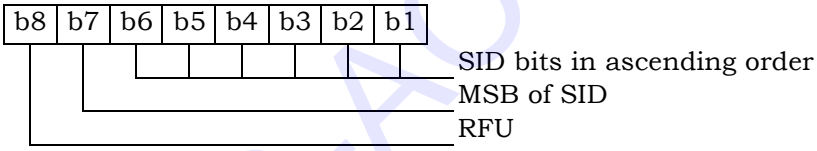
Bytes	Description	M/O	Length
1-2	Analog home SID ($HOME\_SID_p$)	M	2 bytes

Coding:

Byte 1:



Byte 2:



3.4.6 EF_{AOP} (Analog Operational Parameters)

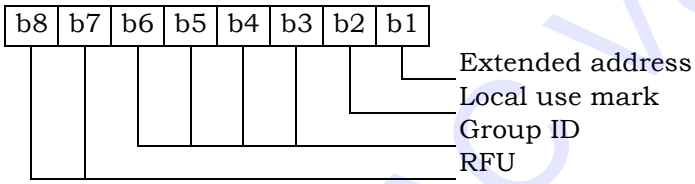
This EF includes the Extended Address bit (Ex_p), the Local Use Mark (LCM) and the Group ID (GID) field.

Identifier: '6F26'	Structure: transparent	Optional
File size: 1 byte	Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Analog Operational Parameters (Ex_p, LCM, GID)	M	1 byte

Coding:

Byte 1:



3.4.7 EF_{ALOC} (Analog Location and Registration Indicators)

This EF stores parameters related to Autonomous Registration memory ($NXTREG_{s-p}$ and SID_{s-p}) as well as the Location Area memory ($LOCAID_{s-p}$ and $PUREG_{s-p}$).

Identifier: '6F27'	Structure: transparent	Optional
File size: 7 bytes	Update activity: high	

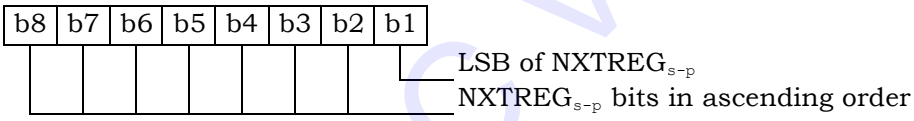
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

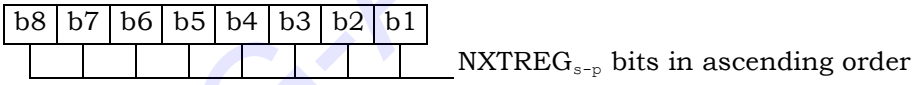
Bytes	Description	M/O	Length
1-3	$NXTREG_{s-p}$	M	3 bytes
4-5	SID_{s-p}	M	2 bytes
6-7	$LOCAID_{s-p}$, $PUREG_{s-p}$	M	2 bytes

Coding:

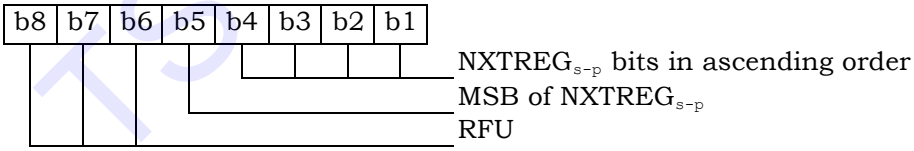
Byte 1:



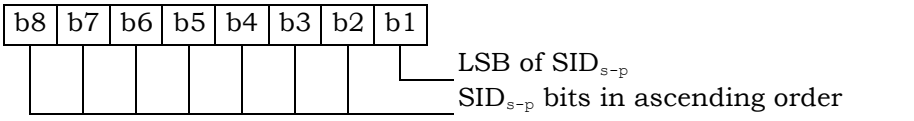
Byte 2:



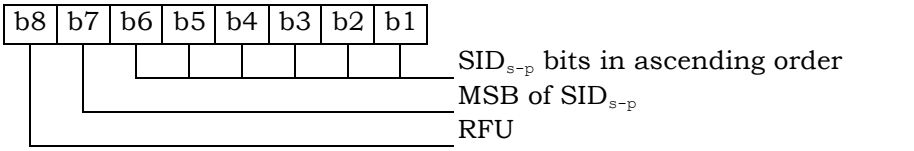
Byte 3:



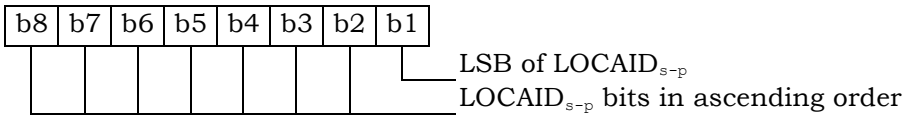
Byte 4:



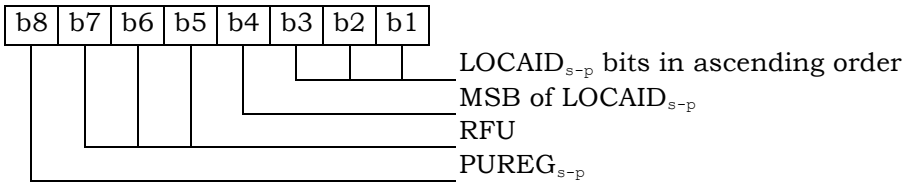
Byte 5:



Byte 6:



Byte 7:



3.4.8 $EF_{CDMAHOME}$ (CDMA Home SID, NID)

This EF identifies the home SID and NID when the mobile station is operating in the CDMA mode.

Identifier: '6F28'	Structure: linear fixed	Mandatory
Record length: 5 bytes	Update activity: low	

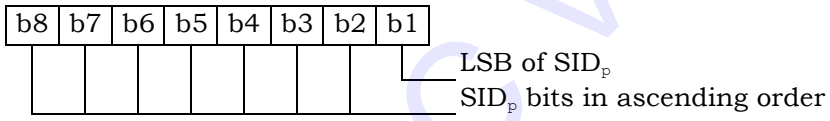
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

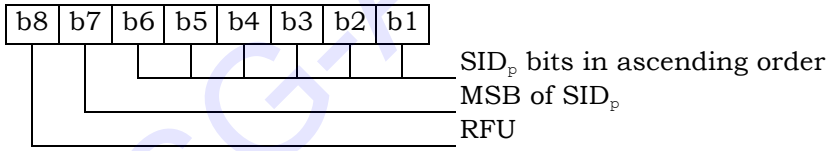
Bytes	Description	M/O	Length
1 – 2	CDMA Home SID (SID_p)	M	2 bytes
3 – 4	CDMA Home NID (NID_p)	M	2 bytes
5	Band Class	M	1 byte

Coding:

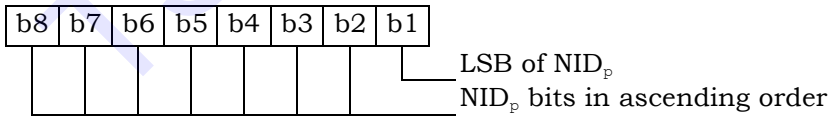
Byte 1:



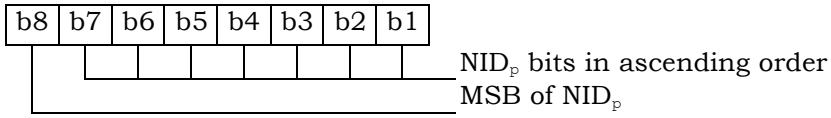
Byte 2:



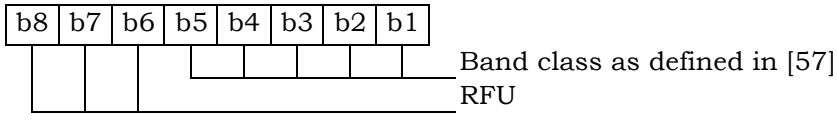
Byte 3:



Byte 4:



Byte 5:



3.4.9 EF_{ZNREGI} (CDMA Zone-Based Registration Indicators)

This EF stores the zone-based registration list "ZONE_LIST". The list includes a REG_ZONE and a corresponding SID, NID pair. Details are described in sections titled "Registration Memory", "Zone-Based Registration" and "Registration Procedures" of [5] and [14].

Identifier: '6F29'	Structure: linear fixed	Mandatory
Record length: 8 bytes	Update activity: high	

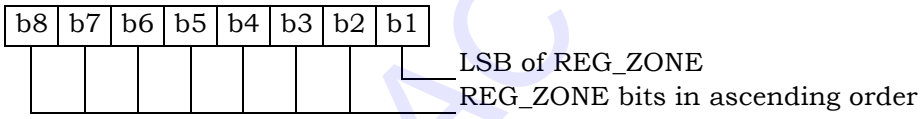
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

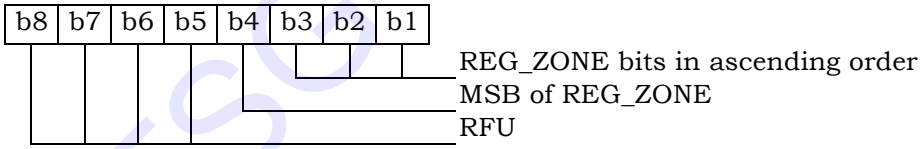
Bytes	Description	M/O	Length
1 – 2	REG_ZONE	M	2 bytes
3 – 4	SID	M	2 bytes
5 – 6	NID	M	2 bytes
7 – 8	RFU	M	2 bytes

Coding:

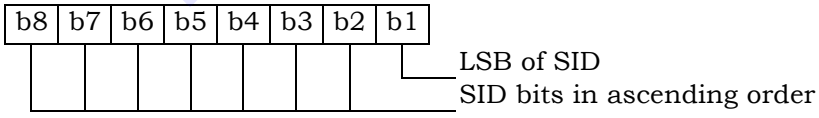
Byte 1:



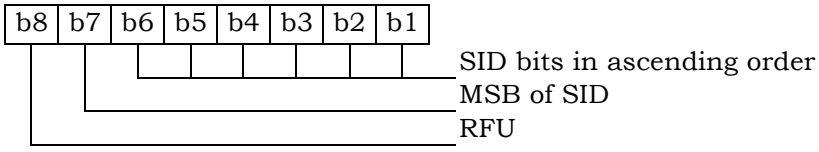
Byte 2:



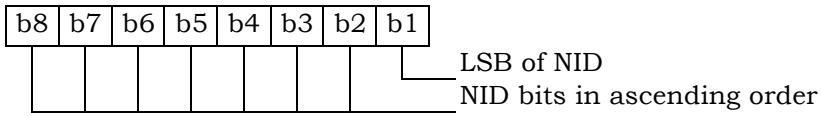
Byte 3:



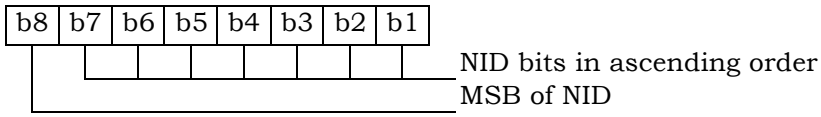
Byte 4:



Byte 5:



Byte 6:



3.4.10 EF_{SNREGI} (CDMA System-Network Registration Indicators)

This EF stores the SID and NID of the wireless system in which the mobile station last registered. This is described in sections of [5] and [14] titled "Registration Memory" and "Zone-Based Registration", respectively.

Identifier: '6F2A'	Structure: transparent	Mandatory
File size: 7 bytes	Update activity: high	

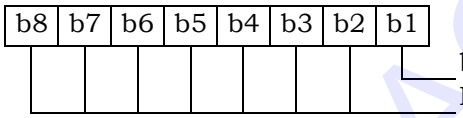
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

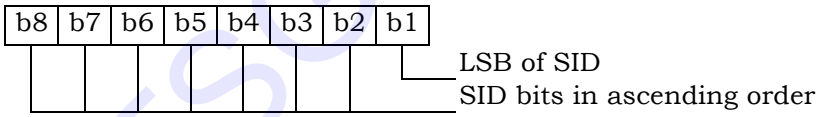
Bytes	Description	M/O	Length
1	N, size of SID/NID list (N=1)	M	1 byte
2 – 3	SID	M	2 bytes
4 – 5	NID	M	2 bytes
6 – 7	RFU	M	2 bytes

Coding:

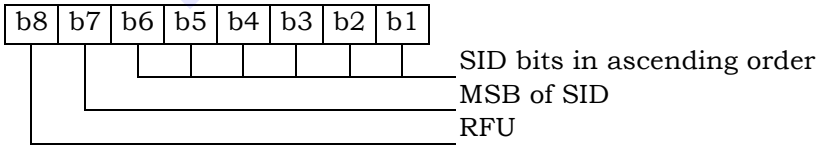
Byte 1:



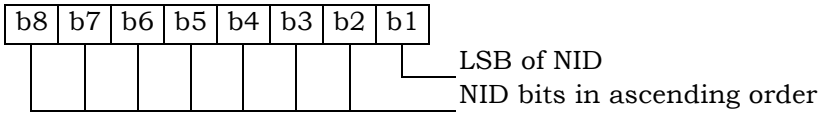
Byte 2:



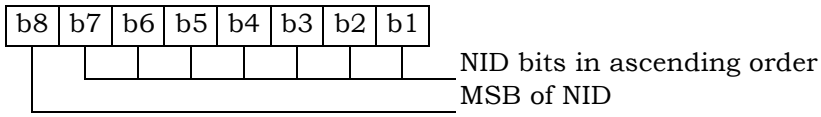
Byte 3:



Byte 4:



Byte 5:



3.4.11 $EF_{DISTREGI}$ (CDMA Distance-Based Registration Indicators)

This EF stores the Base Station Latitude (BASE_LAT_REG), the Base Station Longitude (BASE_LONG_REG) and the Registration Distance (REG_DIST_REG) of the base station to which the first access probe (for a Registration Message, Origination Message or Page Response Message) was transmitted after entering the System Access State.

Identifier: '6F2B'	Structure: transparent	Mandatory
File size: 8 bytes	Update activity: high	

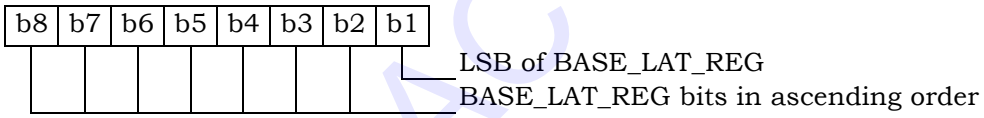
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1-3	BASE_LAT_REG	M	3 bytes
4-6	BASE_LONG_REG	M	3 bytes
7-8	REG_DIST_REG	M	2 bytes

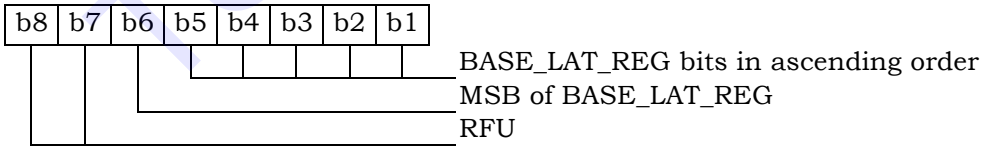
Coding:

Byte 1:

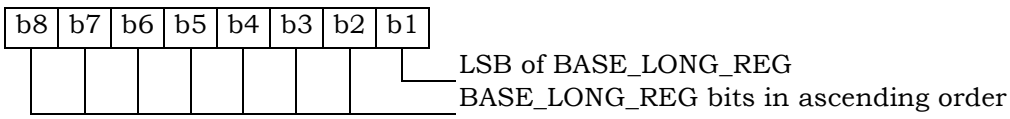


Byte 2:

Byte 3:

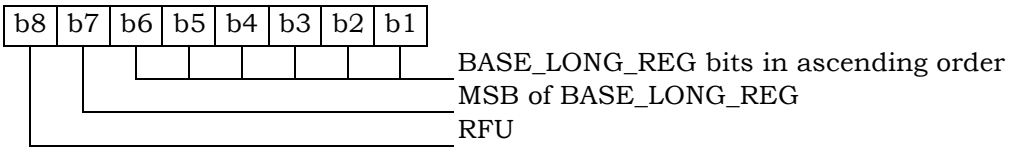


Byte 4:

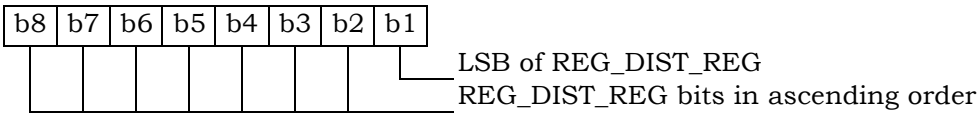


Byte 5:

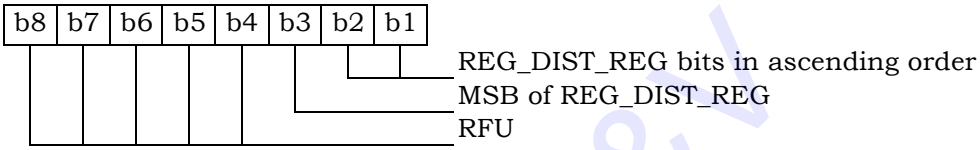
Byte 6:



Byte 7:



Byte 8:



NOTE: The parameters for Distance-Based Registration are described in Section 2.6.5.1.4 of [5] and Section 6.6.5.1.4 of [14].

3.4.12 EF$_{ACCOLC}$ (Access Overload Class ACCOLCp)

This EF defines the access overload class for the mobile station. This access overload class identifies which overload class controls access attempts by the mobile station and is used to identify redirected overload classes in global service redirection. For normal mobile stations, the 4-bit access overload class indicator is derived from the last digit of the associated decimal representation of the IMSI_M via decimal to binary conversion as specified in [5] and [14].

Identifier: '6F2C'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	

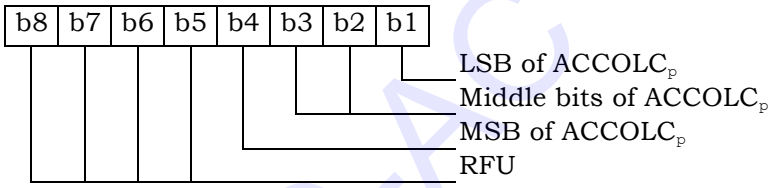
Access Conditions:

READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Access overload class (ACCOLC$_p$)	M	1 byte

Coding:

Byte 1:



3.4.13 EF_{TERM} (Call Termination Mode Preferences)

This EF contains the call termination preference MOB_TERM_HOMEp, MOB_TERM_SIDp and MOB_TERM_FOR_NIDp.

Identifier: '6F2D'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	

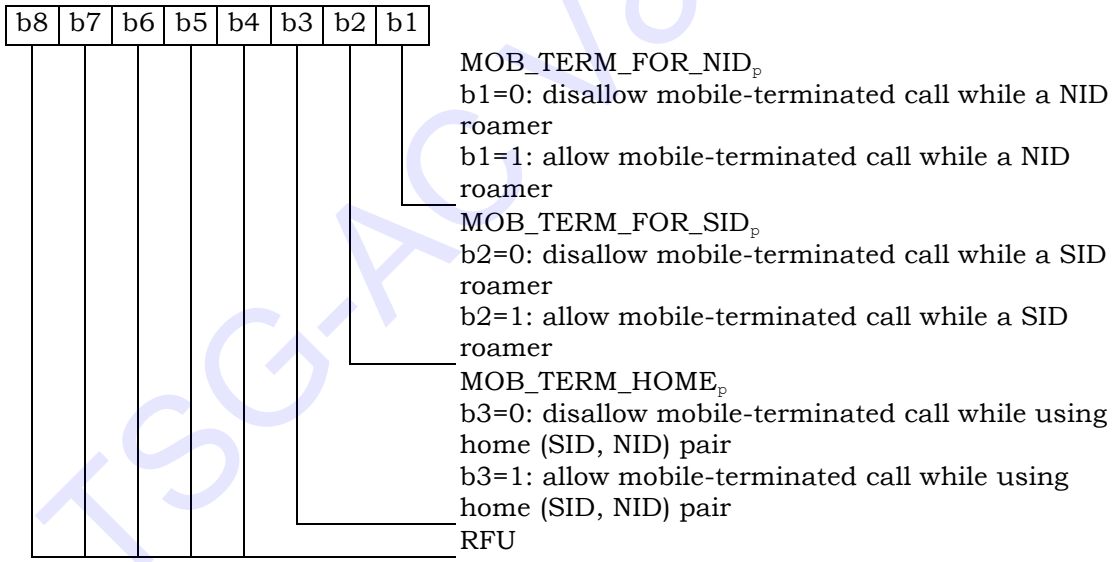
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Call termination preferences	M	1 byte

Coding:

Byte 1:



3.4.14 EF_{SSCI} (Suggested Slot Cycle Index)

This EF suggests a value for the mobile station's preferred slot cycle index for CDMA operation (see Section 2.3.11 of [5] or Section 6.3.11 of [14]). Since the mobile equipment may not support all the slot cycle indexes, the mobile equipment shall select the minimum, as the preferred slot cycle index defined in [5], between the slot cycle index supported by the mobile equipment and the suggested slot cycle index contained in the EF_{SSCI}.

Identifier: '6F2E'	Structure: transparent	Optional
File size: 1 byte	Update activity: low	

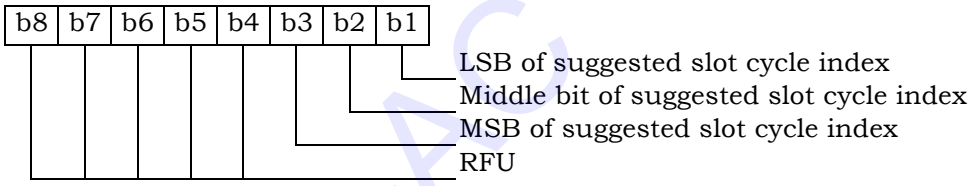
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Suggested slot cycle index	M	1 byte

Coding:

Byte 1:



3.4.15 EF_{ACP} (Analog Channel Preferences)

This EF specifies the analog mode channel preferences as determined by the service provider in accordance with the terms of the subscription. The items addressed are the Analog Initial Paging Channel, the Analog First Dedicated Control Channel for System A, the Analog First Dedicated Control Channel for System B, and the Number of Dedicated Control Channels to scan.

Identifier: '6F2F'	Structure: transparent	Optional
File size: 7 bytes	Update activity: low	

Access Conditions:

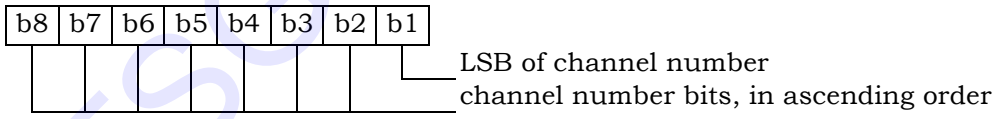
READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1-2	Analog Initial Paging Channel	M	2 bytes
3-4	Analog First Dedicated Control Channel System A	M	2 bytes
5-6	Analog First Dedicated Control Channel System B	M	2 bytes
7	Number of Dedicated Control Channel to Scan	M	1 byte

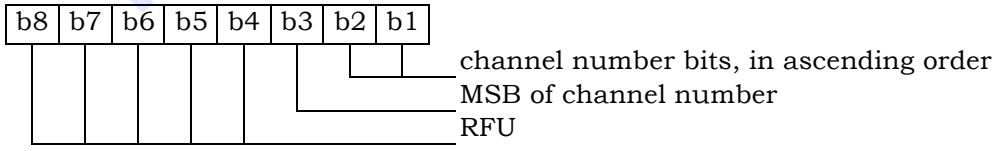
NOTE: Each channel is represented by an 11-bit binary number.

Coding:

Byte 1, 3, 5:



Byte 2, 4, 6:



3.4.16 EF_{PRL} (Preferred Roaming List)

This EF stores the Preferred Roaming List, as described in Section 3.5.5 of [7]. The Preferred Roaming List includes selection parameters from [5] and [14].

Identifier: '6F30'	Structure: transparent	Mandatory
File size: 'MAX_PR_LIST_SIZE for EF_{PRL}'	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-PR_LIST_SIZE	PR_LIST	M	PR_LIST_SIZE

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

- PR_LIST

  Contents:

  The Preferred Roaming List.

  Coding:

  As defined in section 3.5.5 of [7].

3.4.17 EF_{RUIMID} (Removable UIMID)

This EF stores a 32-bit electronic identification number (ID) unique to the R-UIM or a 32-bit pUIMID of the R-UIM. The file may store a 32-bit pUIMID constructed in the following way: The most significant 8 bits shall be 0x 80. The least significant 24 bits shall be the 24 least significant bits of SHA-1 digest of the entire EUIMID, either LF_EUIMID or SF_EUIMID[3] (based on n8 in CDMA service table) .[4]

Identifier: '6F31'	Structure: transparent	Mandatory
File size: 5 or 8 bytes	Update activity: low	

Access Conditions:

READ	ALW
UPDATE	Never
INVALIDATE	Never
REHABILITATE	Never

Bytes	Description	M/O	Length
1	Number of bytes	M	1 byte
2	Lowest-order byte	M	1 byte
3	:	M	1 byte
4	:	M	1 byte
5	:	M	1 byte
6	:	O	1 byte
7	:	O	1 byte
8	Highest-order byte	O	1 byte

[3] Example: if the LF_EUIMID (ICCID) is (hexadecimal) 89 (MSB) 01 01 01 23 45 67 89 01 4F (LSB), the pseudo-UIMID is (hexadecimal) 80 (Byte 5) 7D ED 89 (Byte 2), and with Byte 1 set to 04; if the 56-bit SF_EUIMID is (hexadecimal) FF (MSB) 00 00 01 12 34 56 (LSB), the pseudo-UIMID is (hexadecimal) 80 (Byte 5) 07 37 E1 (Byte 2), and with Byte 1 set to 04.

[4] The EUIMID (either form) is loaded into a 512-bit SHA-1 input block, starting with bit 1 of this block, to produce an output, from which the least significant 24 bits are used as the least significant 24 bits of EF(RUIMID). The 4-bit digits of EUIMID are loaded in the order d1, d2, d3, d4...dn-1, dn. Numbering the SHA-1 input buffer bits from 1 (first loaded) upwards, for each digit the most significant bit is loaded into the lowest numbered of four consecutive SHA-1 input bits and the least significant bit into the highest.

3.4.18 EF_{CST} (CDMA Service Table)

This EF indicates which services are allocated, and whether, if allocated, the service is activated. If a service is not allocated or not activated in the R-UIM, the ME shall not select or use that service.

Identifier: '6F32'	Structure: transparent	Mandatory
File size: N bytes (N ≥ 5)	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Services n1 to n4	M	1 byte
2	Services n5 to n8	M	1 byte
3	Services n9 to n12	M	1 byte
4	Services n13 to n16	M	1 byte
5	Services n17 to n20	M	1 byte
6	Services n21 to n24	O	1 byte
:	:	:	:
N	Services n(4N-3) to n(4N), where N>6	O	1 byte

Services:

Service n1 : CHV disable function
Service n2 : Abbreviated Dialing Numbers (ADN)
Service n3 : Fixed Dialing Numbers (FDN)
Service n4 : Short Message Storage (SMS)
Service n5 : HRPD
Service n6 : Enhanced Phone Book
Service n7 : Multi Media Domain (MMD)
Service n8 : SF_EUIMID-based EUIMID
Service n9 : MEID Support
Service n10 : Extension1
Service n11 : Extension2
Service n12 : SMS Parameters
Service n13 : Last Number Dialed (LND)
Service n14 : Service Category Program for BC-SMS
Service n15 : Messaging and 3GPD Extensions
Service n16 : Root Certificates
Service n17 : CDMA Home Service Provider Name
Service n18 : Service Dialing Numbers (SDN)
Service n19 : Extension3
Service n20 : 3GPD-SIP
Service n21 : WAP Browser
Service n22 : Java
Service n23 : Reserved for CDG

Service n24 : Reserved for CDG
Service n25 : Data Download via SMS Broadcast [56]
Service n26 : Data Download via SMS-PP [56]
Service n27 : Menu Selection [56]
Service n28 : Call Control [56]
Service n29 : Proactive R-UIM [56]
Service n30 : AKA
Service n31 : IPv6
Service n32 : RFU
Service n33 : RFU
Service n34 : RFU
Service n35 : RFU
Service n36 : RFU
Service n37 : RFU
Service n38 : 3GPD-MIP
Service n39 : BCMCS
Service n40 : Multimedia Messaging Service (MMS)
Service n41 : Extension 8
Service n42 : MMS User Connectivity Parameters
Service n43 : Application Authentication
Service n44 : Group Identifier Level 1
Service n45 : Group Identifier Level 2
Service n46 : De-Personalization Control Keys
Service n47 : Cooperative Network List
Service n48 : Call Control for Mobile Originated SMS Services [56]

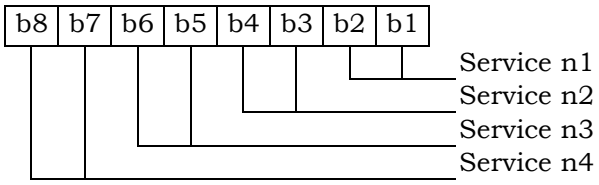
NOTE: Additional services, when defined, will be coded on further bytes in the EF.

Coding:
Each byte is used to code 4 services.
2 bits are used to code each service:
first bit = 1: service allocated
first bit = 0: service not allocated
where the first bit is b1, b3, b5 or b7;
second bit = 1: service activated
second bit = 0: service not activated
where the second bit is b2, b4, b6 or b8.

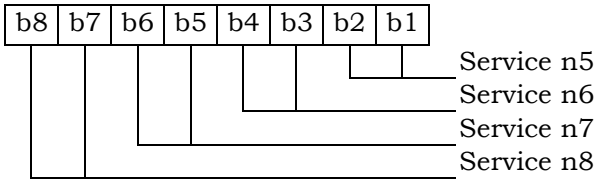
"Service allocated" means that the R-UIM has the capability to support the service.
"Service activated" means that the service is available.
Service delivery can only occur when service is allocated, service is activated and the R-UIM is operating in an environment that supports delivery of the service.
The following codings are possible:
first bit = 0: service not allocated, second bit has no meaning;
first bit = 1 and second bit = 0: service allocated but not activated;
first bit = 1 and second bit = 1: service allocated and activated.

The bits for services not yet defined shall be set to RFU. All bytes that are RFU shall be set to '00' and RFU bits will be set to '0'.

Byte 1:



Byte 2:



Etc.

If the R-UIM supports the FDN feature (FDN allocated and activated), a special mechanism shall exist in the R-UIM which invalidates EF_{IMSI_T}, EF_{IMSI_M} and EF_{TMSI} once during each CDMA session. This mechanism shall be invoked by the R-UIM automatically if FDN is enabled. This invalidation shall occur at least before the next command following selection of one of the above three EFs. FDN is enabled when the ADN is invalidated or not activated.

If service n8 (SF_EUIMID-based EUIMID) is not activated (either allocated or not), ME shall fill in EXT_UIM_ID INFO RECORD with the entire contents of EF_{ICCID} in response to *Status Request Message* defined in [5]. Otherwise, ME shall fill in EXT_UIM_ID INFO RECORD with SF_EUIMID from EF_{SF_EUIMID}.

3.4.19 EF_{SPC} (Service Programming Code)

This EF includes the Service Programming Code (SPC), having a value from 0 to 999,999. The default value is 0. Details of SPC are in [7], section 3.3.6.

Identifier: '6F33'	Structure: transparent	Mandatory
File size: 3 bytes	Update activity: low	

Access Conditions:

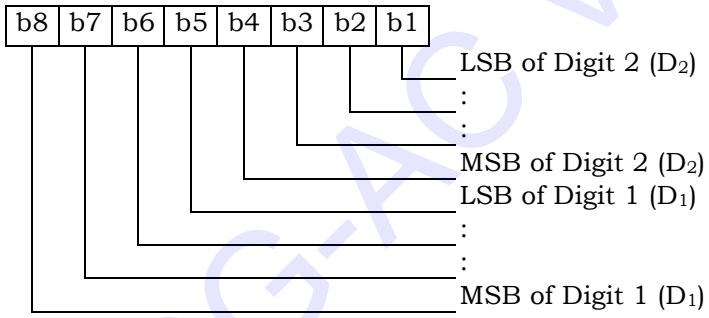
READ	ADM
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1-3	Service Programming Code	M	3 bytes

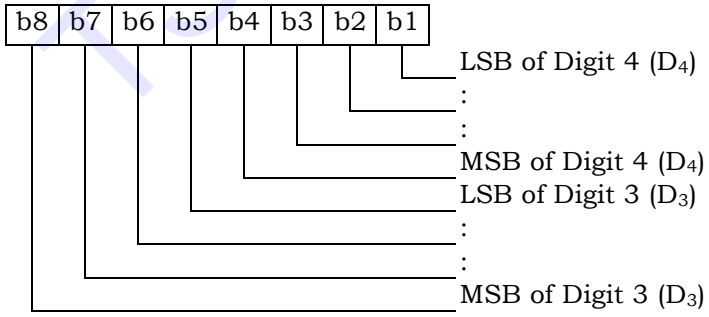
Coding:

SPC is a 6-digit number$D_1D_2D_3D_4D_5D_6$, where D_1 is the most significant digit and D_6 is the least significant digit. The coding of SPC in this EF is according to [7], section 4.5.4.2, whereby each digit is encoded in BCD format.

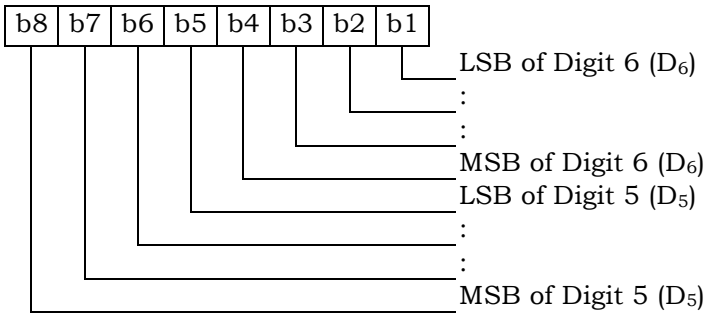
Byte 1:



Byte 2:



Byte 3:



3.4.20 $EF_{OTAPASPC}$ (OTAPA/SPC_Enable)

This EF contains user-entered control information that either prevents or (else) permits network manipulation of the SPC, and either prevents or (else) permits OTAPA to be performed on the NAM. This EF is based upon information in [7], sections 3.2.2 and 3.3.6. A successful base station response to an R-UIM initiated challenge is required prior to any network manipulation of OTAPA accessible files.

Identifier: '6F34'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	

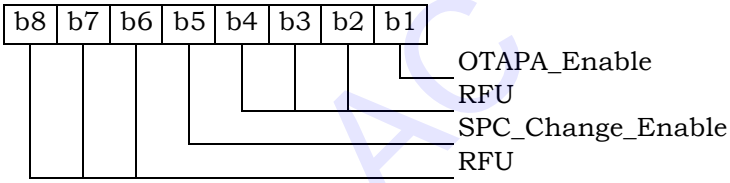
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	OTAPA/SPC_Enable	M	1 byte

Coding:

Byte 1:



For OTAPA_Enable, a value of '0' for the NAM indicates that the user consents to the performance of OTAPA for the NAM by the service provider. A value of '1' indicates that the user does not permit OTAPA to be performed on the NAM. Refer to [7], Section 3.2.2.

For SPC_Change_Enable~~SPC_Change Enable~~, a value of '0' for the R-UIM indicates that the user consents to allow the service provider to change the Service Programming Code from a default value (zero) to a non-default value (non-zero). An SPC_Change_Enable value of '1' indicates that the user denies permission for the service provider to change the SPC from a default value to a non-default value. See Sec. 3.3.6 of [7].

3.4.21 $EF_{NAMLOCK}$ (NAM_LOCK)

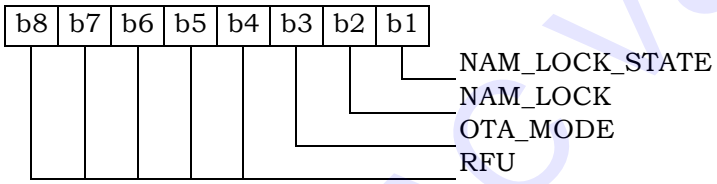
This EF stores the locked/unlocked state of the NAM. This EF is based upon information in [7].

Identifier: '6F35'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	SPASM protection indicator (NAM_LOCK) status	M	1 byte

Coding:

Byte 1:



Bit 1 gives the current NAM_LOCK_STATE. A value of '1' indicates that the NAM is locked by the SPASM protection mechanism. A value of '0' indicates that the NAM is unlocked.

Bit 2 gives the permanent NAM_LOCK setting. A value of '1' indicates that the SPASM protection mechanism must be satisfied for OTAPA. A value of '0' indicates that SPASM protection is not required.

Bit 3 gives the OTA_MODE for the current OTASP session. A value of '0' indicates user-initiated, and a value of '1' indicates network-initiated (OTAPA).

If an OTA programming session was initiated by the user as described in Section 3.2.1 of [7], SPASM does not protect access to the NAM parameters and indicators. In this case, the ME shall set the NAM_LOCK_STATE to '0.' The NAM_LOCK bit shall not be changed.

On invocation of an OTAPA session, the ME shall set the NAM_LOCK_STATE=NAM_LOCK.

The ME updates the OTA_MODE bit to tell the R-UIM how an OTASP session was initiated. The ME shall set this bit on initiation of an OTASP session. The R-UIM shall comply with the requirements in [7] (e.g. shall reject OTAPA REQUEST while in a user-initiated OTASP session).

3.4.22 EF_{OTA} (OTASP/OTAPA Features)

This EF stores a listing of OTASP/OTAPA features supported by the R-UIM, along with protocol revision codes. This EF is based on the information in [7] using the format and coding rules in section 3.5.1.7, including the subset of fields described below.

Identifier: '6F36'	Structure: transparent	Mandatory
File size: 2*NUM_FEATURES + 1 bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	NUM_FEATURES, number of OTASP/OTAPA features	M	1 byte
2	First FEATURE_ID	M	1 byte
3	First FEATURE_P_REV	M	1 byte
	…		
2*NUM_FEATURES	Last FEATURE_ID	M	1 byte
2*NUM_FEATURES+ 1	Last FEATURE_P_REV	M	1 byte

NOTE: Coding of features (FEATURE_ID) and protocol revisions (FEATURE_P_REV) is described in Table 3.5.1.7-1 (Feature Identifier) of [7].

3.4.23 EF$_{SP}$ (Service Preferences)

This EF describes the user's service preferences as defined in Section 2.3.10.1 of [5] or Sections 6.3.10.1 and 6.3.10.2 of [14].

Identifier: '6F37'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	

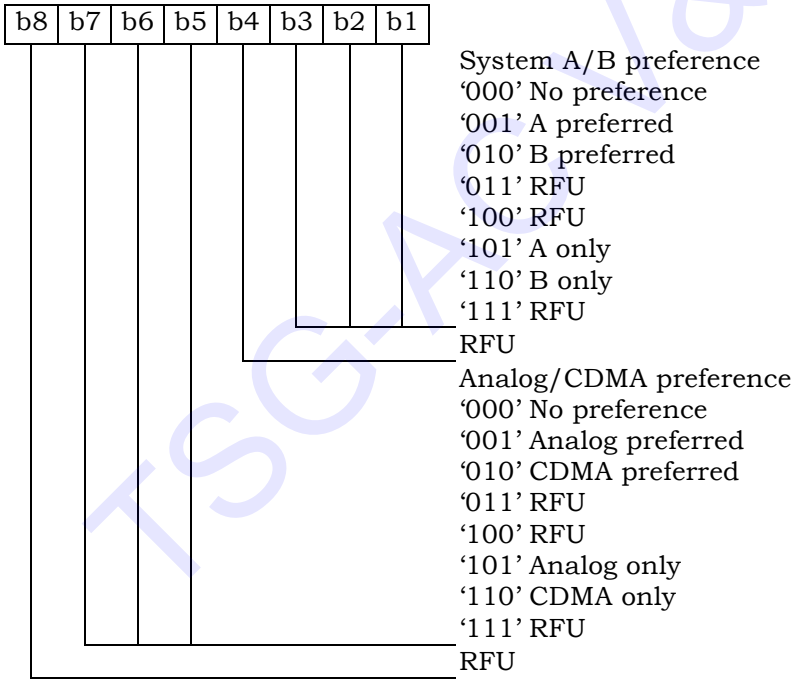
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Service Preferences (e.g. band class, analog vs. CDMA)	M	1 byte

Coding:

Byte 1:



3.4.24 $EF_{ESN\_MEID\_ME}$ (ESN_ME or MEID_ME)

This EF stores the 32-bit ESN_ME or 56-bit MEID_ME to which the R-UIM is attached.

Identifier: '6F38'	Structure: transparent	Mandatory
File size: 8 bytes	Update activity: low	

Access Conditions:

READ	ALW
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Number of bytes for ESN_ME or MEID_ME	M	1 byte
2	Least significant byte	M	1 byte
3	:	M	1 byte
4	:	M	1 byte
5	:	M	1 byte
6	:	M	1 byte
7	:	M	1 byte
8	Most significant byte	M	1 byte

Unused bytes shall be set to '00'.

3.4.25 $EF_{Revision}$ (R-UIM Revision)

This EF allows the ME to communicate with different versions of the R-UIM (i.e. R-UIM with different set of capabilities).

Identifier: '6F39'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	

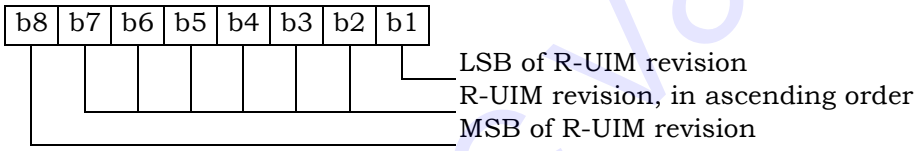
Access Conditions:

READ	ALW
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	R-UIM Revision	M	1 byte

Coding:

Byte 1:



An R-UIM complying with this specification shall set the R-UIM Revision to '00000100'.

3.4.26 EF_{RUIM_PL} (Preferred Languages)[5]

This EF assists the ME in offering a set of different languages (i.e. English, German, French, Japanese, etc.). From this set of languages, the user can choose to have the information displayed in the desired language.

Identifier: '6F3A'	Structure: transparent	Mandatory
File size: 2N bytes	Update activity: low	

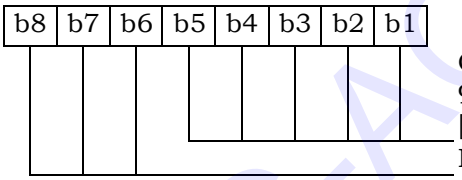
Access Conditions:

READ	ALW
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 – 2	First language code (highest priority)	M	2 bytes
3 – 4	Second language code	O	2 bytes
:	:	:	:
2N-1 – 2N	N^{th} language code (lowest priority)	O	2 bytes

Coding:

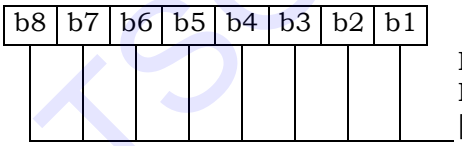
Byte 1:



CHARi encoding type as shown in Table 9.1-1, Data Field Encoding Assignments, in [Informative 1]

RFU

Byte 2:



Language Indicator as shown in Table 9.2-1, Language Indicator Value Assignments, in [Informative 1]

[5] This EF was originally labeled EF_{PL} in C.S0023-D v1.0 and previous revisions. The name has been changed to EF_{RUIM_PL} to avoid confusion with EF_{PL} under the MF [30].

3.4.27 EF_{SMS} (Short Messages)

This EF contains information in accordance with [8] comprising short messages (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: '6F3C'	Structure: linear fixed	Optional
Record Length: variable (1)	Update activity: high	

Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

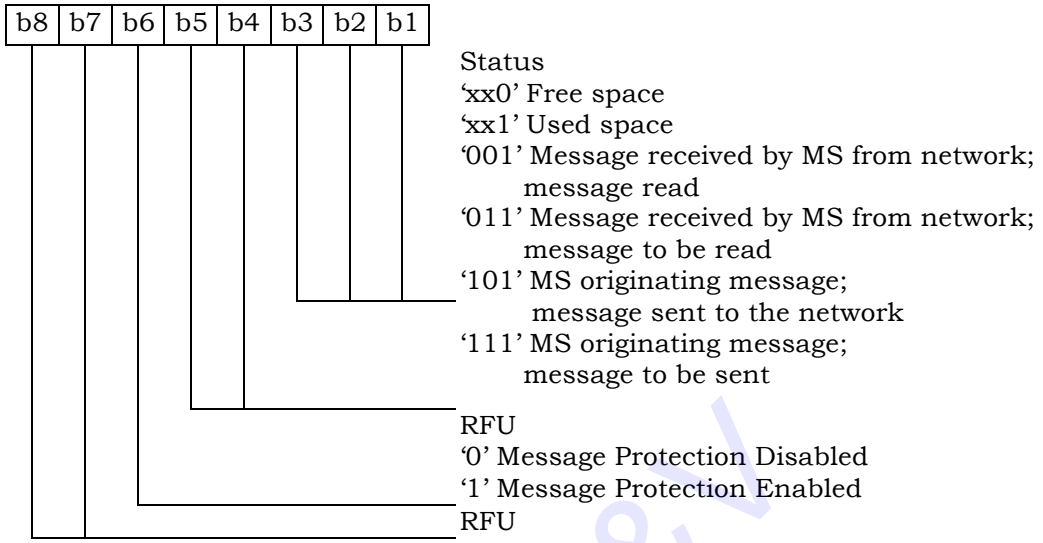
Bytes	Description	M/O	Length
1	Status	M	1 byte
2	MSG_LEN	M	1 byte
3 – 3+MSG_LEN	SMS Transport Layer Message	M	MSG_LEN bytes

Note: (1) The length and the byte allocations are variable according to the actual size of the SMS Transport Layer message. The maximum length is 255, which includes the length of the short message plus two bytes for storing "status" and "MSG_LEN".

- Status
  Status byte of the record which can be used as a pattern in the SEEK command. For MS originating messages sent to the network, the status shall be updated when the MS receives a status report or sends a successful SMS Command relating to the status report.

Coding:

Byte 1:



- MSG_LEN
  The length of the message not including MSG_LEN. Note that the definition of this EF does allow multiple occurrences of the segment, which consists of "PARAMETER_ID", "PARAMETER_LEN", and "Parameter Data" as described in [8]. The number of repetitions of the aforementioned segment is determined by MSG_LEN and the PARAMETER_LEN of each segment.

- SMS Transport Layer Message
  Contents: see Section 3.4.1 of [8].

3.4.28 EF_{SMSP} (Short Message Service Parameters)

If service n12 is allocated, this EF shall be present.

This EF contains values for Short Message Service Parameters (SMSP), which can be used by the Mobile Equipment (ME) for user assistance in preparation of mobile originated short messages.

The EF consists of one or more records, with each record able to hold a set of SMS parameters. The first (or only) record in the EF shall be used as a default set of parameters, if no other record is selected. To distinguish between records, a four-byte Teleservice Identifier as defined in [8] shall be included within each record. The SMS parameters stored within a record may be present or absent independently. When an SMS message is to be sent, the parameters in the R-UIM record that has the same Teleservice Identifier as the one in the mobile-originated message, if present, can be used by the ME when a value is not supplied by the user.

Identifier: '6F3D'	Structure: linear fixed	Optional
Record Length: variable (10+X)	Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 - 4	Teleservice Identifier	M	4 bytes
5 – 6	Parameter Indicators	M	2 bytes
7	Reserved	M	1 byte
8 – (8+N-1)	Destination Address	M	N (N>=1) [NOTE 1]
8+N	MSG_ENCODING	M	1 byte
9+N	Validity Period	M	1 byte
[NOTE 2]	Service Category	O	4 bytes
[NOTE 2]	Destination Subaddress	O	Variable [NOTE 2]
[NOTE 2]	Bearer Reply Option	O	3 bytes
[NOTE 2]	Bearer Data	O	Variable [NOTE 2]
[NOTE 2] [NOTE[3]	Padding	O	Variable [NOTE 2] [NOTE 3]

NOTE 1: N is 1 if the Parameter Indicators field indicates that the Destination Address is absent. Otherwise, N is the length of a valid destination address.

NOTE 2: Starting and ending bytes and length depend on the presence and absence of parameters indicated by the Parameter Indicators field.

NOTE 3: Padding is mandatory if the fields before it do not occupy all the 10+X bytes. Padding, if present, always ends at byte number 10+X.

Any bytes unused, due to parameters not requiring all of the bytes, or due to absent parameters, shall be set to 'FF'.

- Teleservice Identifier

  Contents:

  The supported teleservices include *IS-91 Extended Protocol Enhanced Services*, *Wireless Paging Teleservice*, *Wireless Messaging Teleservice*, *Voice Mail Notification* and *Wireless Application Protocol*. See 3.4.3.1 of [8] for details.

  Coding:

  4-byte Teleservice Identifier as defined in 3.4.3.1 of [8].

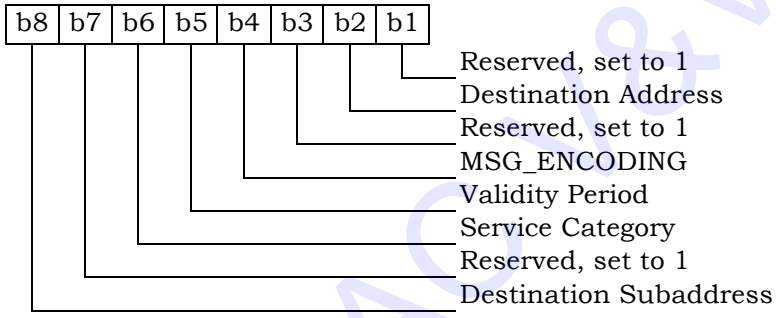
- Parameter Indicators

  Contents:

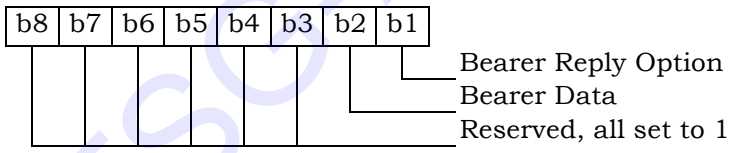
  Each of the default SMS parameters which can be stored in the remainder of the record are marked absent or present by individual bits within this byte.

  Coding:

  Byte 5:



  Byte 6:



Note: Bit value 0 means parameter present
Bit value 1 means parameter absent

- Reserved

  Set to 'FF'.

- Destination Address

  Contents and Coding:

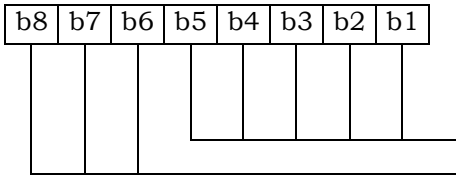
  If the Parameter Indicators field indicates this parameter is present, the contents and coding are defined in section 3.4.3.3 Address Parameters of [8]. It contains PARAMETER_ID, PARAMETER_LEN and parameter data.

  If the Parameter Indicators field indicates this parameter is absent, then it shall be set to 'FF' with a length of 1 byte.

- MSG_ENCODING
  Contents:

  If the Parameter Indicators field indicates this parameter is present, the contents and coding are defined in Table 9.1-1 Data Field Encoding Assignments of [Informative 1]. If Bearer Data is present and includes a Subparameter, that is, User Data or Service Category Program Data, which also includes a MSG_ENCODING field, then this parameter shall contain the same value.

  Coding:



  If the Parameter Indicators field indicates this field is absent, it shall be set to 'FF'.

- Validity Period
  Contents and Coding:

  If the Parameter Indicators field indicates this parameter is present, the contents and coding are defined in section 4.5.6 of [8] for the VALIDITY field of the relative time format. If Bearer Data is present and includes the Subparameter “Validity Period – Relative”, then this parameter shall contain the same value.

  If the Parameter Indicators field indicates this field is absent, it shall be set to 'FF'.

- Service Category
  Contents and Coding:.

  As defined in section 3.4.3.2 Service Category of [8]. It contains PARAMETER_ID, PARAMETER_LEN and parameter data.

- Destination Subaddress
  Contents and Coding:

  As defined in section 3.4.3.4 Subaddress of [8]. It contains PARAMETER_ID, PARAMETER_LEN and parameter data.

- Bearer Reply Option
  Contents and Coding:

  As defined in section 3.4.3.5 Bearer Reply Option of [8]. It contains PARAMETER_ID, PARAMETER_LEN and parameter data.

- Bearer Data
  Contents and Coding:

  As defined in section 3.4.3.7 Bearer Data of [8]. It contains PARAMETER_ID, PARAMETER_LEN and parameter data.

- Padding

  Contents and Coding:

  All bytes for this field shall be set to ‘FF’.

3.4.29 EF_{SMSS} (SMS Status)

This EF contains status information relating to the short message service.

The provision of this EF is associated with EF_{SMS}. Both files shall be present together or both shall be absent from the R-UIM.

Identifier: '6F3E'	Structure: transparent	Optional
File size: 5 + X bytes	Update activity: low	

Access Conditions:

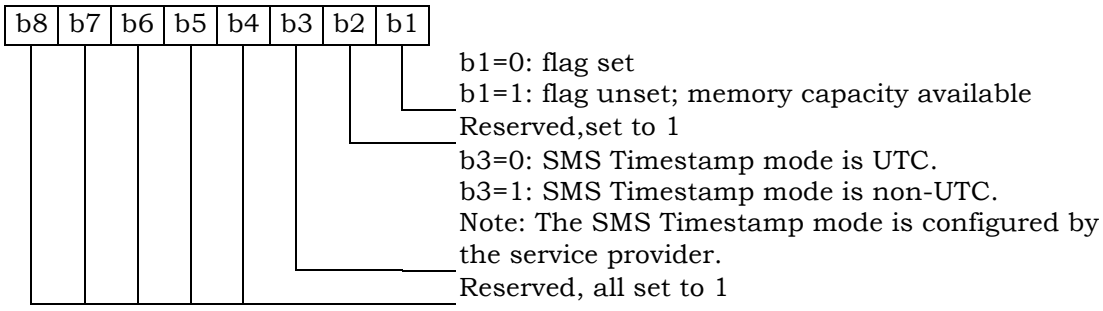
READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 – 2	MESSAGE_ID	M	2 bytes
3 – 4	WAP MESSAGE_ID	M	2 bytes
5	SMS "Memory Cap. Exceeded" Not. Flag/SMS Timestamp Mode	M	1 byte
6-5 + X	Reserved	O	X bytes

- MESSAGE_ID
  Contents: the value of the MESSAGE_ID in the last sent *SMS Submit Message* from a teleservice which requires message identifiers other than the WAP teleservice.
  Coding: as defined in [8].

- WAP MESSAGE_ID
  Contents: the value of the MESSAGE_ID in the last sent *SMS Submit Message* from the WAP teleservice.
  Coding: as defined in [8].

- SMS "Memory Capacity Exceeded" Notification Flag/SMS Timestamp Mode.
  Contents: Includes a flag that indicates whether or not there is memory capacity available to store SMS messages. Also includes a bit that indicates whether the SMS Timestamp mode is UTC or non-UTC.
  Coding:
  Byte 5:



3.4.30 EF_{SSFC} (Supplementary Services Feature Code Table)

This EF stores the numeric feature code to be used by the ME when a supplementary service is invoked in CDMA or analog mode via an implementation-dependant user interface (such as a menu) that automatically inserts a feature code into the dialed digit string. Because feature codes are service-provider specific, this EF is required to enable the ME to perform the mapping to the feature code.

When a supplementary service is invoked in CDMA or analog mode, the mobile station shall determine the feature code by reading the Supplementary Service Feature Code Table entry for the selected supplementary service, and pre-pending with asterisk.

Identifier: '6F3F'	Structure: transparent	Optional
File size: 2N+1	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	N, Number of Feature Codes	M	1 byte
2 – 3	Activate Call Delivery (CD)	M	2 bytes
4 – 5	De-activate Call Delivery (CD)	M	2 bytes
6 – 7	Register new Call Forwarding – Busy (CFB) forward-to number	M	2 bytes
8 – 9	Register Call Forwarding – Busy (CFB) to voice mail	M	2 bytes
10 – 11	De-register Call Forwarding – Busy (CFB)	M	2 bytes
12 – 13	Activate Call Forwarding – Busy (CFB)	M	2 bytes
14 – 15	De-activate Call Forwarding – Busy (CFB)	M	2 bytes
16 – 17	Register new Call Forwarding – Default (CFD) forward-to number	M	2 bytes
18 – 19	Register Call Forwarding – Default (CFD) to voice mail	M	2 bytes
20 – 21	De-register Call Forwarding – Default (CFD)	M	2 bytes
22 – 23	Activate Call Forwarding – Default (CFD)	M	2 bytes
24 – 25	De- activate Call Forwarding – Default (CFD)	M	2 bytes
26 – 27	Register new Call Forwarding – No Answer (CFNA) forward-to number	M	2 bytes
28 – 29	Register Call Forwarding – No Answer (CFNA) to voice mail	M	2 bytes
30 – 31	De-register Call Forwarding – No Answer (CFNA)	M	2 bytes
32 – 33	Activate Call Forwarding – No Answer (CFNA)	M	2 bytes
34 – 35	De-activate Call Forwarding – No Answer (CFNA)	M	2 bytes
36 – 37	Register new Call Forwarding – Unconditional (CFU) forward-to number	M	2 bytes
38 – 39	Register Call Forwarding – Unconditional (CFU) to voice mail	M	2 bytes
40 – 41	De-register Call Forwarding – Unconditional (CFU)	M	2 bytes

42 – 43	Activate Call Forwarding – Unconditional (CFU)	M	2 bytes
44 – 45	De-activate Call Forwarding – Unconditional (CFU)	M	2 bytes

Bytes	Description	M/O	Length
46 – 47	Activate Call Waiting (CW)	M	2 bytes
48 – 49	De-activate Call Waiting (CW)	M	2 bytes
50 – 51	Temporarily De-activate Call Waiting (Cancel Call Waiting - CCW)	M	2 bytes
52 – 53	Temporarily Activate Calling Number Identification Restriction (CNIR) (per-call blocking)	M	2 bytes
54 – 55	Temporarily De-activate Calling Number Identification Restriction (CNIR) (per-call allowed)	M	2 bytes
56 – 57	Invoke Conference Calling (CC)	M	2 bytes
58 – 59	Invoke Drop Last Conference Calling (CC) Party	M	2 bytes
60 – 61	Activate Do Not Disturb (DND)	M	2 bytes
62 – 63	De-activate Do Not Disturb (DND)	M	2 bytes
64 – 65	Activate Message Waiting Notification (MWN) Alert Pip Tone	M	2 bytes
66 – 67	De-activate Message Waiting Notification (MWN) Alert Pip Tone	M	2 bytes
68 – 69	Activate Message Waiting Notification (MWN) Pip Tone	M	2 bytes
70 – 71	De-activate Message Waiting Notification (MWN) Pip Tone	M	2 bytes
72 – 73	Temporarily De-activate Message Waiting Notification (MWN) Pip Tone (Cancel MWN - CMWN)	M	2 bytes
74 – 75	Invoke Priority Access and Channel Assignment (PACA)	M	2 bytes
76 – 77	Invoke Voice Message Retrieval (VMR)	M	2 bytes
78 – 79	Activate Calling Name Presentation (CNAP)	M	2 bytes
80 – 81	De-activate Calling Name Presentation (CNAP)	M	2 bytes
82 – 83	Activate Calling Name Restriction (CNAR)	M	2 bytes
84 – 85	De-activate Calling Name Restriction (CNAR)	M	2 bytes
86 – 87	Activate Automatic Callback (AC)	M	2 bytes
88 – 89	De-activate Automatic Callback (AC)	M	2 bytes
90 – 91	Activate Automatic Recall (AR)	M	2 bytes
92 – 93	De-activate Automatic Recall (AR)	M	2 bytes
94 – 95	Register new network registered User Selectable Call Forwarding (USCF) directory number	M	2 bytes
96 – 97	Activate Rejection of Undesired Annoying Calls (RUAC)	M	2 bytes
98 – 99	De-activate Rejection of Undesired Annoying Calls (RUAC)	M	2 bytes
100 – 101	Invoke Advice of Charge (AOC)	M	2 bytes
102 – 103	Invoke Call Trace (COT)	M	2 bytes
2N – 2N+1	FCN	M	2 bytes

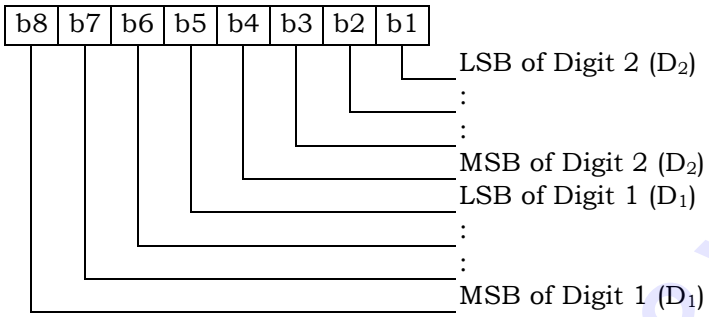
N, Number of Feature Codes" is coded in hexadecimal value, which indicates the number of feature codes.

A feature code of up to four digits shall be encoded via BCD into the two bytes of the feature code table entry as follows:

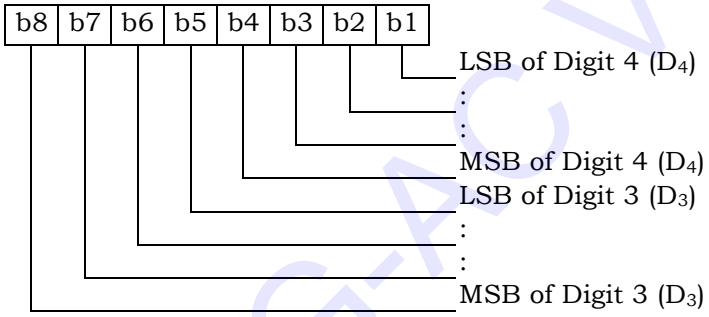
- represent these four digits as $D_1D_2D_3D_4$.
- if the feature code (FC) of less than four digits is used, the digits shall be right justified and the unused digits shall be set to 'F'.

Coding:

First byte:



Second byte:



3.4.31 EF_{SPN} (CDMA Home Service Provider Name)

If service n17 is allocated, this EF shall be present. This EF contains the home service provider name and appropriate requirements for display by the ME.

Identifier: '6F41'	Structure: transparent	Optional
File size: 35 bytes	Update activity: low	

Access Conditions:

READ	ALW
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

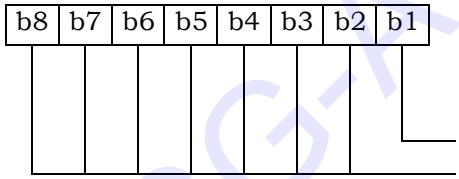
Bytes	Description	M/O	Length
1	Display Condition	M	1 byte
2	Character Encoding	M	1 byte
3	Language Indicator	M	1 byte
4 – 35	Service Provider Name	M	32 bytes

- Display Condition

  Contents: An indication of whether or not a service provider name shall be displayed by a MS which supports this feature when the MS is registered in the home service area.

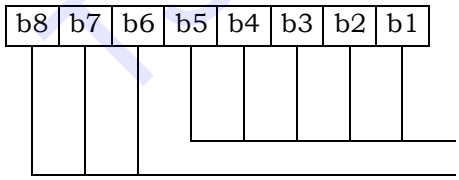
  Coding:

  Byte 1:



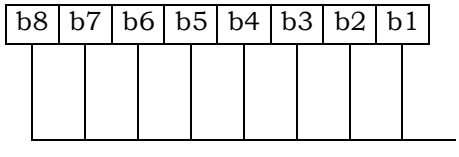
b1=0: display of registered system is not required
b1=1: registered system shall be displayed
RFU

  Byte 2:



CHARi encoding type as specified in Table 9.1-1, Data Field Encoding Assignments, in [Informative 1]
RFU

  Byte 3:



Language Indicator as specified in Table 9.2-1, Language Indicator Value Assignments, in [Informative 1]

  Bytes 4 – 35:

  - Service Provider Name

    Contents: service provider string to be displayed

Coding:

~~the~~ The string shall use SMS conventions as defined in Tables 9.1-1 and 9.2-1 of [Informative 1]. The string shall be stored in sequence with the first character in byte 4. Unused bytes shall be stored in the highest numbered bytes and shall be set to 'FF'.

If the string is coded as 7-bit, the SMS default 7 bit coded alphabet as referenced in [Informative 1] with bit 8 set to 0 shall be used.

3.4.32 EF_{USGIND} (Removable UIMID/SF_EUIMID Usage Indicator)

This EF indicates whether the UIMID or ESN_ME is used as the ESN value for CAVE authentication and MS identification, as per Section 4.6.1. This EF also indicates whether the SF_EUIMID or MEID_ME shall be used as the MEID field over the air when Service n8 is allocated and activated.

Identifier: '6F42'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

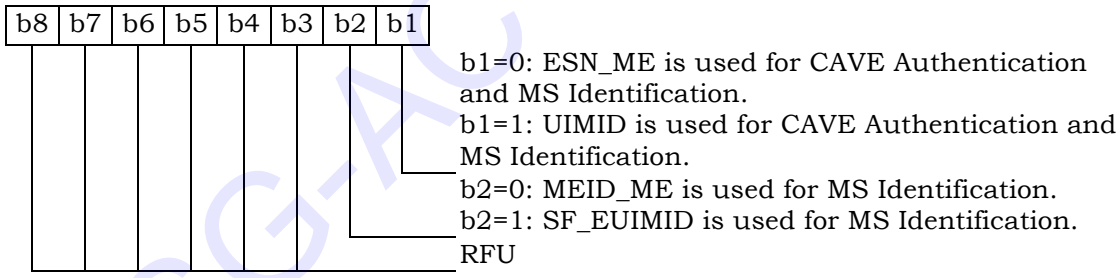
Bytes	Description	M/O	Length
1	UIMID/SF_EUIMID Usage Indicator	M	1 byte

Coding:

~~b1 is used as the UIMID usage indicator.~~

~~b2 is used as the SF_EUMID usage indicator.~~

Byte 1:



The ME shall interpret b2 only if the ME is assigned with an MEID_ME and service n8 is allocated and activated.

3.4.33 EF_{AD} (Administrative Data)

This EF contains information concerning the mode of operation according to the type of UIM. It also provides an indication whether some ME features should be activated during the normal operation.

Identifier: '6F43'	Structure: transparent	Mandatory
File size: 3+X bytes	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
2 – 3	Additional information	M	2 bytes
4 – 3+X	RFU	O	X bytes

- MS operation mode

  Contents: mode of operation for the MS.

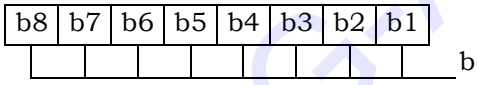
  Coding:

  Initial value

  - normal operation '00'

  Refer to [17] for other operational values.

  Byte 1:

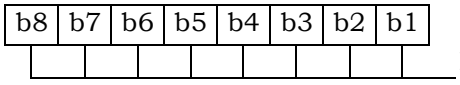
  b8 through b1= '00000000'.

- Additional information

  Coding:

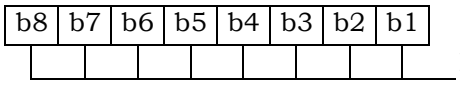
  - specific facilities (if b1=1 in byte 1);

  Byte 2: (first byte of additional information)

  RFU

  Byte 3:

  RFU

3.4.34 EF_{MDN} (Mobile Directory Number)

This EF stores the Mobile Directory Number, Type of Number, Numbering Plan, Presentation Indicator and Screening Indicator.

Identifier: '6F44'	Structure: linear fixed	Optional
Record length: 11 bytes	Update activity: low	

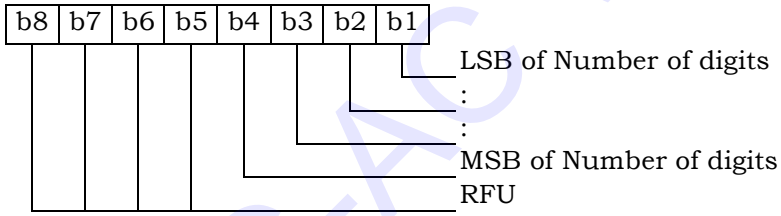
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Number of digits	M	1 byte
2 – 9	MDN	M	8 bytes
10	NUMBER_TYPE and NUMBER_PLAN	M	1 byte
11	PI and SI	M	1 byte

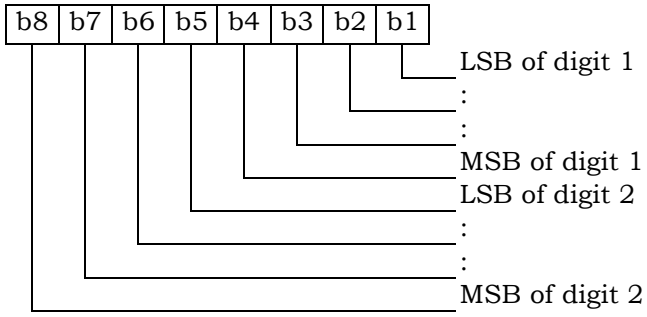
Coding:

Byte 1:



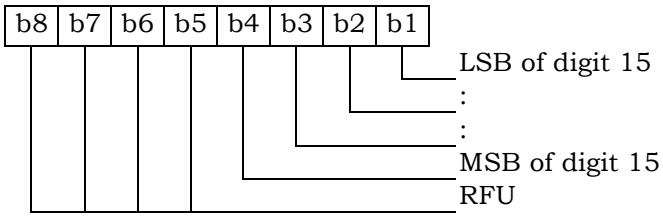
Byte 2 through 9 store MDN up to 15 digits described in Section 2.3.1.4 of [5] and Section 6.3.1.4 of [14]. Each digit shall be encoded according to Table 2.7.1.3.2.4-4 of [5] and Table 6.7.1.3.2.4-4 of [14]. If MDN requires less than 15 digits, excess nibbles at the end of data shall be set to 'F'.

Byte 2:

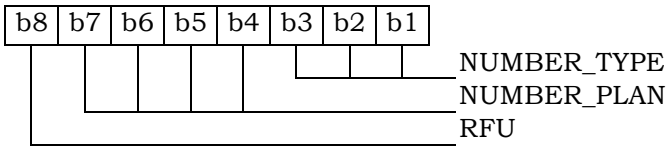


Bytes 3 through 8 shall follow the same format as Byte 2.

Byte 9:

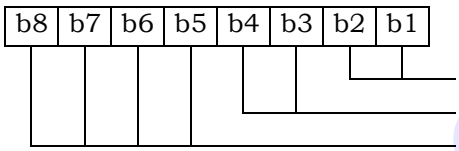


Byte 10:



Refer to Section 2.7.4.4 of [5] or Section 6.7.4.4 of [14].

Byte 11:



Refer to Section 2.7.4.4 of [5] or Section 6.7.4.4 of [14].

3.4.35 EF_{MAXPRL} (Maximum PRL)

This EF stores the maximum size, in octets, that the R-UIM can support for EF Preferred Roaming List and EF Extended Preferred Roaming List. See 3.5.3.1 and 3.5.3.3 of [7] for more detail.

Identifier: '6F45'	Structure: transparent	Mandatory
File size: 2 or 4 bytes	Update activity: Never	

Access Conditions:

READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 – 2	MAX_PR_LIST_SIZE for EF_{PRL}	M	2 bytes
3 – 4	MAX_PR_LIST_SIZE for EF_{EPRL}	O	2 bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

The 'MAX_PR_LIST_SIZE for EF_{EPRL}' field shall be included if EF_{EPRL} is present.

3.4.36 EF_{SPCS} (SPC Status)

This EF identifies whether the EF_{SPC} (Service programming code) is set to default and internally updated in the card to reflect the current state of SPC after an OTASP COMMIT if the SPC was changed. Details of SPC are in [7], section 3.3.6.

Identifier: '6F46'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	

Access Conditions:

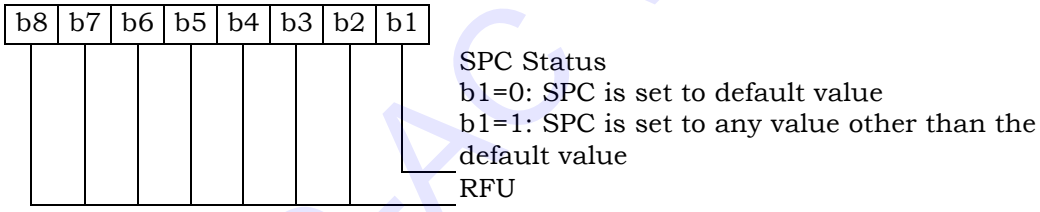
READ	CHV1
UPDATE	Never
INVALIDATE	Never
REHABILITATE	Never

Bytes	Description	M/O	Length
1	SPC Status	M	1 byte

- SPC Status

Coding:

Byte 1:



3.4.37 EF_{ECC} (Emergency Call Codes)

This EF contains up to 5 emergency call codes (ECCs).

Identifier: '6F47'	Structure: transparent	Optional
File size: 3n (n ≤ 5) bytes	Update activity: low	
Access Conditions: READ ALW UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 - 3	Emergency Call Code 1	O	3 bytes
4 - 6	Emergency Call Code 2	O	3 bytes
(3n-2) to 3n	Emergency Call Code n	O	3 bytes

- Emergency Call Code

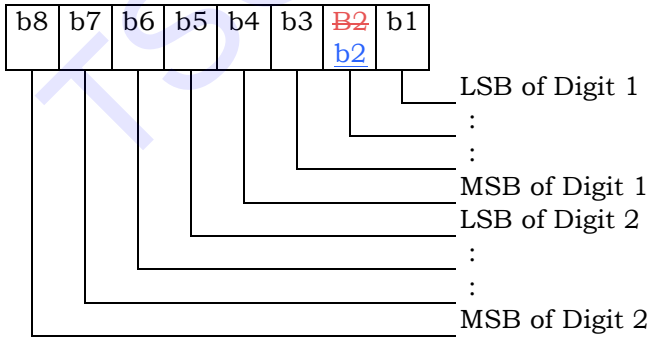
Contents:

Emergency Call Code. Each digit is encoded in BCD format.

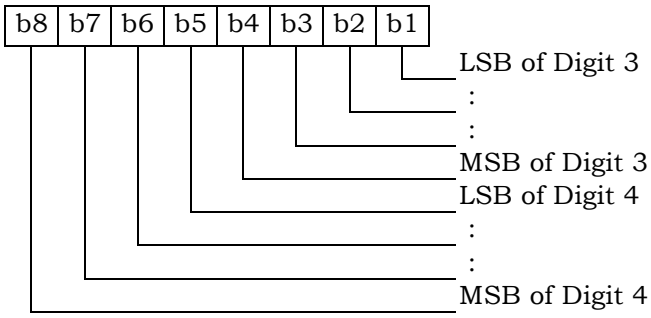
Coding:

The emergency call code is of a variable length with a maximum length of 6 digits. Each emergency call code is coded on three bytes, with each digit within the code being coded on four bits as shown below. If a code of less than 6 digits is chosen, then the unused nibbles shall be set to 'F'.

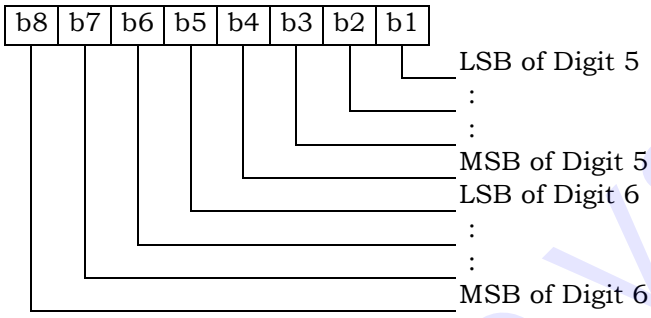
Byte 1:



Byte 2:



Byte 3:



After R-UIM activation, the ME selects the Dedicated File DF_{CDMA} and optionally attempts to select $EF_{ECC.}$ If EF_{ECC} is available, the ME requests the emergency call codes. If the user dials a number that matches one of the codes in EF_{ECC}, then the ME shall treat the call as an emergency call as specified in [5].

3.4.38 $EF_{ME3GPDOPC}$ (ME 3GPD Operation Capability)

If either service n20 or n38 is allocated (See Section 3.4.18), this EF shall be present. This EF stores IP operation capabilities supported by the ME.

Identifier: '6F48'	Structure: transparent	Optional
File size: 1 byte	Update activity: low	

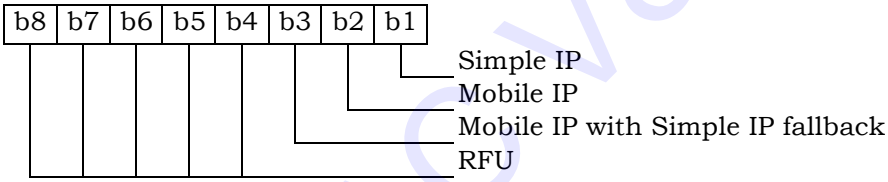
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	ME_3GPD_OP_~~MODE~~CAP	M	1 byte

Coding:

Byte 1:



After the selection of DF_{CDMA} (7F25) during the initialization, the R-UIM shall set the value of this byte to "0". Mobile equipment that supports Simple IP or Mobile IP shall set each subfield to '1' if it supports the corresponding operating ~~mode~~capability.

3.4.39 $EF_{3GPDOPM}$ (3GPD Operation Mode)

If either service n20 or n38 is allocated (See Section 3.4.18), this EF shall be present. This EF stores the 3GPD Operation Mode Parameter Block defined in [7].

Identifier: '6F49'	Structure: transparent	Optional
File size: 1 byte	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	See [7], 3GPD Operation Mode Parameter Block	M	1 byte

Coding:

Byte 1:

Note that the position of the bits differs from the location of the Operation Mode field in [7].

3.4.40 EF_{SIPCAP} (Simple IP Capability Parameters)

If service n20 is allocated (See Section 3.4.18), this EF shall be present. This EF stores the SimpleIP Capability Parameter Block defined in [7].

Identifier: '6F4A'	Structure: transparent	Optional
File size: 4 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 – 4	See [7], SimpleIP Capability Parameter Block	M	4 bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

3.4.41 EF_{MIPCAP} (Mobile IP Capability Parameters)

If service n38 is allocated (See Section 3.4.18), this EF shall be present. This EF stores the MobileIP Capability Parameter Block defined in [7].

Identifier: '6F4B'	Structure: transparent	Optional
File size: 5 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-5	See [7], MobileIP Capability Parameter Block	M	5 bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

3.4.42 EF_{SIPUPP} (Simple IP User Profile Parameters)

If service n20 is allocated (See Section 3.4.18), this EF shall be present. This EF stores the SimpleIP User Profile Parameter Block defined in [7].

Identifier: '6F4C'	Structure: transparent	Optional
File size: 1+X	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Length of SimpleIP User Profile Parameter Block	M	1 bytes
2 – X+1	See [7], SimpleIP User Profile Parameter Block	M	X bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

3.4.43 EF_{MIPUPP} (Mobile IP User Profile Parameters)

If service n38 is allocated (See Section 3.4.18), this EF shall be present. This EF stores the MobileIP User Profile Parameter Block defined in [7].

Identifier: '6F4D'	Structure: transparent	Optional
File size: 1+X	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Length of MobileIP User Profile Parameter Block	M	1 bytes
2 – X+1	See [7], MobileIP User Profile Parameter Block	M	X bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

3.4.44 EF_{SIPSP} (Simple IP Status Parameters)

If service n20 is allocated (See Section 3.4.18), this EF shall be present. This EF stores the SimpleIP Status Parameters Block defined in [7].

Identifier: '6F4E'	Structure: transparent	Optional
File size: 1	Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	See [7], SimpleIP Status Parameters Block	M	1 byte

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

3.4.45 EF_{MIPSP} (Mobile IP Status Parameters)

If service n38 is allocated (See Section 3.4.18), this EF shall be present. This EF stores the MobileIP Status Parameters Block defined in [7].

Identifier: '6F4F'	Structure: transparent	Optional
File size: X	Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 – X	See [7], MobileIP Status Parameters Block	M	X bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

3.4.46 $EF_{SIPPAPSS}$ (Simple IP PAP SS Parameters)

If service n20 is allocated (See Section 3.4.18), this EF shall be present. This EF stores the SimpleIP PAP SS Parameter Block defined in [7].

Identifier: '6F50'	Structure: transparent	Optional
File size: 1+X	Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Length of SimpleIP PAP SS Parameter Block	M	1 bytes
2 – X+1	See [7], SimpleIP PAP SS Parameter Block	M	X bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

3.4.47 Reserved

3.4.48 Reserved

3.4.49 EF_{PUZL} (Preferred User Zone List)

This EF stores the Preferred User Zone List, as described in Section 3.5.7 of [7].

Identifier: '6F53'	Structure: transparent	Optional
File size: 'MAX_UZ_LIST_SIZE'	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-CUR_UZ_LIST_SIZE	PUZL (see Section 3.5.7 of [7])	M	CUR_UZ_LIST_SIZE

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

3.4.50 $EF_{MAXPUZL}$ (Maximum PUZL)

This EF stores the maximum size, in octets, that the R-UIM can support for EF Preferred User Zone List (See 3.5.7 of [7] for more details) and the maximum number of User Zone entries that the R-UIM can support for EF_{PUZL} (See 3.5.6.1 of [7] for more details).

Identifier: '6F54'	Structure: transparent	Optional
File size: 5 bytes	Update activity: Never	

Access Conditions:

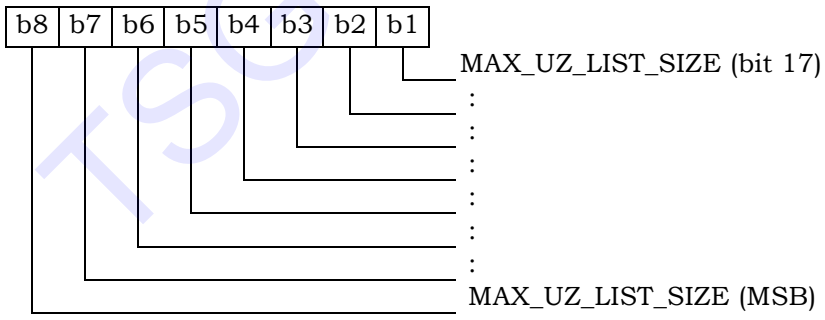
READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 –3	MAX_UZ_LIST_SIZE	M	3 bytes
4 - 5	MAX_NUM_UZ	M	2 bytes

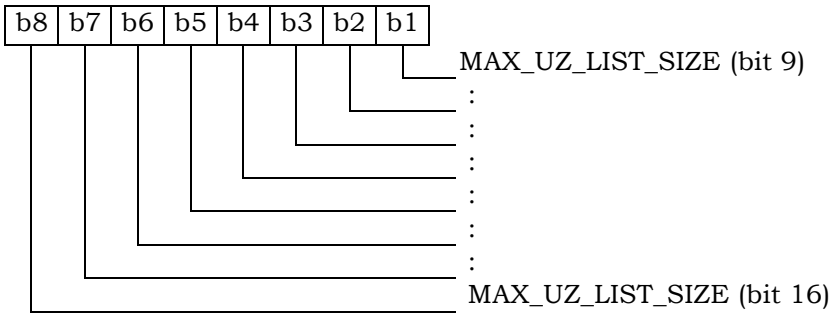
This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

Coding:

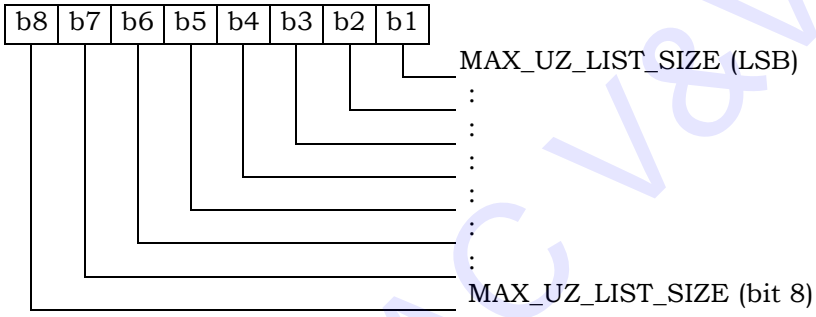
Octet 1:



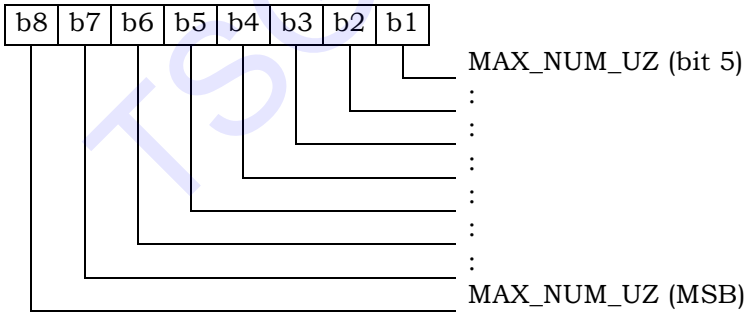
Octet 2:



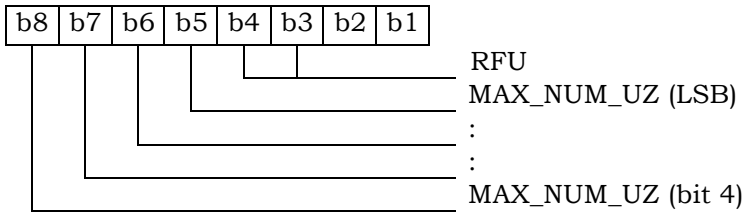
Octet 3:



Octet 4:



Octet 5:



3.4.51 EF_{MECRP} (ME-specific Configuration Request Parameters)

This EF stores ME-specific parameters to be used to form the response to the CONFIGURATION REQUEST command while secure mode is active. The ME shall update these ME-specific parameters during initializations.

Identifier: '6F55'	Structure: transparent	Mandatory
File size: 3 bytes	Update activity: low	

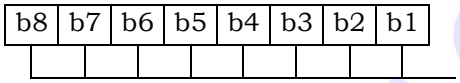
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	SCM	M	1 byte
2	MOB_P_REV	M	1 byte
3	Local Control	M	1 byte

Coding:

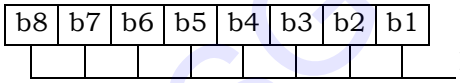
Byte 1:



SCM (Station Class Mark) [5]

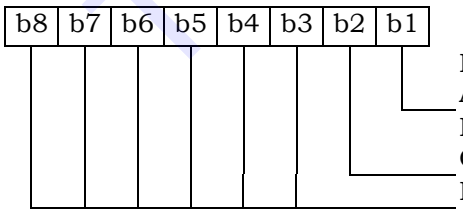
Note: b6 indicates if the ME is operating in slotted mode.

Byte 2:



MOB_P_REV

Byte 3:



LOCAL_CONTROL_ANALOG (Local Control for Analog Operation – Section 3.5.2.1 or 4.5.2.1 of [7])

LOCAL_CONTROL_CDMA (Local Control for CDMA Operation - Section 3.5.2.3 or 4.5.2.3 of [7])

RFU

3.4.52 $EF_{HRPDCAP}$ (HRPD Access Authentication Capability Parameters)

If service n5 is allocated (See Section 3.4.18), this EF shall be present. This EF stores the HRPD Access Authentication Capability Parameters Block defined in Section 3.5.8.12 of [7].

Identifier: '6F56'	Structure: transparent	Optional
File size: 3 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 – 3	See [7], HRPD Access Authentication Capability Parameters Block	M	3 bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

3.4.53 $EF_{HRPDUPP}$ (HRPD Access Authentication User Profile Parameters)

If service n5 is allocated (See Section 3.4.18), this EF shall be present. This EF stores the HRPD Access Authentication User Profile Parameters Block defined in Section 3.5.8.13 of [7].

Identifier: '6F57'	Structure: transparent	Optional
File size: 1+X bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Length of HRPD Access Authentication User Profile Parameters Block	M	1 byte
2 – X+1	See [7], HRPD Access Authentication User Profile Parameters Block	M	X bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

3.4.54 EF_{CSSPR} (CUR_SSPR_P_REV)

This EF stores the protocol revision (CUR_SSPR_P_REV) of the current extended preferred roaming list stored in the EF_{EPRL}. This information, described in section 3.5.3.3 of [7], is used by the ME to parse the EF_{EPRL}.

Identifier: '6F58'	Structure: transparent	Optional
File size: 1	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	CUR_SSPR_P_REV	M	1 byte

Notes:

1. It is recommended that CUR_SSPR_P_REV in Octet 7 of EF_{EPRL} (as defined in section 3.5.3.3 of [7]) be used instead of this EF_{CSSPR}.
2. According to [7], CUR_SSPR_P_REV is used to indicate if the PRL or EPRL is stored in PR_LISTs-p and according to section 3.3.1.3 of [7], the MS shall store CUR_SSPR_P_REV for not only the PRL but also the EPRL after an SSPR Download Request. However, since an R-UIM can store the PRL and EPRL in EF_{PRL} and EF_{EPRL}, respectively, there is no need to distinguish what is stored in EF_{PRL}. Hence, EF_{CSSPR} is only applicable for EF_{EPRL} and not EF_{PRL} as [7] would seem to require.

3.4.55 EF_{ATC} (Access Terminal Class)

If service n5 is allocated (See Section 3.4.18), this EF shall be present. This EF stores the class of access terminal used for Persistence Test in the system defined in [28].

Identifier: '6F59'	Structure: transparent	Optional
File size: 1	Update activity: low	

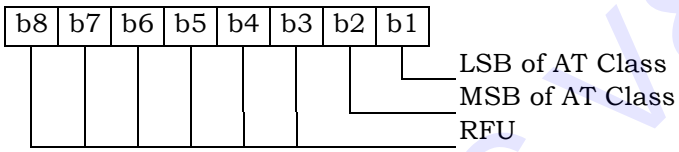
Access Conditions:

READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Access Terminal Class	M	1 byte

Coding:

Byte 1:



3.4.56 EF_{EPRL} (Extended Preferred Roaming List)

This EF stores the Extended Preferred Roaming List, as described in Section 3.5.5 of [7]. The Preferred Roaming List includes selection parameters from [5] and [14], Annex F.

Identifier: '6F5A'	Structure: transparent	Optional
File size: 'MAX_PR_LIST_SIZE for EF_{EPRL}'	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-PR_LIST_SIZE	PR_LIST	M	PR_LIST_SIZE

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

- PR_LIST

  Contents:

  The Extended Preferred Roaming List.

  Coding:

  As defined in section 3.5.5 of [7].

3.4.57 $EF_{BCSMScfg}$ (Broadcast Short Message Configuration)

If service n14 is allocated, this EF shall be present.

This EF contains the operator broadcast configuration setting for Broadcast SMS. This information, determined by the operator, defines the filtering criteria that can be used by the Mobile Equipment (ME) to receive Broadcast SMS.

Identifier: '6F5B'	Structure: transparent	Optional
File size: 1 byte	Update activity: low	

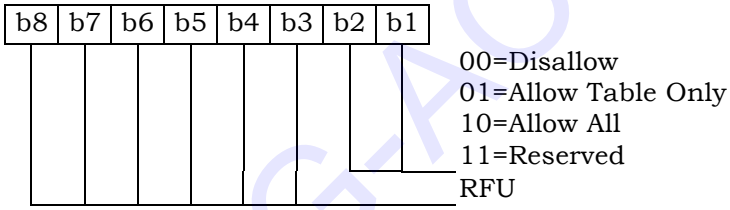
Access Conditions:

READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Operator Broadcast Configuration	M	1 byte

Coding:

Byte 1:



Operator configuration includes filtering criteria imposed by a service provider.

Field Name	Description
Disallow	This setting disables the mobile station's broadcast SMS capability (i.e., the mobile station will not process broadcast SMS).
Allow Table Only	This setting allows the mobile station to receive only broadcast messages for the service categories that have been programmed in $EF_{BCSMStable}$.
Allow All	This setting allows the mobile station to receive broadcast messages for all service categories.

3.4.58 $EF_{BCSMSpref}$ (Broadcast Short Message Preference)

If service n14 is allocated, this EF shall be present.

This EF contains the user broadcast configuration setting for Broadcast SMS. This information, determined by the user, defines the filtering criteria that can be used by the Mobile Equipment (ME) to receive Broadcast SMS.

Identifier: '6F5C'	Structure: transparent	Optional
File size: 1 byte	Update activity: high	

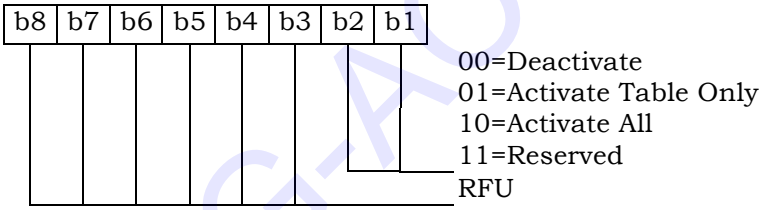
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	User Broadcast Configuration	M	1 byte

Coding:

Byte 1:



User configuration includes filtering criteria determined by the mobile user.

Field Name	Description
Deactivate	This setting deactivates the mobile station's broadcast SMS functions (i.e., the mobile station will not process broadcast SMS).
Activate Table Only	This setting allows the mobile station to receive only broadcast messages for the service categories that have been programmed in $EF_{BCSMStable}$, subject to any additional filtering criteria included in $EF_{BCSMStable}$ based on user preferences. This setting is only valid if the operator configuration is not Disallow. Moreover, the mobile user can selectively enable and disable individual programmed entries in $EF_{BCSMStable}$.

Field Name	Description
Activate All	Activate All This setting allows the mobile station to receive broadcast messages for all service categories. This setting is only valid if the operator configuration is “Allow All”. $EF_{BCSMStable}$ will not be consulted for this setting.

1

3.4.59 $EF_{BCSMStable}$ (Broadcast Short Message Table)

If service n14 is allocated, this EF shall be present.

This EF contains information in accordance with [8] comprising service category program parameters, which can be used by the Mobile Equipment (ME) for Broadcast SMS filtering. See Section 4.5.19 of [8] for more detail.

Each record in this EF is linked to a record with the same record index in EF_{BCSMSP}.

Identifier: '6F5D'	Structure: linear fixed	Optional
Record Length: 7+X byte	Update activity: high	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Status	M	1 byte
2 – 3	Service Category	M	2 bytes
4	Language	M	1 byte
5	Max Messages	M	1 byte
6	Alert Option	M	1 byte
7	Label Encoding	M	1 byte
8 to 7+X	Label	M	X byte

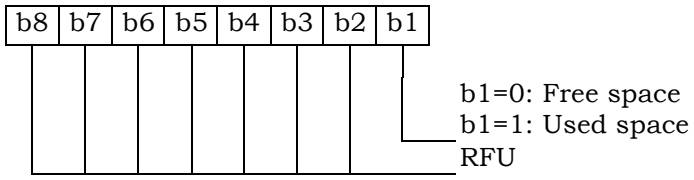
- Status

Contents:

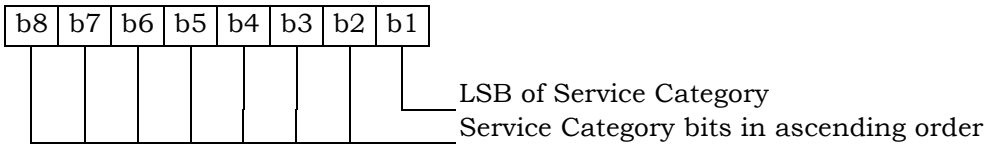
Status byte of the record which can be used as a pattern in the SEEK command.

Coding:

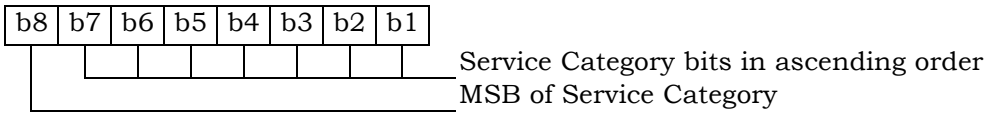
Byte 1:



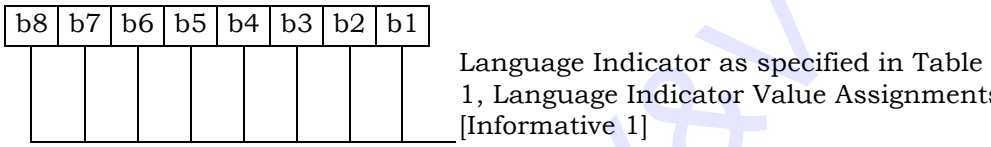
Byte 2:



Byte 3:

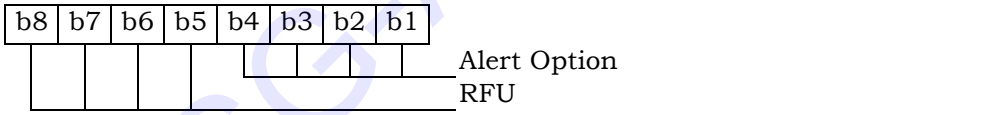


Byte 4:

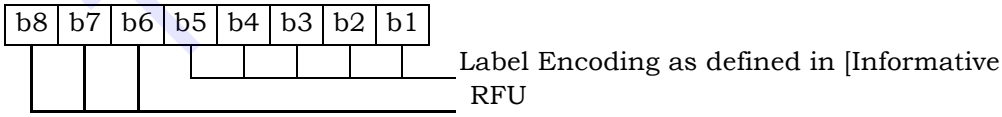


Byte 5:

Byte 6:



Byte 7:



3.4.60 EF_{BCSMSP} (Broadcast Short Message Parameter)

If service n14 is allocated, this EF shall be present.

This EF contains selection flag and priority associated with service categories and used by the ME for filtering of BC-SMS. Each record in this EF is linked to a record with the same record index in $EF_{BCSMStable}$.

Identifier: '6F5E'	Structure: linear fixed	Optional
Record Length: 2 bytes	Update activity: high	

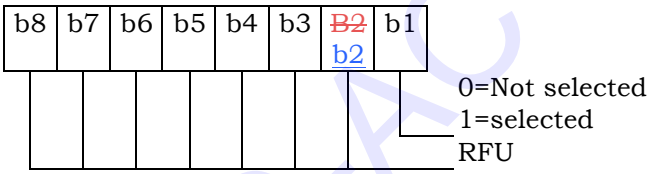
Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

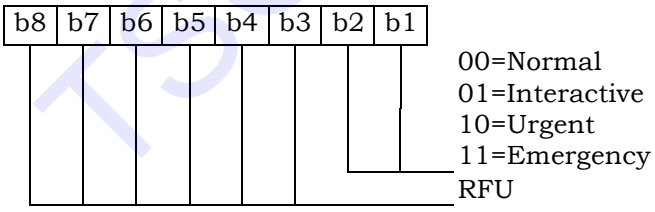
Bytes	Description	M/O	Length
1	Select	M	1 byte
2	Priority	M	1 byte

Coding:

Byte 1:



Byte 2:



Unused records are filled with 'FF'. When the b1 of Byte 1 is set to '1', then the ME shall filter the BC-SMS according to the priority indicated in Byte 2.

3.4.61 EF_{IMPI} (IMS private user identity)

If service n7 is allocated, this EF shall be present.

This EF contains the private user identity of the user [31].

Identifier: '6F5F'	Structure: transparent	Optional
File size: X bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM DEACTIVATE ADM ACTIVATE ADM		

Bytes	Description	M/O	Length
1 to X	NAI TLV data object	M	X bytes

- NAI

Contents:

- Private user identity of the user.

Coding:

- For contents and syntax of NAI TLV data object values see [34]. The NAI shall be encoded to an octet string according to UTF-8 encoding rules as specified in [46]. The tag value of the NAI TLV data objects shall be '80'.

3.4.62 EF_{DOMAIN} (Home Network Domain Name)

If service n7 is allocated, this EF shall be present.

This EF contains the home operator's network domain name SIP URI [31].

Identifier: '6F60'	Structure: transparent	Optional
File size: X bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM DEACTIVATE ADM ACTIVATE ADM		

Bytes	Description	M/O	Length
1 to X	URI TLV data object	M	X bytes

- URI

Contents:

-Home Network Domain Name SIP URI.

Coding:

-For contents and syntax of URI TLV data object values see [33]. The URI shall be encoded to an octet string according to UTF-8 encoding rules as specified in [46]. The tag value of the URI TLV data objects shall be '80'.

3.4.63 EF_{IMPU} (IMS public user identity)

If service n7 is allocated, this EF shall be present.

This EF contains values for public SIP Identities (SIP URI) of the user [31].

The EF consists of one or more records, with each record able to hold a set of public user identities.

Identifier: '6F61'	Structure: linear fixed	Optional
Record length: X bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM DEACTIVATE ADM ACTIVATE ADM		

Bytes	Description	M/O	Length
1 to X	URI TLV data object	M	X bytes

- URI

Contents:

- Public user identity by which other parties know the subscriber, in the format of SIP URL, tel URL, or both.

Coding:

-For contents and syntax of URI TLV data object values see [33]. The URI shall be encoded to an octet string according to UTF-8 encoding rules as specified in [46]. The tag value of the URI TLV data objects shall be ‘80’.

3.4.64 EF_{PCSCF} (Proxy Call Session Control Function)

If service n7 is allocated, this EF shall be present.

This EF contains one or more Proxy Call Session Control Function addresses [31]. The first record in the EF shall be considered to be of the highest priority. The last record in the EF shall be considered to be the lowest priority.

Identifier: '6F62'	Structure: linear fixed	Optional
Record length: X bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	ADM
DEACTIVATE	ADM
ACTIVATE	ADM

Bytes	Description	M/O	Length
1 to X	P-CSCF TLV data object	M	X bytes

- P-CSCF

Contents:

- Address of Proxy Call Session Control Function, in the format of FQDN, an IPv4 address, or an IPv6 address.

Coding:

- The tag value of this P-CSCF TLV data objects shall be '80'. The format of the data object is as follows:

Field	Length (bytes)
Tag	1
Length	2

Address Type	1
Address Length	1
P-CSCF Address	Address Length

Address Type: Type of the P-CSCF address.
This field shall be set to the type of the P-CSCF address according to the following:

Value	Name
00000000	FQDN
00000001	Ipv4
00000010	Ipv6
Reserved	Reserved

Address Length: Length of the P-CSCF address
This field shall be set to the length of the P-CSCF address, in units of byte.

P-CSCF Address: Address of the Proxy Call Session Control Function
This field shall be set to the address of the Proxy Call Session Control

Function. When the P-SCSF type is set to 0x00, the corresponding P-CSCF Address shall be encoded to an octet string according to UTF-8 encoding rules as specified in [46].

3.4.65 $EF_{BAKPARA}$ (Currently used BAK Parameters)

If service n39 is allocated, this EF shall be present.

This EF contains the triple (BCMCS_Flow_ID, BAK_ID, BAK_Expire) corresponding to BAK keys that have been delivered to the R-UIM and are currently used. See [36] for more details.

Identifier: '6F63'	Structure: Linear Fixed	Optional
Record length: X+Y+Z+3 bytes	Update activity: high	

Access Conditions:

READ	CHV1
UPDATE	ADM
DEACTIVATE	ADM
ACTIVATE	ADM

Bytes	Description	M/O	Length
1	Length of BCMCS_Flow_ID	M	1 byte
2 to X +1	BCMCS_Flow_ID	M	X bytes
X+2	Length of BAK_ID	M	1 byte
X+3 to X+Y+2	BAK_ID	M	Y bytes
X+Y+3	Length of BAK_Expire	M	1 byte
X+Y+4 to X+Y+Z+3	BAK_Expire	M	Z bytes

- Length of BCMCS_Flow_ID

  Content: number of bytes of the following data item containing the BCMCS flow identifier.

  Coding: Binary.

- BCMCS_Flow_ID

  Content: BCMCS Flow Identifier

  Coding: Binary.

- Length of BAK_ID

  Content: number of bytes of the following data item containing the BAK identifier.

  Coding: Binary

- BAK_ID

  Content: BAK Identifier

  Coding: Binary.

- Length of BAK_Expire

  Content: number of bytes of the following data item containing the BAK_Expire.

  Coding: Binary

- BAK_Expire

  Content: BAK_Expire

  Coding: Binary.

3.4.66 $EF_{UpBAKPARA}$ (Updated BAK Parameters)

If service n39 is allocated, this EF shall be present.

This EF contains the triple (BCMCS_Flow_ID, BAK_ID, BAK_Expire) corresponding to BAK keys that have been delivered to the R-UIM but have not yet been used. See [36] for more details.

Identifier: '6F64'	Structure: cyclic	Optional
Record length: X+Y+Z+3 bytes	Update activity: high	

Access Conditions:

READ	CHV1
UPDATE	ADM
DEACTIVATE	ADM
ACTIVATE	ADM

Bytes	Description	M/O	Length
1	Length of BCMCS_Flow_ID	M	1 byte
2 to X +1	BCMCS_Flow_ID	M	X bytes
X+2	Length of BAK_ID	M	1 byte
X+3 to X+2+Y	BAK_ID	M	Y bytes
X+Y+3	Length of BAK_Expire	M	1 byte
X+Y+4 to X+Y+Z+3	BAK_Expire	M	Z bytes

Length of BCMCS_Flow_ID

Content: number of bytes of the following data item containing the BCMCS flow identifier.

Coding: Binary

BCMCS_Flow_ID

Content: BCMCS Flow Identifier

Coding: Binary.

Length of BAK_ID

Content: number of bytes of the following data item containing the BAK identifier.

Coding: Binary

BAK_ID

Content: BAK Identifier

Coding: Binary.

Length of BAK_Expire

Content: number of bytes of the following data item containing the BAK_Expire.

Coding: Binary

BAK_Expire

Content: BAK_Expire

Coding: Binary.

3.4.67 EF_{MMSN} (MMS Notification)

If service n40 is allocated, this file shall be present.

This EF contains information in accordance with [37] comprising MMS notifications (and associated parameters) which have been received by the ME from the network.

Identifier: '6F65'	Structure: Linear fixed	Optional
Record length: X bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 - 2	MMS Status	M	2 bytes
3	MMS Implementation	M	1 byte
4 to Y+3	MMS Notification	M	Y bytes
Y+4	Extension file record number	M	1 byte

Note: X ≥ Y+4 for every record.

- MMS Status

Content:

-The status bytes contain the status information for the notification.

Octet 1:

b8	b7	b6	b5	b4	b3	b2	b1	
X	X	X					0	Unused record
X	X	X					1	Record in use
X	X					0	1	Notification not read
X	X					1	1	Notification read
0	0	X					1	MM not retrieved
0	1	X					1	MM retrieved
1	0	X					1	MM rejected
1	1	X					1	MM forwarded
~~Reserved for future use~~RFU								

Octet 2:

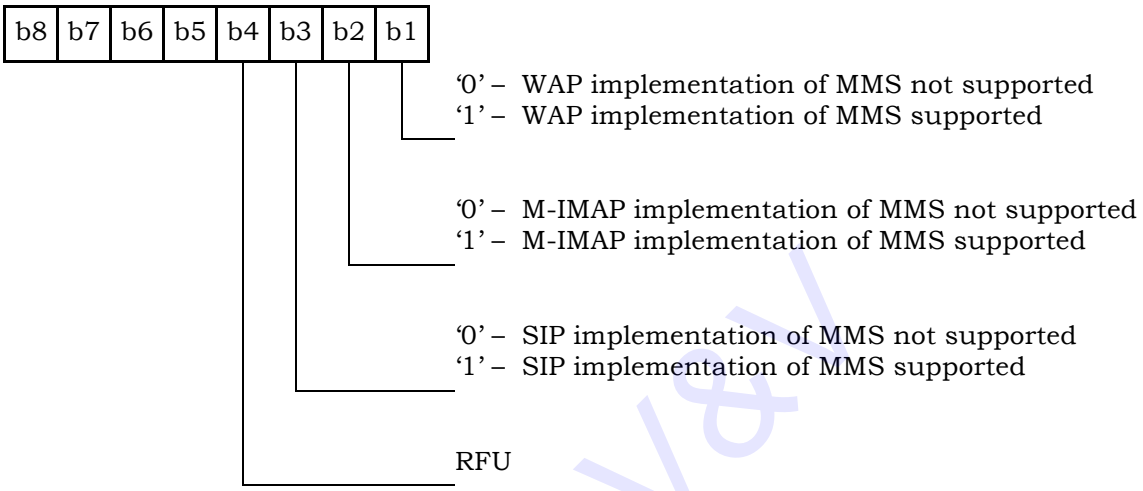
b8	b7	b6	b5	b4	b3	b2	b1	
								~~Reserved for future use~~RFU

- MMS Implementation

Contents:

-The MMS Implementation indicates the used implementation type, e.g. WAP, M-IMAP, SIP.

Octet 3:



- MMS Notification

Contents:

- The MMS Notification contains the MMS notification.

Coding:

- The MMS Notification is coded according to the MMS Implementation as indicated in octet 3.
- Any unused octets shall be set to 'FF'.

- Extension file record number

Contents:

~~e~~Extension file record number. This octet identifies the number of a record in the EF_{EXT8} containing extension data for the notification information. The use of this octet is optional. If it is not used it shall be set to 'FF'.

Coding:

~~b~~Binary.

3.4.68 EF_{EXT8} (Extension 8)

If service n41 is allocated, this file shall be present.

This EF contains extension data of a MMS Notification (Multimedia Messaging Service).

Identifier: '6F66'	Structure: linear fixed	Optional
Record length: X+2 bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Record type	M	1 byte
2 to X+1	Extension data	M	X bytes
X+2	Identifier	M	1 byte

For contents and coding see [30].

3.4.69 EF_{MMSICP} (MMS Issuer Connectivity Parameters)

If service n40 is allocated, this file shall be present.

This EF contains values for Multimedia Messaging Connectivity Parameters as determined by the issuer, which can be used by the ME for MMS network connection. This file may contain one or more sets of Multimedia Messaging Issuer Connectivity Parameters. The first set of Multimedia Messaging Issuer Connectivity Parameters is used as the default set. Each set of Multimedia Messaging Issuer Connectivity Parameters may consist of one or more Interface to Core Network and Bearer information TLV objects (only for WAP), but shall contain only one MMS implementation TLV object (for WAP, M-IMAP and SIP), one MMS Relay/Server TLV object (for WAP, M-IMAP and SIP) and one Gateway TLV object (only for WAP). The order of the Interface to Core Network and Bearer information TLV objects in the MMS Connectivity TLV object defines the priority of the Interface to Core Network and Bearer information, with the first TLV object having the highest priority.

Identifier: '6F67'	Structure: Transparent	Optional
File Size: X_1+…+ X_n bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 to X_1	MMS Connectivity Parameters TLV object	M	X_1 bytes
X_1+1 to X_1 + X_2	MMS Connectivity Parameters TLV object	O	X_2 bytes
…	…		
X_1+…+ X_{n-1}+1 to X_1+…+ X_n	MMS Connectivity Parameters TLV object	O	X_n bytes

- **MMS Connectivity Parameters tags**

Description	Tag Value
MMS Connectivity Parameters Tag	'AB'
MMS Implementation Tag	‘80’
MMS Relay/Server Tag	‘81’
Interface to Core Network and Bearer Information Tag	'82'
Gateway Tag	'83'
MMS Authentication Mechanism Tag	‘84’
MMS Authentication ID Tag	‘85’

- **MMS Connectivity Parameters contents**

Description	Value	M/O	Length (bytes)
MMS Connectivity Parameters Tag	'AB'	M	1
Length	Note 1	M	Note 2
MMS Implementation Tag	'80'	M	1
Length	1	M	1
MMS Implementation Information	--	M	1
MMS Relay/Server Tag	'81'	M	1
Length	X	M	Note 2
MMS Relay/Server Address	--	M	X
First Interface to Core Network and Bearer Information Tag (highest priority)	'82'	C2	1
Length	Y1	C2	Note 2
First Interface to Core Network and Bearer information	--	C2	Y1
Second Interface to Core Network and Bearer Information Tag	'82'	C2	1
Length	Y2	C2	Note 2
Second Interface to Core Network and Bearer information	--	C2	Y2
…			
Nth Interface to Core Network and Bearer Information Tag (lowest priority)	'82'	C2	1
Length	Y3	C2	Note 2
Nth Interface to Core Network and Bearer information	--	C2	Y3
Gateway Tag	'83'	O	1
Length	Z	O	Note 2
Gateway Information	--	O	Z
MMS Authentication Mechanism Tag	'84'	C1	1
Length	X	C1	Note 2
MMS Authentication Mechanism	--	C1	X
MMS Authentication ID Tag	'85'	C1	1
Length	X	C1	Note 2
MMS Authentication ID (Login_ID)	--	C1	X

NOTE 1: This is the total size of the constructed TLV object (not including the tag and this length).
NOTE 2: The length is coded according to [49] using primitive encoding and the minimum number of octets.
C1: only present if M-IMAP or SIP indicated in tag 80
C2: only present if WAP is indicated in tag 80

- **MMS Implementation Tag '80'**

See [30] for contents and coding.

- **MMS Relay/server Tag '81'**

  Contents:
  - The MMS relay/server contains the address of the associated MMS relay/server; In addition, for M-IMAP and SIP, authentication mechanism and authentication ID (Login ID) are also included.

  Coding:
  - The MMS relay/server address is coded as URI appropriate to the MM1 implementation being used, for example SIP, or M-IMAP.

- **Interface to Core Network and Bearer Information Tag '82'**

  Contents:
  - The Interface to Core Network and Bearer Information may contain the following information to set up the bearer: Bearer, Address, Type of address, Speed, Call type, Authentication type, Authentication id, Authentication password.

  Coding:
  - The coding is according to the guideline provided in [37]. If MMS implementation type is WAP, all instances of Interface to Core Network and Bearer Information are optional. If MMS implementation type is M-IMAP or SIP, no Interface to Core Network and Bearer Information is needed.

- **Gateway Tag '83'**

  Contents:
  - The Gateway may contain the following information; Address, Type of address, Port, Service, Authentication type, Authentication id and Authentication password.

  Coding:
  - The coding is according to the guideline provided in [37].

- **MMS Authentication Mechanism Tag ‘84’**

  Contents:

  - The MMS authentication mechanism contains the authentication mechanism for MMS. It is mandatory for M-IMAP and SIP.

  Coding:
  - The MMS authentication mechanism is coded as Table 10.

- **MMS Authentication ID Tag ‘85’**

  Contents:

  - The MMS authentication ID contains the authentication ID for MMS. It is mandatory for M-IMAP and SIP.

  Coding:
  - The coding is according to the guideline provided in [37].

Unused bytes shall be set to 'FF'.

3.4.70 EF_{MMSUP} (MMS User Preferences)

If service n40 is allocated, this file shall be present.

This EF contains values for Multimedia Messaging Service User Preferences, which can be used by the ME for user assistance in preparation of mobile multimedia messages (e.g. default values for parameters that are often used).

Identifier: '6F68'	Structure: Linear Fixed	Optional
Record Length: X bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 to X	MMS User Preference TLV Objects	M	X bytes

- **MMS User Preference tags**

Description	Tag Value
MMS Implementation Tag	'80'
MMS User preference profile name Tag	'81'
MMS User Preference information Tag	'82'

- **MMS User Preference TLV Objects**

Description	Value	M/O	Length (bytes)
MMS Implementation Tag	'80'	M	1
Length	1	M	1
MMS Implementation information	--	M	1
MMS User preference profile name Tag	'81'	M	1
Length	Y	M	Note
MMS User profile name	--	M	Y
MMS User Preference information Tag	'82'	M	1
Length	Z	M	Note
MMS User Preference information	--	M	Z
NOTE: The length is coded according to [49] using primitive encoding and the minimum number of octets.			

- **MMS Implementation Tag '80'**

For contents and coding see [30]

- **MMS User preference profile name Tag '81'**

Contents:
-Alpha tagging of the MMS user preference profile.
Coding:
-this alpha-tagging shall use either:

- the SMS default 7-bit coded alphabet as defined in [38] with bit 8 set to 0. The alpha identifier shall be left justified; or

- one of the UCS2 coded options as defined in the annex of [30].

- **MMS User Preference information Tag '82'**

  Contents:
  -The following information elements may be coded; Sender Visibility, Delivery Report, Read-Reply, Priority, Time of Expiry and Earliest Delivery Time. Refer to [37], [39], [40], and [41].
  Coding:
  -Depending upon the MMS implementation as indicated in Tag '80'.

3.4.71 EF_{MMSUCP} (MMS User Connectivity Parameters)

If service n40 and n42 are allocated, this file shall be present.

This EF contains values for Multimedia Messaging Connectivity Parameters as determined by the user, which can be used by the ME for MMS network connection. This file may contain one or more sets of Multimedia Messaging User Connectivity Parameters. Each set of Multimedia Messaging User Connectivity Parameters may consist of one or more Interface to Core Network and Bearer information TLV objects (only for WAP), but shall contain only one MMS implementation TLV object (for WAP, M-IMAP and SIP), one MMS Relay/Server TLV object (for WAP, M-IMAP and SIP) and one Gateway TLV object (only for WAP). The order of the Interface to Core Network and Bearer information TLV objects in the MMS Connectivity TLV object defines the priority of the Interface to Core Network and Bearer information, with the first TLV object having the highest priority.

Identifier: '6F69'	Structure: Transparent	Optional
File Size: X_1+...+ X_n bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	CHV1/CHV2 (fixed during administrative management)
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 to X_1	MMS Connectivity Parameters TLV object	O	X_1 bytes
X_1+1 to X_1 + X_2	MMS Connectivity Parameters TLV object	O	X_2 bytes
...	...		
X_1+...+ X_{n-1}+1 to X_1+...+ X_n	MMS Connectivity Parameters TLV object	O	X_n bytes

For the contents and coding see 3.4.69.

3.4.72 $EF_{AuthCapability}$ (Authentication Capability)

If service n43 is allocated, this file shall be present. This EF stores authentication capabilities for each application supported by the R-UIM.

Identifier: '6F6A'	Structure: Linear Fixed	Optional
Record Length: 5 bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Application ID	M	1 byte
2-3	Authentication Capability	M	2 bytes
4-5	Reserved	M	2 bytes

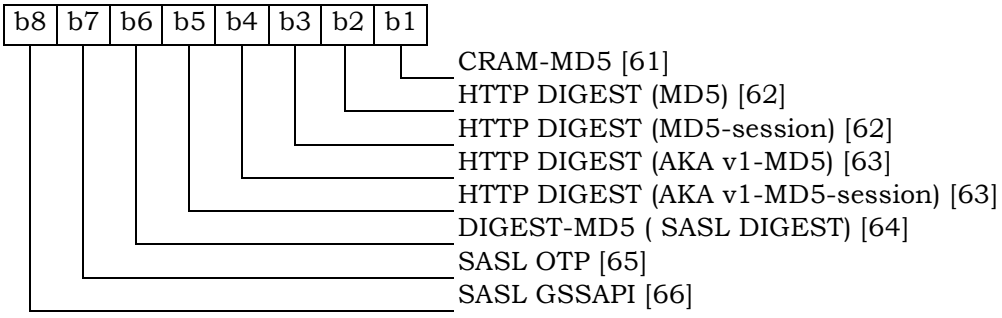
Coding:

Byte 1:

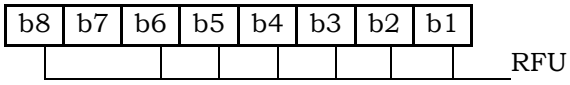
The coding for Application ID is as follows:

Binary Value	Application ID
'00000000'	MMS
'00000001'	MMD
'00000010'-'11111111'	Reserved

Byte 2:



Byte 3:



Bytes 4-5:

RFU.

The R-UIM shall set each subfield to '1' if it supports the corresponding authentication mechanism.

3.4.73 EF_{3GCIK} (3G Cipher and Integrity Keys)

If service n30 is allocated, this file shall be present.

This EF contains the cipher key CK and the integrity key IK produced by the '3G Access AKA' AUTHENTICATE command.

Identifier : '6F6B'	Structure : transparent	Optional
File size: 32 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 - 16	Cipher key CK	M	16 bytes
17 - 32	Integrity key IK	M	16 bytes

- Cipher key CK.
  Coding:
  -The least significant bit of CK is the least significant bit of the sixteenth byte. The most significant bit of CK is the most significant bit of the first byte.

- Integrity key IK.
  Coding:

The least significant bit of IK is the least significant bit of the thirty-second byte. The most significant bit of IK is the most significant bit of the seventeenth byte.

3.4.74 EF_{DCK} (De-Personalization Control Keys)

If service n46 is allocated, this EF shall be present.

This EF provides storage for the de-personalization control keys associated with the OTA de-personalization cycle of [44].

.

Identifier: '6F6C'	Structure: transparent	Optional
File size: 20 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 to 4	8 digits of Network Type 1 de-personalization control key	M	4 bytes
5 to 8	8 digits of Network Type 2 de-personalization control key	M	4 bytes
9 to 12	8 digits of service provider de-personalization control key	M	4 bytes
13 to16	8 digits of corporate de-personalization control key	M	4 bytes
17 to 20	8 digits of HRPD Network de-personalization control key	M	4 bytes

Empty control key fields shall be coded 'FFFFFFFF'.

3.4.75 EF_{GID1} (Group Identifier Level 1)

If service n44 is allocated, this EF shall be present.

This EF contains identifiers for particular R-UIM/ME associations. It can be used to identify a group of R-UIMs for a particular application.

Identifier: '6F6D'	Structure: transparent	Optional
File size: 1 to n bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 to n	R-UIM group identifier(s)	O	n bytes

3.4.76 EF_{GID2} (Group Identifier Level 2)

If service n45 is allocated, this EF shall be present.

This EF contains identifiers for particular R-UIM/ME associations. It can be used to identify a group of R-UIMs for a particular application.

Identifier: '6F6E'	Structure: transparent	Optional
File size: 1 to n bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 to n	R-UIM group identifier(s)	O	n bytes

NOTE: The structure of EF_{GID1} and EF_{GID2} are identical. They are provided to allow the network operator to enforce different levels of security dependant on an application.

3.4.77 EF$_{CDMACNL}$ (CDMA Co-operative Network List)

If service n47 is allocated, this EF shall be present.

This EF contains the Co-operative Network List for the multiple network personalization services defined in [44].

Identifier: '6F6F'	Structure: transparent	Optional
File size: 7n bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 to 7	Element 1 of co-operative net list	M	7 bytes
7n-6 to 7n	Element n of co-operative net list	O	7 bytes

- Co-operative Network List

Contents:

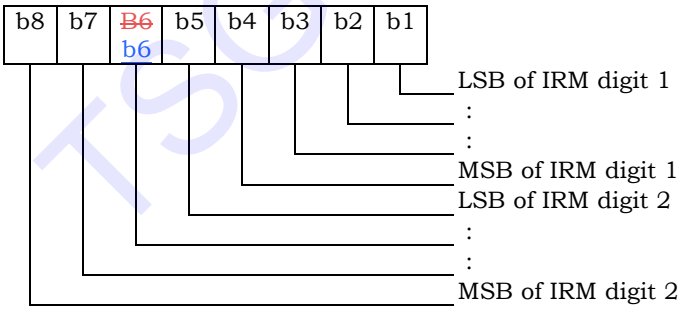
Service provider ID and corporate ID of co-operative networks.

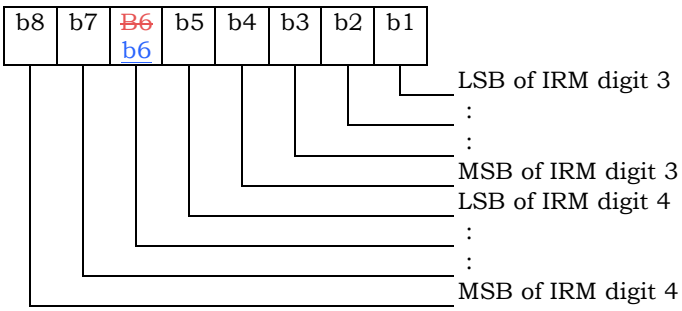
Coding:

For each 7 byte list element

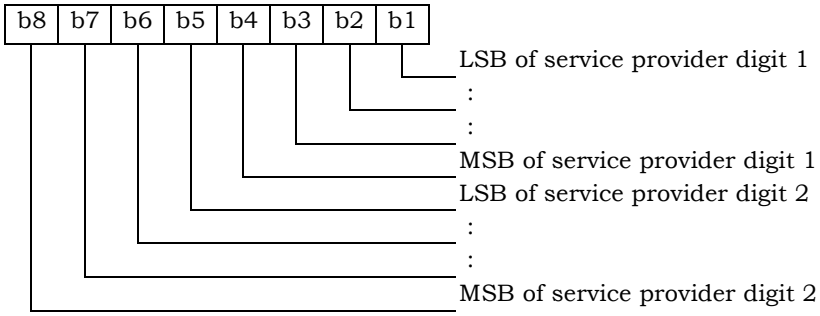
Byte 1 to 3: MCC + MNC: As per Annex A of [9].

Byte 4 to 5: 4 most significant digits of the International Roaming based MIN.

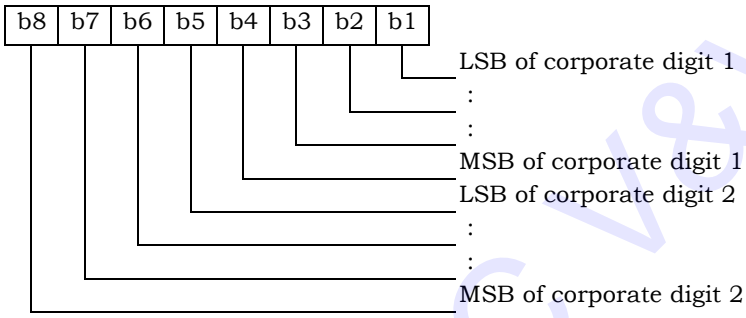




Byte 6:



Byte 7:



Empty fields shall be coded with 'FF'.
The end of the list is delimited by the first MCC field coded 'FFF'.

3.4.78 EF_{HOME_TAG} (Home System Tag)

This EF stores the Home System Tag, as described in Section 3.5.10.1 of [7].

Identifier: '6F70'	Structure: transparent	Mandatory
File size: X bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 - X	Home System Tag (see Section 3.5.10.1 of [7])	M	Variable

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

3.4.79 EF_{GROUP_TAG} (Group Tag List)

This EF stores the Group Tag List, as described in Section 3.5.10.3 of [7].

Identifier: '6F71'	Structure: transparent	Mandatory
File size: 'GROUP_TAG_LIST_SIZE'	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-GROUP_TAG_LIST_SIZE	Group Tag List (see Section 3.5.10.3 of [7])	M	Variable

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

3.4.80 $EF_{SPECIFIC\_TAG}$ (Specific Tag List)

This EF stores the Specific Tag List, as described in Section 3.5.10.5 of [7].

Identifier: '6F72'	Structure: transparent	Mandatory
File size: 'SPEC_TAG_LIST_SIZE'	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-SPEC_TAG_LIST_SIZE	Specific Tag List (see Section 3.5.10.5 of [7])	M	Variable

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

3.4.81 EF_{CALL_PROMPT} (Call Prompt List)

This EF stores the Call Prompt List, as described in Section 3.5.10.7 of [7].

Identifier: '6F73'	Structure: transparent	Mandatory
File size: 'CALL_PRMPT_LIST_SIZE'	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-CALL_PRMPT_LIST_SIZE	Call Prompt List (see Section 3.5.10.7 of [7])	M	Variable

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

3.4.82 EF_{SF_EUIMID} (Short Form EUIMID)

If service n8 is allocated, this file shall be present.

This EF stores the 56-bit electronic identification number (ID) unique to the R-UIM. The order of the digits when treated as 14 four-bit digits is shown in the table below, with 'd1' representing the leftmost/most significant digit and 'd14' representing the rightmost/least significant digit.

Identifier: '6F74'	Structure: transparent	Optional
File size: 7 bytes	Update activity: low	

Access Conditions:

READ	ALW
UPDATE	Never
INVALIDATE	Never
REHABILITATE	Never

	Description									
Bytes	8	7	6	5	4	3	2	1	M/O	Length
1	d13				d14				M	1 byte
2	d11				d12				M	1 byte
3	d9				d10				M	1 byte
4	d7				d8				M	1 byte
5	d5				d6				M	1 byte
6	d3				d4				M	1 byte
7	d1				d2				M	1 byte

3.4.83 EF_{ICCID} (ICC Identification)

EF_{ICCID} is defined in [18] with the following restrictions:

- This EF shall contain 18 digits of the actual ICCID followed by the check digit and a single 0xF filler digit.
- The ICCID shall be globally unique, using an Issuer Identifier Number registered with the ITU-T as specified in [47].

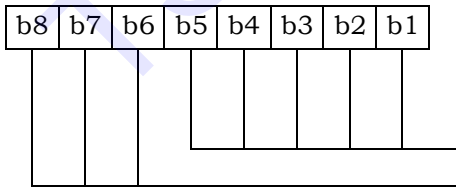
3.4.84 $EF_{AppLabels}$ (Application Labels)

This EF contains text labels that shall be associated with the icons or menu items used to launch applications. Use of these labels is optional and need only be provisioned if an operator desires to override the ME-defined labels.

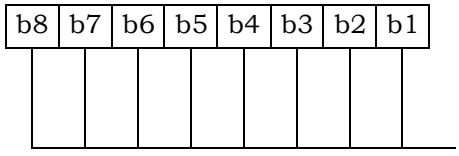
Identifier: '6F92'	Structure: Transparent	Optional
File size: 4+N*32	Update Activity: Low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Character Encoding	M	1 byte
2	Language Indicator	M	1 byte
3 – 4	Application Labels Present	M	2 bytes
5 – 36	Application $Label_1$	O	32 bytes
37 – 68	Application $Label_2$	O	32 bytes
…	…	O	…
5+(N-1)*32 to 36+(N-1)*32	Application $Label_N$	O	32 bytes

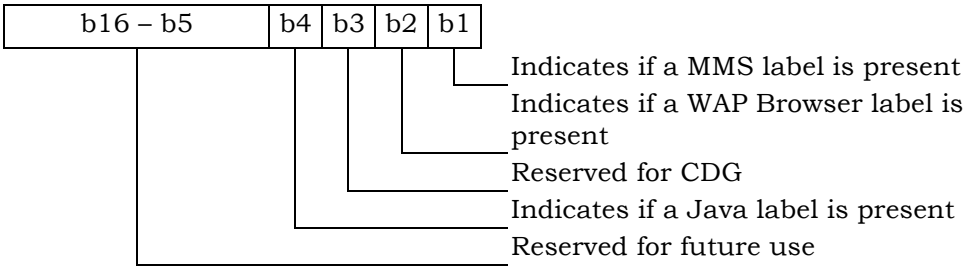
Character Encoding:



Language Indicator:



Application Labels Present: This field is a bitmask used to identify which Application Label Fields are present in the EF. Each bit represents a particular application as shown below:



If a bit is set to '1,' an Application Label Field for that application shall be present. If the bit is set to '0,' an Application Label Field for that application shall not be present and the ME's user interface will display the generic label for that application.

Application Label: Each Application Label field contains the text label to be displayed with the icon or menu item used to launch that application. The Application Label Present field identifies which Application Label fields are present in the EF. These Application Label fields shall be present in the same order as their corresponding bits in the Application Labels Present field. The string contents of each Application Label field shall use the SMS convention as defined in Tables 9.1-1 and 9.2-1 of [Informative 1]. The string shall be left justified. Unused bytes shall be set to 'FF.'

If the string is coded as 7-bit, the SMS default 7-bit coded alphabet as referenced in [Informative 1] with bit 8 set to 0 shall be used.

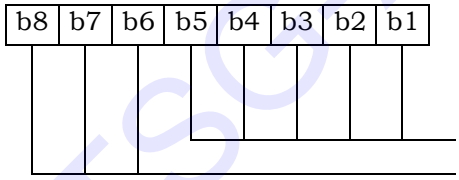
3.4.85 EF_{Model} (Device Model Information)

This EF contains the model information of the ME. Similar to $EF_{ESN\_MEID\_ME}$, this EF is populated by the device during power-up. This EF enables applications running in the R-UIM to provide model information to the network either automatically or on demand.

Identifier: '6F90'	Structure: Transparent	Optional
File Size: 126	Update activity: Low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

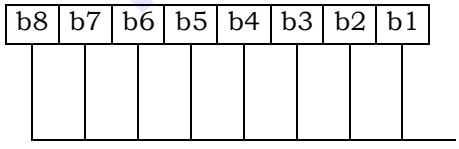
Bytes	Description	M/O	Length
1	Character Encoding	M	1 byte
2	Language Indicator	M	1 byte
3-34	Model Information	M	32 bytes
35-66	Manufacturer Name	M	32 bytes
67-126	Software Version Information	M	60 bytes

Character Encoding:



CHARi encoding per [Informative 1, Table 9.1-1, Data Field Encoding Assignments]
Reserved for future use

Language Indicator:



Language Indicator as specified in Table 9.2-1, Language Indicator Value Assignments, in [Informative 1]

Model Information: This field is a string indicating the model name of the device (e.g., “ABCCOM-XYZ”). The string contents shall use the SMS convention as defined in Tables 9.1-1 and 9.2-1 of [Informative 1]. The string shall be left justified. Unused bytes shall be set to ‘FF.’

Manufacturer Name: This field is a string indicating the manufacturer of the device. The string contents shall use the SMS convention as defined in Tables 9.1-1 and 9.2-1 of [Informative 1]. The string shall be left justified. Unused bytes shall be set to ‘FF.’

Software Version Information: This field is a string indicating the software version of the device (e.g., “6.0 patch 01”). The string contents shall use the SMS convention as defined in Tables 9.1-1 and 9.2-1 of [Informative 1]. The string shall be left justified. Unused bytes shall be set to ‘FF.’

3.4.86 EF_{RC} (Root Certificates)

If service n16 (Root Certificates) is allocated, this EF shall be present.

This EF contains the root certificates for applications on the device. One or more applications are associated with each certificate.

Identifier: '6F91'	Structure: Transparent	Optional
File Size: X_1+…+X_n	Update activity: Low	
Access Conditions: READ ALW UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 to X_1	Certificate TLV Object	M	X_1 bytes
X_1+1 to X_1+X_2	Certificate TLV Object	O	X_2 bytes
…	…	O	…
X_1+…+X_{n-1}+1 to X_1+…+X_n	Certificate TLV Object	O	X_n bytes

Unused bytes shall be set to 'FF.' A Tag value of 'FF' indicates the end of valid data.

Certificate TLV Object – Contents:

Description	Value	M/O	Length
Certificate Tag	'80'	M	1 byte
Length	Note 1	M	Note 2
Certificate Type	Note 3	M	1 byte
Certificate Information	Note 4	M	Variable
Applications	Note 3	M	2 bytes

NOTE 1: This is the total size of the constructed TLV object (not including the tag and this length).
NOTE 2: The length is coded according to [49] using primitive encoding and the minimum number of octets.
NOTE 3: See coding below.
NOTE 4: Binary data for the certificate information as defined in the corresponding Certificate Type as defined below, e.g., X.509.

Certificate Type – Coding:

Value	Name	Notes
0	DER Encoded Binary X.509	See section 7 "Public-keys and public-key certificates" in [48] for the definition. The binary encoding is per DER encoding defined in [49].
1	Base64 Encoded X.509	See section 7 "Public-keys and public-key certificates" in [48]. The encoding is per DER encoding defined in [49] and the DER binary data is converted to Base 64 text format.
2	PKCS #7	See section 6.5 "ExtendedCertificateOrCertificate" in [50] for the definition. The binary encoding is per DER encoding defined in [49].
3	PKCS #12	See section 4.2.3 "The CertBag type" in [51] for the definition. The binary encoding is per DER encoding defined in [49].
4-255	Reserved for future use	

APPLICATIONS: This field is a bitmask used to indicate which applications are associated with a particular certificate. If the same certificate is being used for all applications signed by the operator, only bit 1 (Unspecified) will be set. Otherwise, if the operator signs different applications using different certificates, the bit for each application associated with the certificate shall be set. Note that, while each certificate may be associated with multiple applications, each application may only be associated with one certificate.

Bit	Application
1	Unspecified (all applications use the same profile)
2	Reserved
3	WAP Browser
4	Reserved for CDG
5	Java
6	Reserved for CDG
7	Terminal (tethered mode for terminal access)
8-16	Reserved for future use

3.4.87 EF_{SMSCAP} (SMS Capabilities)

If services n4 (Short Message Storage) and n15 (Messaging and 3GPD Extensions) are allocated, this EF shall be present.

This EF contains information about SMS Capabilities.

Identifier: '6F76'	Structure: Transparent	Optional
File size: 4 bytes	Update Activity: Low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	SMS Retry Period	M	1 byte
2	SMS Retry Interval	M	1 byte
3	SMS Flags	M	1 byte
4	SMS Preferred Service Option	M	1 byte

SMS Retry Period: This is the overall time period (in seconds) during which the Mobile Originated (MO) SMS retries can be performed. 0 means that MO SMS retry is disabled.

SMS Retry Interval: This is the time interval (in seconds) that the device shall wait before the next retry attempt can be made after a MO SMS failure.

SMS Flags: 0 – disabled; 1 – enabled

Bit	Parameter Indicated
1	Send On Access (Allow MO SMS to be sent over Access Channel)
2	Send On Traffic (Allow MO SMS to be sent over Traffic Channel)
3	Send as Standard EMS (Network supports standard EMS per [8])
4-8	Reserved for future use

SMS Preferred Service Option: This is the preferred service option to be used when the device sets up SMS traffic channel for sending messages.

Value	Description
0	Device Default
1	Service Option 6
2	Service Option 14
3-255	Reserved for future use

3.4.88 $EF_{MIPFlags}$ (Mobile IP Flags)

If services n38 (3GPD-MIP) and n15 (Messaging and 3GPD Extensions) are allocated, this EF shall be present.

This EF contains the configuration flags for Mobile IP.

Identifier: '6F78'	Structure: Transparent	Optional
File size: 1 byte	Update Activity: Low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	MIP_FLAGS	M	1 byte

MIP_FLAGS: 0 – disabled; 1 – enabled

Bit	Parameter Indicated
1	Mobile IP MN HA Authentication [23]
2	Mobile IP Pre Rev 6 handoff optimization
3	Mobile IP PPP Re-sync during hand-down from 1xEV-DO Rev 0 to 1x
4	Mobile IP Re-registration only if data has been transferred since last registration in order to extend Mobile IP address lifetime
5-8	Reserved for future use

3.4.89 $EF_{3GPDUPPExt}$ (3GPD User Profile Parameters Extension)

If service n20 (3GPD-SIP) or n38 (3GPD-MIP) is allocated and service n15 (Messaging and 3GPD Extensions) is allocated, this EF shall be present.

This EF contains the additional parameters for Simple IP and Mobile IP User Profiles in order to fully support the feature of multiple profiles.

Identifier: '6F7D'	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
X	UPP Extension Block	M	X bytes

Unused bytes shall be set to 'FF.'

UPP Extension Block structure:

Field	Length (bits)
NUM_NAI	4
NUM_NAI occurrences of the following fields:	
NAI_ENTRY_INDEX	4
APPLICATIONS	32
PRIORITY	8
DATA_RATE_MODE	4
DATA_BEARER	4
RESERVED	0 or 4

NUM_NAI: Number of UPP Extension instances. This number shall be the same as NUM_NAI in the base user profile EF (EF_{SIPUPP} or EF_{MIPUPP}).

NAI_ENTRY_INDEX: Index to the list of UPP Extension instances. This index shall point to the UPP Extension instance that is corresponding to the base UPP instance with the same index value as defined in EF_{SIPUPP} or EF_{MIPUPP}.

APPLICATIONS: This field is a bitmask used to indicate which applications are associated with a particular profile. The applications shall use the profile having the

"Unspecified" bit set in the APPLICATIONS bitmask if they are not present in any other profiles.

Bit	Application
1	Unspecified *(used by applications not present in any other profile)*
2	MMS
3	WAP Browser
4	Reserved for CDG
5	Java
6	Reserved for CDG
7	Terminal *(tethered mode for terminal access)*
8	Operator Administration (e.g. BIP)
9-32	Reserved for future use

PRIORITY: When attempting to launch a new application, it is possible that another application is already active and has already established a data session. If the new application has the same PRIORITY value as the previous application that established the existing data session, the new application may simply reuse the existing data session.

If the new application has a different PRIORITY than the previous application that set up the existing data session, the device may use the PRIORITY to determine which application has higher priority, as follows:

Value	Priority
0	Highest priority category
1	Second highest priority category (lower than 0; higher than 2 and others)
2	Third highest priority category (lower than 0 or 1; higher than 3 and others)
:	:
255	Lowest priority

DATA_RATE_MODE: Data Rate Mode

Value	Application
0	Low Speed: Low speed service options only
1	Medium Speed: F-SCH with service option 33 only
2	High Speed: F-SCH and R-SCH with service option 33
3-15	Reserved for future use

DATA_BEARER: Data Bearer

Value	Application
0	Hybrid 1x/1xEV-DO
1	1x only
2	1xEV-DO only
3-15	Reserved for future use

3.4.90 Reserved

3.4.91 $EF_{IPV6CAP}$ (IPv6 Capabilities)

If services n31 (IPv6) and n15 (Messaging and 3GPD Extensions) are allocated, this EF shall be present.

This EF contains information about IPv6 capabilities.

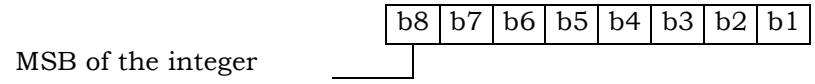
Identifier: '6F77'	Structure: Transparent	Optional
File size: 21 bytes	Update Activity: Low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-2	Initial neighbor solicitation delay time	M	2 bytes
3-4	Solicitation interval	M	2 bytes
5-6	Re-solicitation interval	M	2 bytes
7-8	Maximum solicitation attempts	M	2 bytes
9-10	Maximum re-solicitation attempts	M	2 bytes
11-12	Pre-RA expiry re-solicitation time	M	2 bytes
13-20	IID Information	M	8 bytes
21	IPv6 Flags	M	1 byte

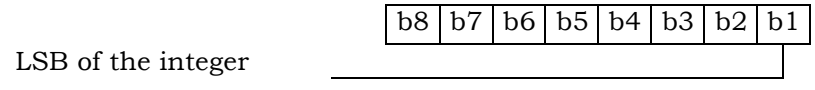
Initial neighbor solicitation delay time *(in units of 100ms)***:** Time MS waits after the IID (Interface ID) has been negotiated before sending an RS (Router Solicitation) in an attempt to receive an RA (Router Advertisement).

Coding: 16-bit integer.

Byte 1:



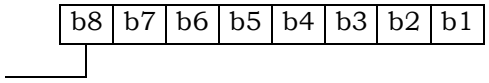
Byte 2:



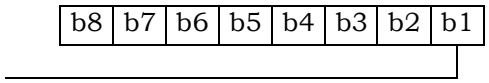
Solicitation interval *(in units of 100ms)***:** Amount of time the MS waits before sending a subsequent RS after a previous one.

Coding: 16-bit integer.

Byte 1:



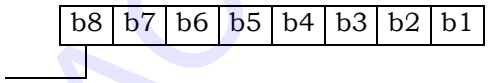
Byte 2:



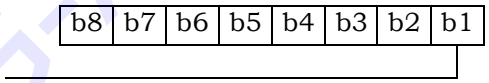
Re-solicitation interval *(in units of 100ms)***:** Amount of time between solicitations sent while re-soliciting for a new RA. This interval applies only after the MS has previously received one valid RA and is soliciting for a new one to renew the lifetimes of the current prefix or retrieve a non-deprecated prefix.

Coding: 16-bit integer.

Byte 1:



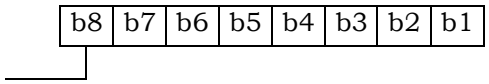
Byte 2:



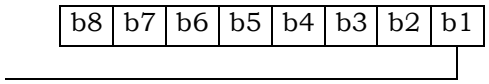
Max solicitation attempts: Number of solicitation attempts to make for initial IPv6 session establishment, when an RA is not received in response before giving up IPv6 auto-configuration.

Coding: 16-bit integer.

Byte 1:



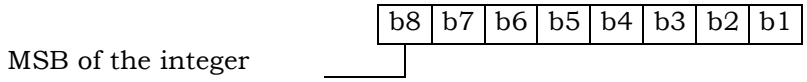
Byte 2:



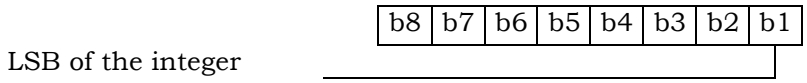
Max re-solicitation attempts: Number of solicitation attempts to make to re-solicit for a new RA.

Coding: 16-bit integer.

Byte 1:



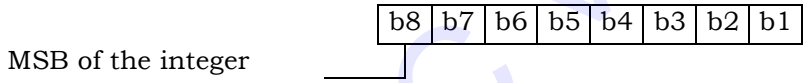
Byte 2:



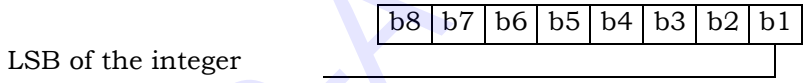
Pre-RA expiry re-solicitation time *(in units of 100ms)* **:** Amount of time before the current RA expires to begin re-solicitations.

Coding: 16-bit integer.

Byte 1:



Byte 2:



IID Information: IID is part of the IPv6 address. See [51] for information on coding.

IPv6 Flags: Identify IPv6 behavior. Coding (0 – Disabled; 1 – Enabled).

Bit	Parameter Indicated
1	Use IPv6
2	Failover from IPv6 to IPv4
3	PDSN as proxy IPv6 DNS server. When enabled, the MS forwards all DNS requests to the PDSN. The PDSN forwards requests to the appropriate DNS server. This parameter is meaningful only if the primary and secondary DNS server addresses are not available.
4-8	~~Reserved for future use~~RFU

3.4.92 $EF_{TCPConfig}$ (TCP Configurations)

If service n20 (3GPD-SIP) or n38 (3GPD-MIP) is allocated and service n15 (Messaging and 3GPD Extensions) is allocated, this EF shall be present.

This EF contains information about Transmission Control Protocol configurations.

Identifier: '6F79'	Structure: Transparent	Optional
File size: 2 bytes	Update Activity: Medium	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	TCP Flags	M	1 byte
2	TCP Keep-Alive Idle Timer	M	1 byte

TCP Flags:

Coding (0 – Disabled; 1 – Enabled):

Bit	Parameter Indicated
1	TCP Graceful close of dormant connections
2-8	Reserved for future use

TCP Keep-Alive Idle Timer:

Coding: Number of minutes. A value of 0 means that the TCP keep-alive feature is disabled on the ME.

3.4.93 EF_{DGC} (Data Generic Configurations)

If service n20 (3GPD-SIP) or n38 (3GPD-MIP) is allocated and service n15 (Messaging and 3GPD Extensions) is allocated, this EF shall be present.

This EF contains miscellaneous data configuration items.

Identifier: '6F7A'	Structure: Transparent	Optional
File size: 3 bytes	Update Activity: Medium	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Data dormant timer	M	1 byte
2	EPZID Type Information	M	1 byte
3	Hysteresis Activation Time	M	1 byte

Data dormant timer: Number of seconds to wait before going into data dormant mode, which shall be at least 20 seconds.

EPZID Type Information: Contains the Extended Packet Zone ID Types.

Value	Description
0	Packet Zone ID
1	Packet Zone ID plus SID
2	Packet Zone ID plus SID and NID
3-255	Reserved for future use

Hysteresis Activation Time: This is the number of seconds that the device should wait before it goes into hysteresis state and adds new Packet Zone IDs to the packet zone list as needed. See [54] for details on the usage of this timer.

3.4.94 $EF_{WAPBrowserCP}$ (WAP Browser Connectivity Parameters)

If service n21 (WAP Browser) is allocated, this EF shall be present.

This EF contains the connectivity parameters for a WAP Browser application, such as Gateway and Home URL information. At least one gateway shall be configured in this EF as the primary gateway for browsing. Additional gateways as part of the additional instances of Connectivity Parameters can be optionally configured as secondary gateways in the order of priority as they appear in this EF.

Identifier: '6F7B'	Structure: Transparent	Optional
File Size: X_1+…+ X_n	Update activity: Low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 to X_1	WAP Browser Connectivity Parameters TLV object	M	X_1 bytes
X_1+1 to X_1 + X_2	WAP Browser Connectivity Parameters TLV object	O	X_2 bytes
…	…		
X_1+…+ X_{n-1}+1 to X_1+…+ X_n	WAP Browser Connectivity Parameters TLV object	O	X_n bytes

Unused bytes shall be set to 'FF.' A Tag value of 'FF' indicates the end of valid data.

WAP Browser Connectivity Parameters Tags:

Description	**Tag Value**
WAP Browser Connectivity Parameters Tag	'AC'
Gateway Tag	'83'
HomeURL Tag	'80'

WAP Browser Connectivity Parameters TLV Object contents:

Description	Value	M/O	Length (bytes)
WAP Browser Connectivity Parameters Tag	'AC'	M	1
Length	Note 1	M	Note 2
Gateway Tag	'83'	O	1
Gateway Length	Z	O	Note 2
Gateway Information	--	O	Z
HomeURL Tag	'80'	M	1
HomeURL Length	X	M	Note 2
HomeURL Information	--	M	X
NOTE 1: This is the total size of the constructed TLV object (not including the tag and this length). NOTE 2: The length is coded according to [49] using primitive encoding and the minimum number of octets.			

Gateway Tag: This contains information needed to access the WAP Gateway/Proxy server. See description of EF_{MMSICP} for the definition of Gateway TLV Object.

HomeURL Tag: This contains the URL for the WAP browser's home page for the current particular connectivity parameters. For contents and syntax of URL TLV data object values, see [53]. The URL shall be encoded to an octet string according to UTF-8 encoding rules as specified in [46].

3.4.95 $EF_{WAPBrowserBM}$ (WAP Browser Bookmarks)

If service n21 (WAP Browser) is allocated, this EF shall be present.

This EF contains bookmarks that may be provisioned by the operator and/or updated by the user.

Identifier: '6F7C'	Structure: Transparent	Optional
File Size: Variable	Update activity: High	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 to X_1	Bookmark TLV object	M	X_1 bytes
X_1+1 to X_1+X_2	Bookmark TLV Object	O	X_2 bytes
…	…	O	…
X_1+X_2+…+X_{n-1}+1 to X_1+X_2+…+X_{n-1}+X_n	Bookmark TLV Object	O	X_n bytes

Unused bytes shall be set to 'FF.' A value of 'FF' in place of Bookmark Tag field indicates the end of valid data.

Bookmark TLV object contents:

Description	**Value**	**M/O**	**Length (bytes)**
Bookmark Tag	'AD'	M	1
Length	Note 1	M	Note 2
URL Tag	'80'	M	1
Length	Y	M	Note 2
URL Information	--	M	Y
Bookmark Name Tag	'81'	O	1
Length	Z	O	Note 2
Bookmark Name Information	--	O	Z
NOTE 1: This is the total size of the constructed TLV object (not including the tag and this length). NOTE 2: The length is coded according to [49] using primitive encoding and the minimum number of octets.			

URL Information: For contents and syntax of URL TLV data object values, see [53]. The URL shall be encoded to an octet string according to UTF-8 encoding rules, as specified in [46].

Bookmark Name Information: This field shall be encoded to an octet string according to UTF-8 encoding rules as specified in [46].

3.4.96 $EF_{MMSConfig}$ (MMS Configuration)

If services n40 (Multimedia Messaging Service) and n15 (Messaging and 3GPD Extensions) are allocated, this EF shall be present.

This EF contains the configuration of MMS.

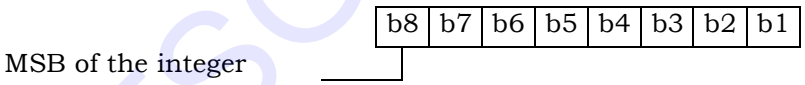
Note that this EF does not contain configuration associated with how the MMS client connects to the MMS service. This type of configuration information is included in EF_{MMSICP}.

Identifier: '6F7E'	Structure: Transparent	Optional
File size: 8 bytes	Update Activity: Medium	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

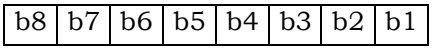
Bytes	Description	M/O	Length
1-4	Max Message Size Value	M	4 bytes
5	Retry Times Value	M	1 bytes
6	Retry Interval Value	M	1 bytes
7-8	MMSC Timeout Value	M	2 bytes

Max Message Size: This is the maximum MMS message size (in bytes) allowed by the operator. Coding: 32-bit integer.

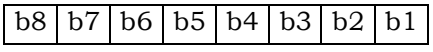
Byte 1:



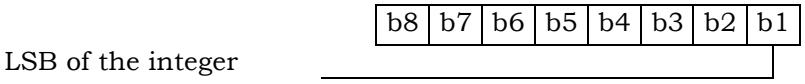
Byte 2:



Byte 3:



Byte 4:



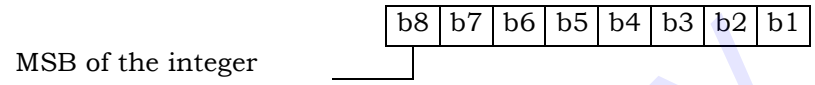
Retry Times: This is the number of times the MMS application will retry for sending a message. Coding: 8-bit integer.

Retry Interval: This is the number of seconds to wait before the next retry is attempted. Coding: 8-bit integer.

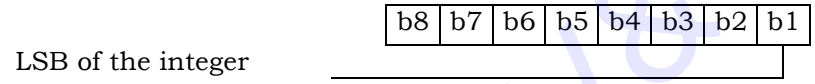
MMSC Timeout: This is the number of seconds for the device to wait for response from Mobile Messaging Service Center (MMSC) before declaring it as an MMSC timeout.

Coding: 16-bit integer.

Byte 1:



Byte 2:



3.4.97 EF_{JDL} (Java Download URL)

If service n22 (Java) is allocated, this EF shall be present.

This EF contains the information for downloading Java applications from the Java download server.

Identifier: '6F7F'	Structure: Transparent	Optional
File size: Variable (Y≥X)	Update Activity: Low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-X	Java Download URL	M	X bytes

Unused bytes shall be set to 'FF'.

Java Download URL:

This field of X bytes contains the URL for the Java download server and a termination byte. For contents and syntax, see [53]. The URL shall be encoded as an octet string according to UTF-8 encoding rules, as specified in [46]. The termination byte is set to '00'.

3.5 Coding of Packet Data Security-Related Parameters

This section specifies the coding of packet data security-related parameters to be stored in the R-UIM securely. These parameters are used for IP based authentication functions by the R-UIM. Also, these parameters can be read or updated via OTA commands (i.e. 3GPD CONFIGURATION/DOWNLOAD REQUEST command) only when the Secure Mode is turned on. If the R-UIM receives the 3GPD CONFIGURATION REQUEST command or 3GPD DOWNLOAD REQUEST command containing Block_ID for Simple IP CHAP SS, Mobile IP SS or HRPD Access Authentication CHAP SS Parameters Block and Secure Mode is not active, then the R-UIM shall return SW1='69' and SW2='82' (Security status not satisfied [55])~~a Result Code of '00110011' (Rejected – Secure Mode not active)~~.

3.5.1 Simple IP CHAP SS Parameters

The Simple IP CHAP SS Parameters shall be present if service n20 is allocated (See Section 3.4.18) and coded as follows:

Bytes	Description	Length
1	Length of SimpleIP CHAP SS Parameter Block	1 bytes
2 – X+1	See [7], SimpleIP CHAP SS Parameter Block	X bytes

Details of the SimpleIP CHAP SS Parameters Block are defined in Section 3.5.8.10 of [7].

3.5.2 Mobile IP SS Parameters

The Mobile IP SS Parameters shall be present if service n38 is allocated (See Section 3.4.18) and coded as follows:

Bytes	Description	Length
1	Length of MobileIP SS Parameter Block	1 bytes
2 – X+1	See [7], MobileIP SS Parameter Block	X bytes

Details of the MobileIP SS Parameters Block are defined in Section 3.5.8.11 of [7].

3.5.3 HRPD Access Authentication CHAP SS Parameters

The HRPD Access Authentication CHAP SS Parameters shall be present if service n5 is allocated (See Section 3.4.18) and coded as follows:

Bytes	Description	Length
1	Length of HRPD Access Authentication CHAP SS Parameters Block	1 bytes
2 – X+1	See [7], HRPD Access Authentication CHAP SS Parameters Block	X bytes

Details of the HRPD Access Authentication CHAP SS Parameters Block are defined in Section 3.5.8.14 of [7].

3.6 Coding of Shared Secret Used in IETF Protocol

This section specifies the coding of the shared secret to be stored in the R-UIM securely, which is used in Authentication Functions by the R-UIM.

The Shared Secret shall be present if service n40 is allocated (See Section 3.4.18) and coded as follows:

Bytes	Description	Length
1-2	Length of Shared Secret	2 bytes
3 – X+2	Shared Secret, see IETF RFCs in 3.4.72	X bytes

3.7 Multi-Mode Card

Multi mode card (e.g. CDMA and GSM) shall comply with both this document and [17]. In case of multi- mode MS supporting multiple modes, if one mode fails to initialize, then the MS shall attempt to initialize the other modes.

4 AUTHENTICATION, SECURITY AND COMMANDS

This section describes the interface between the ME and the R-UIM. Details of the ANSI-41 protocols [15] are provided in order to clarify the interface. Section 4.1 describes parameter storage and flow. Section 4.2 describes the components of the ANSI-41-based security procedures [15] within the context of a R-UIM environment. Section 4.3 specifies detailed commands and responses between the ME and the R-UIM, and uses section 4.2 as a reference. This section also describes Security-Related Commands (Sec. 4.4), OTASP/OTAPA Commands (Sec. 4.5) to support OTASP [7], ESN and MEID Management Commands (Sec. 4.6), Packet Data Security-Related Commands (Sec. 4.7), their corresponding Commands (Sec. 4.8), BCMCS Commands (Sec. 4.9), Application Authentication Commands (Sec. 4.10), Authentication and Key Agreement procedure (AKA)-related Functions (Sec. 4.11) and their corresponding commands (Sec. 4.12).

The authentication procedures may be tested using the test vectors from Section 3 of [20].

4.1 Parameter Storage and Parameter Exchange Procedures

The following parameters are stored on the R-UIM:

- Algorithm(s) for Authentication and Key Generation. Currently [15]-related security functions utilize the CAVE algorithm for these functions.
- A-key, which is accessible only to the algorithm used for Key Generation. The A-key may be programmed into the R-UIM directly by the service provider or it may be programmed into the R-UIM through an over-the-air procedure. The A-key is not accessible by the ME. Therefore the method of storage on the R-UIM is not specified in this document. During the execution of some procedures, it is necessary that two values (“old” and “new”) of the A-key be stored.
- Shared Secret Data (SSD), which is accessible only to the Authentication and Key Generation functions. SSD is not accessible by the ME. Therefore the method of storage on the R-UIM is not specified in the document. During the execution of some procedures, it is necessary that two values, SSD_S (new) and SSD (old) be stored.
- Temporary (typically per-call) secret parameters used for the generation of ciphering keys subsequent to the authentication process.
- COUNT, accessible by the ME. COUNT is incremented upon network command.
- International Mobile Station Identity, consisting of both IMSI_M and IMSI_T. IMSI_M contains a Mobile Identification Number (MIN) in its lower 10 digits. IMSI_T is not related to the MIN. Subscription Identity is accessible by the ME.
- UIMID, a parameter that is stored in EF_{RUIMID}.
- Service Programming Code (SPC), stored in EF_{SPC} and used in the OTASP/OTAPA procedures.

- OTAPA/SPC_Enable, storing the user's input to the OTASP/OTAPA procedures in $EF_{OTAPASPC}$.
- NAM_LOCK, storing the lock/unlock status of the NAM in $EF_{NAMLOCK}$.
- Root Key, which is accessible only to the algorithm used for Key Generation. The Root Key may be programmed into the R-UIM directly by the service provider or it may be programmed into the R-UIM through the procedures defined in [7]. The Root Key is not accessible by the ME. Therefore the method of storage on the R-UIM is not specified in this document. During the execution of some procedures, it is necessary that two values ("old" and "new") of the Root Key be stored.

The following parameters are stored in the ME:

- All algorithms used for the encryption of voice, user data and signaling messages.
- Key-processing for ECMEA and ECMEA_NF functions.
- ESN_ME.
- MEID_ME.
- Control mechanism for OTASP/OTAPA procedures

The following parameters are passed from the ME to the R-UIM during the course of security-related procedures:

- RAND, the "global" random challenge, available in the overhead information.
- Last Dialed Digits, a subset of the digits used to identify the called party. The R-UIM uses these to compose the "Auth Data" field for some ME messages. Refer to Table 2.3.12.1-1 of [5] or Table 6.3.12.1-1 of [14], entitled "Auth_Signature Input Parameters".
- RANDU, a "unique" random challenge sent by the network.
- AUTHBS, an authentication response sent from the network during the SSD Update process.
- RANDSeed, a random number that may be used to generate RANDBS.
- RANDSSD, the parameter that accompanies an SSD update command sent by the network to initiate an SSD update.
- ESN_ME, passed from the ME to the R-UIM upon insertion of the R-UIM into the ME. Also it is sent in an AUTHENTICATE (Run CAVE) Command or an UPDATE SSD command. If EF_{USGIND} bit 1 = '0', the ESN value received in a security command shall be used in the authentication algorithm regardless of what is stored in $EF_{ESN\_MEID\_ME}$.

The following parameters are passed from the ME to the R-UIM during the course of OTASP/OTAPA procedures:

- RANDSeed, a 32-bit random number that accompanies the OTAPA REQUEST.
- RANDSeed, a 160-bit random number that is a parameter in the MS KEY REQUEST.
- A-key/Root Key generation parameters P, P Length, G, G Length, A-key Protocol Revision, BS Result and BS Result Length.
- Block ID, Block Length, Parameter Data, Offset and Size parameters that refer to stored data as components of CONFIGURATION, VALIDATION and DOWNLOAD request messages.
- Start/Stop indicator as part of OTAPA REQUEST Message
- pESN, the parameter that accompanies the OTAPA REQUEST command (if ME is assigned with MEID and service n9 is allocated and activated)

The following parameters are passed from the R-UIM to the ME during the course of security-related procedures:

- AUTHR, the response to the “global challenge”.
- Keys, as needed, for use with the encryption algorithm(s). These may include a 64-bit key and a variable length VPM.
- AUTHU, the response to a “unique” challenge.
- RANDBS, the network authentication challenge for the SSD Update procedure.

The following parameters are passed from the R-UIM to the ME during the course of OTASP/OTAPA procedures:

- RAND_OTAPA, for network validation.
- A-key/Root Key generation parameters MS Result and MS Result Length.
- Result Code for most commands to indicate success/failure and reason(s) for failure.
- Block ID, Block Length, Parameter Data, Offset and Size as needed to identify segments of stored data.

4.2 Description of Security-Related Functions

The ME should start and finish the executions of all of the commands related to an [15] based security procedure in order and within the same Dedicated File (DF) environment.

The R-UIM performs the following operations; managing shared secret data, performing authentication calculations and generating encryption keys and managing the call history parameter.

4.2.1 Managing Shared Secret Data

The R-UIM stores and manages the SSD that is used as the derived secret variable for all authentication response calculations and subsequent key generations. SSD is derived from the "A-key" stored in the R-UIM. SSD updates are initiated when the network issues the command UPDATE SSD, containing the parameter RANDSSD, to the ME. Details of the SSD update procedure are described in [5] and [14].

A subscriber's home network is the only entity that may update the subscriber's Shared Secret Data (SSD). This is illustrated in the figure below. When the network launches an SSD Update to a particular subscriber, the subscriber's ME will first store the parameter RANDSSD and then generate a random number called RANDSeed. The ME begins the BASE STATION CHALLENGE function by passing the parameter RANDSeed to the R-UIM. This in turn causes the R-UIM to generate RANDBS. The relationship of RANDBS to RANDSeed is specified by the issuer of the R-UIM. The R-UIM may derive RANDBS by applying a pseudo-random process to RANDSeed, or it may ignore RANDSeed and generate RANDBS independently. RANDBS should not be the same for consecutive identical values of RANDSeed. The command BASE STATION CHALLENGE directs the R-UIM to pass RANDBS to the ME, which in turn forwards RANDBS to the network.

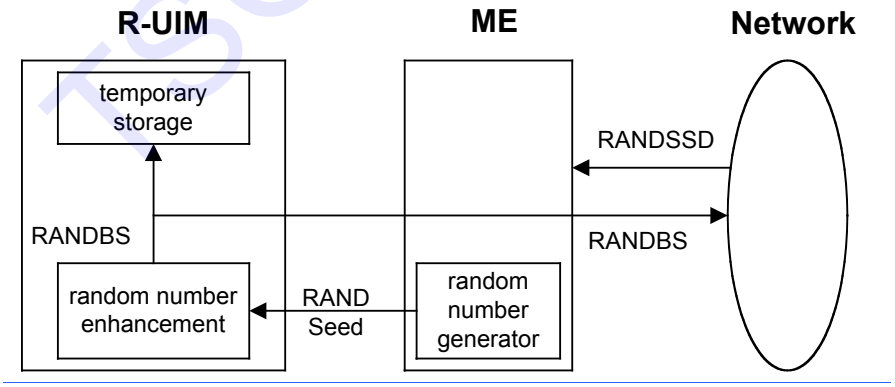


Figure 2. Base Station Challenge Function

Next, the ME updates SSD by sending the UPDATE SSD command to the R-UIM, containing the parameter RANDSSD and a control data field. Refer to Figure 3. The R-UIM then calculates a new (trial) value of SSD (SSD_S) and calculates an expected value of the network's response to RANDBS, called AUTHBS. The parameters ESN and IMSI used for these calculations are determined at the time of R-UIM insertion into the ME in accordance

with EF_{USGIND}. If ESN_ME rather than UIMID is chosen (i.e. EF_{USGIND} bit 1 = '0'), the value used as input to authentication algorithms shall be the one received from security commands, regardless of what is stored in $EF_{ESN\_MEID\_ME}$. For details, refer to section 4.6, "ESN and MEID Management Command", and to section 3.4.2, EF_{IMSI_M}.

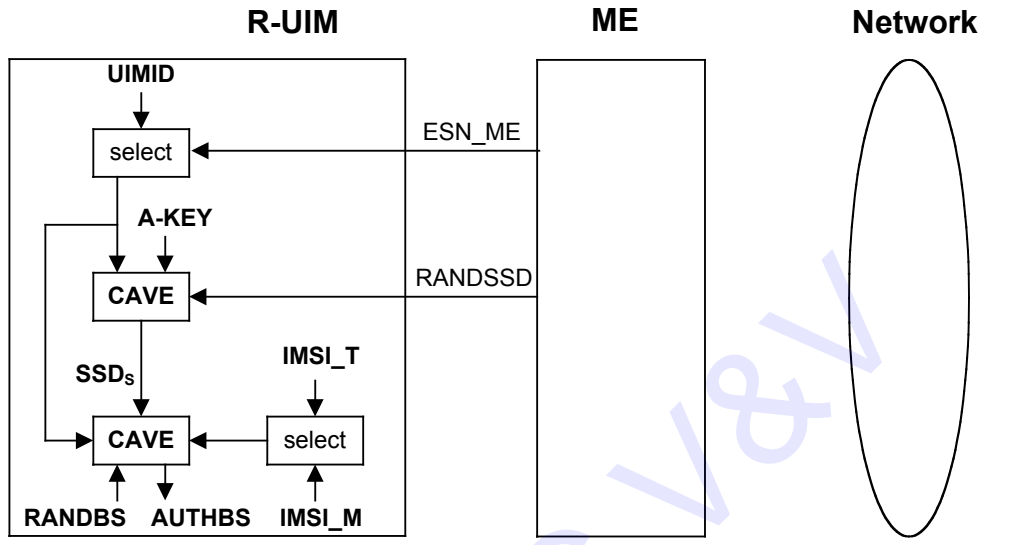


Figure 3. Update SSD Function, AUTHBS Calculation

In the network, the parameter RANDSSD is also used to generate a new value of SSD (SSD_S) for the selected R-UIM. When RANDBS is received from the subscriber's ME, the network combines it with SSD_S to calculate AUTHBS. AUTHBS is then sent from the network to the subscriber's phone. Refer to Figure 4. The ME in turn forwards the received value of AUTHBS to the R-UIM as a parameter of the CONFIRM SSD function. The R-UIM then compares its calculated value of AUTHBS to that sent by the network.

If the R-UIM finds the two values to be equivalent, the SSD Update procedure has been a success. SSD_S is then stored in semi-permanent memory on the R-UIM and used for all subsequent authentication calculations, with one exception, noted below. If the two values of AUTHBS are different, the R-UIM discards SSD_S and continues to retain its current value. Refer to Figure 4.

If the SSD Update procedure is being performed as part of an OTASP/OTAPA procedure, the ME shall set "process control" bit 3 to the value of '1' as an input parameter of the "UPDATE SSD" command. This will cause the R-UIM to retain the current value of SSD in semi-permanent memory but use SSD_S for re-authentication calculations. The R-UIM will set the value of SSD to SSD_S only upon R-UIM acceptance of the "COMMIT Request Message" from the network.

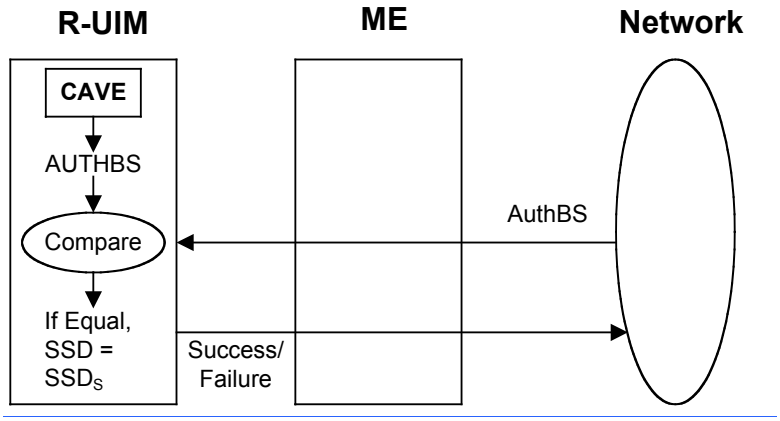


Figure 4. Confirm SSD Function

4.2.2 Performing Authentication Calculations and Generating Encryption Keys

The second R-UIM security-related function is to perform authentication calculations and generate encryption keys for use with ME ciphering techniques. See the following figure. This is performed by the Run CAVE function. The settings of the input parameters for the authentication procedure are defined in [5] and [14]. The parameters ESN and IMSI that are used for the Run CAVE function are determined at the time of R-UIM insertion into the ME. If ESN rather than UIMID is chosen (i.e. EF_{USGIND} bit 1 = '0') for the Run CAVE function, the value used for the CAVE algorithm shall be the one received from security commands, regardless of what is stored in $EF_{ESN\_MEID\_ME}$. For details, refer to section 4.6, "ESN and MEID Management Command", and to section 3.4.2, EF_{IMSI_M}.

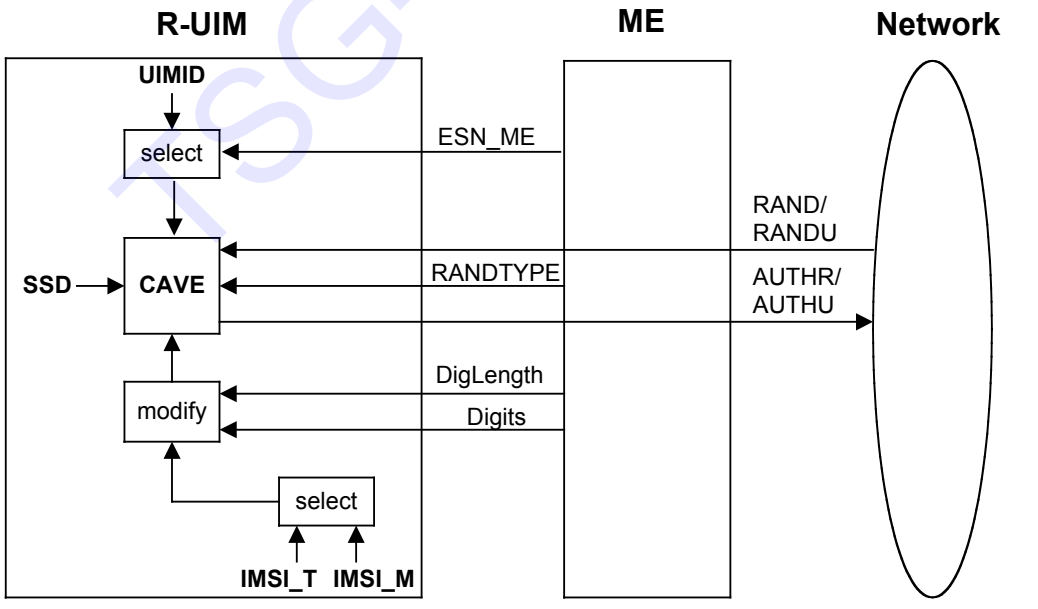


Figure 5. Run CAVE Function

The R-UIM stores both an IMSI_M and an IMSI_T to identify the subscription. The lower 10 digits of each are encoded as 34 bit subsets identified as IMSI_M_S and IMSI_T_S, respectively. These are further subdivided into the 24-bit quantities IMSI_M_S1 and IMSI_T_S1 to identify the coding of the lower 7 digits and the 10-bit quantities IMSI_M_S2 and IMSI_T_S2 to identify the coding of the remaining 3 digits. For the authentication calculation, the 24-bit coding of the lower 7 digits is used for most applications. Furthermore, an 8-bit subset of the coding of the remaining 3 digits may also be used. See Table 2.3.12.1-1 in [5] and Table 6.3.12.1-1 in [14], entitled "Auth_Signature Input Parameters". The IMSI to be used for these calculations is determined at the time of R-UIM insertion into the ME. For details, refer to section 3.4.2, EF_{IMSI_M}.

In order that conformance to [5] and [14] be supported, a 34-bit MIN will be stored in EF_{IMSI_M}. The use of these bits for the calculation of authentication responses shall be as described above.

The Run CAVE command ~~Get Response~~ causes the R-UIM to ~~pass~~ respond with the output AUTHR or AUTHU ("global" challenge response or "unique" challenge response) to the ME. Temporary parameters may be stored on the R-UIM for use in calculating ciphering keys.

The calculation of ciphering keys is performed by execution of the GENERATE KEY/VPM function.

The GENERATE KEY/VPM function is shown in the following figure. This function will produce keys for some of the ciphering mechanisms as specified in [5] and [14]. GENERATE KEY/VPM will process temporary stored parameters that were produced during the calculation of an authentication response by the Run CAVE function and will produce keys. Some may be used directly for ME encryption functions and some may be further processed within the ME for use by the ECMEA and ECMEA_NF encryption functions.

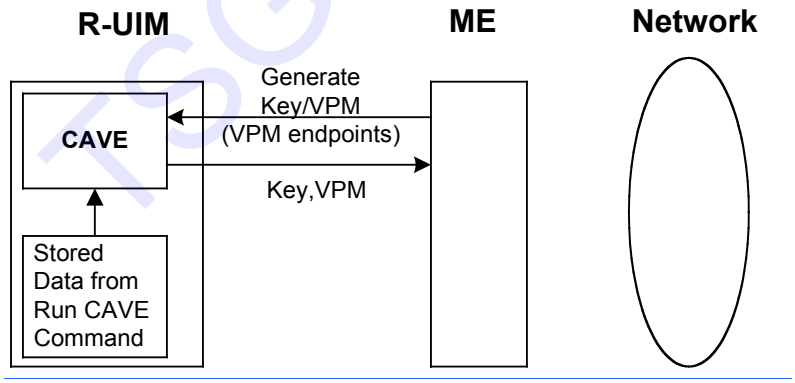


Figure 6. Generate Key/VPM Function

4.2.3 Managing the Call History Parameter

The third security-related function is the generation and management of the call history parameter CALL COUNT. CALL COUNT is used as a simple "clone" detector. During network access protocols, the R-UIM reports its value of CALL COUNT to the network. If the value is consistent with the network's perception of CALL COUNT, the network will

likely grant access based on the authentication process. During the call, the value of CALL COUNT may be incremented upon a command from the network.

If the network determines that a value of CALL COUNT appears to be out of sequence, the network may choose to investigate the possibility that the R-UIM has been “cloned” and take remedial action.

Incrementing and reading the parameter COUNT is accomplished via standard ME-to-R-UIM commands.

4.3 Description of OTASP/OTAPA Functions

A complete description of Over-the-Air Service Provisioning (OTASP) and Over-the-Air Parameter Administration (OTAPA) may be found in [7]. This section highlights the aspects of R-UIM that support OTASP/OTAPA. EFs are described first, followed by [7] "Request/Response" messages that have been mapped to R-UIM commands. In some cases, ME intervention is necessary to accomplish the OTASP/OTAPA functions.

4.3.1 Elementary Files for OTASP/OTAPA

Four EFs are described.

4.3.1.1 EF$_{SPC}$ (Service Programming Code)

The Service Programming Code (SPC) is a simple means to protect the contents of the R-UIM from being programmed without authorization. SPC is described in [7] section 3.3.6.

4.3.1.2 EF$_{OTAPASPC}$ (OTAPA/SPC_Enable)

This EF can be written to and read via the ME. It allows the user to activate OTAPA protection for the NAM on the R-UIM. It also enables the user to allow (or deny) the service provider to change the value of SPC from a default value to a non-default value..

4.3.1.3 EF$_{NAMLOCK}$ (NAM_LOCK)

[7] provides means for "locking" NAM contents under the control of the service provider, with appropriate inputs from the user. This EF stores the current state (locked/unlocked) of the NAM.

4.3.1.4 EF$_{OTA}$ (OTASP/OTAPA Features)

This EF maintains a listing of OTASP/OTAPA features and the associated protocol version for each. The ME reads this EF in order to respond to the "Protocol Capability Request Message" from the network. The ME combines this information with parameters stored in the ME as defined in Sec. 3.5.1.7 of [7] – specifically, its Firmware Revision Number and Manufacturer's Model Number.

4.3.2 Mapping of OTASP/OTAPA Request/Response Messages to R-UIM Commands

The OTASP/OTAPA message pairs are listed in [7]. In some cases, the mapping is one-to-one. In others, the ME intervenes by performing a translation to enable the use of simple R-UIM commands. In still other cases, the ME relies upon security-related commands to prepare a response.

4.3.2.1 Protocol Capability Request/Response Messages

This message requests information that is stored in both the ME and in the R-UIM. The ME reads EF$_{OTA}$ for the list of FEATURE_ID and FEATURE_P_REV pairs that the R-UIM supports, adds information stored in the ME (its Firmware Revision Number and Manufacturer's Model Number) and sends this information to the network to complete the response.

4.3.2.2 MS Key Request Command/Response Messages

This command initiates a Diffie/Hellman key exchange that enables calculation of the "A-key" and/or Root Key. Upon receipt of the MS Key Request message from the network, the ME generates a 160-bit random number called RANDSeed and sends RANDSeed to the R-UIM along with the modulus P and the generator G sent by the network. The R-UIM in turn generates a random number X that may be related to RANDSeed. Then the R-UIM raises G to the power of X, modulo P and temporarily stores the result as MS_RESULT. The R-UIM computes sets the Result Code and sends this in response to the MS KEY REQUEST command. The ME forwards the Result Code to the network to complete this transaction. Details of this process are in sections 3.3.1.5 and 5 of [7].

4.3.2.3 Key Generation Request/Response Messages

This request/response pair completes the ephemeral Diffie/Hellman key exchange. Upon receipt of the Key Generation Request message, the ME sends BS_RESULT to the R-UIM. The R-UIM calculates the Diffie/Hellman result by raising BS_RESULT to the power of X (see section 4.3.2.2), modulo P. A subset of this result is temporarily stored as the A-key and/or Root Key. The R-UIM sets the Result Code and MS RESULT and sends these in the response to the KEY GENERATION REQUEST command. The ME forwards the Result Code and MS RESULT to the network to complete this transaction. Details of this process are in sections 3.3.1.6 and 5 of [7].

4.3.2.4 SSD Update

An SSD Update may be performed as a component of OTASP/OTAPA procedures. This process uses commands and EFs described in other sections of the R-UIM document. The SSD Update procedure that is performed during OTASP/OTAPA uses temporary values of the A-Key and SSD, and does not store these temporary values in semi-permanent memory until the R-UIM accepts the “COMMIT” command. This slight deviation from the procedure in [5] and [14] is accommodated by the setting of bit 3 of the “process control” parameter of the “UPDATE SSD” command to the R-UIM. The R-UIM should reject any Update SSD command and return SW1=‘98’ and SW2=‘34’ (Error, out of sequence) if it is received outside of the context of a key generation procedure.

4.3.2.5 Re-Authentication Request/Response Messages

The ME receives the Re-Authentication Request Message containing the four-octet parameter RAND. The ME constructs the Re-Authentication Response Message by taking the following steps.

(1) Read EF_{COUNT}

(2) Prepare AUTH_DATA (See [7], section 3.3.2)

(3) Truncate RAND to produce RANDC

(4) Compute AUTHR by using the AUTHENTICATE (Run CAVE) command with input parameters:

- RANDTYPE=‘0000 0000’ (i.e., 32 bits)

- RAND=RAND received by ME
- DigLength, DIGITS as specified by AUTH_DATA
- Process Control

  b1: '0' (inactive)

  b2: '0' (inactive)

  b3: '1' (wait for COMMIT before storing A-key, SSD)

  b4: '0' (inactive)

  b5: '1' (save registers)

  b6: '0' (inactive)

  b7: '0' (inactive)

  b8: '0' (inactive)

If message encryption or voice privacy is to be activated, the ME executes the command GENERATE KEY/VPM with the R-UIM.

4.3.2.6 Validation Request/Response Messages

The ME receives the Validation Request Message, which seeks validation of 'NUM_BLOCKS' blocks of data, each block having a length of 'BLOCK_LEN'. In order that R-UIM command coding be simplified, the ME buffers the data into respective blocks, then validates each block via the command VALIDATE, whereby a single block of data having length 'BLOCK_LEN' is validated. For each block, the R-UIM responds with a Result Code. Upon successful execution of the command and depending on the Block ID, the R-UIM shall temporarily store NAM_LOCK$_s$ or SPC$_s$ as specified in section 3.3.1.10 of [7]. The ME then accumulates the R-UIM responses and sends a composite response to the network. The ME should stop sending a VALIDATE (Verify SPC) command to the R-UIM during the same OTASP session after receiving more than five failure responses from the R-UIM as recommended in Sec. 3.4 of [7].

Section 4.5.4 of [7] describes common blocks of data that are validated. These include verification of the SPC, verification that the SPC may be updated by the network and validation of SPASM, whereby AUTH_OTAPA is compared within the R-UIM to an internally-generated value that was calculated as a component of the R-UIM's response to the OTAPA Request command. Thus, the SPASM mechanism requires that an OTAPA Response Message be sent from ME to network prior to the Validation Request message.

4.3.2.7 Configuration Request Command/Response Messages

The ME receives the Configuration Request command, which requests configuration details of 'NUM_BLOCKS' of data, each block having a length of 'BLOCK_LEN'. In order that R-UIM command coding be simplified, the ME buffers the request into 'NUM_BLOCK' single block requests, then asks for configuration details for each block via the CONFIGURATION REQUEST command to the R-UIM. For each block, the R-UIM responds with the Block ID, Block Length, Result Code and Parameter Data (see sections 3.3.1.1, 3.5.1.1 and 4.5.1.1 of [7]). The ME accumulates the set of block responses and sends a composite response to the network. Note that the R-UIM shall use ME-specific parameters (i.e. SCM, MOB_P_REV and Local Control) stored in the EF$_{MECRP}$ to generate a response.

4.3.2.8 Download Request/Response Messages

The ME receives the Download Request Message, which attempts to download 'NUM_BLOCKS' of data to the R-UIM, each block having a Block ID, Block Length and Parameter Data of length 'Block Length'. In order that R-UIM command coding be simplified, the ME buffers the request into NUM_BLOCK single block requests, then attempts to download each block via the DOWNLOAD REQUEST command to the R-UIM. Prior to issuance of multiple DOWNLOAD REQUEST commands, the ME may query appropriate EF data to determine if adequate storage space exists in the R-UIM EFs to successfully complete the downloading operation. For each execution of the DOWNLOAD REQUEST command, the R-UIM returns the Block ID and Result Code (see sections 3.3.1.2, 3.5.1.2 and 4.5.1.2 of [7]). Upon successful execution of the command, the R-UIM shall temporarily store the data. The ME accumulates the set of block responses and sends a composite response to the network.

4.3.2.9 SSPR Configuration Request/Response Messages

The network asks for SSPR data stored in a particular area of the R-UIM. The R-UIM responds with Block ID, Result Code, Block Length and Parameter Data (see sections 3.3.1.8, 3.5.1.8 and 4.5.1.8 of [7]).

If Block ID = '0000 0000' or '0000 0001', the R-UIM uses EF_{PRL}. If Block ID = '0000 0010', the R-UIM uses EF_{EPRL} if present.

4.3.2.10 SSPR Download Request/Response Messages

The network attempts to download SSPR data into the R-UIM. The data contains a Block ID, a Block Length and Parameter Data having 'Block Length' size. If the MS receives 254 or 255 bytes for the Parameter Data from the network (instead of 253 or less), then the ME shall send to the R-UIM two commands - since the maximum length Param Data that the ME can send is 253 bytes. The R-UIM responds with the Block ID, Result Code, Segment Offset and Segment Size, as described in sections 3.3.1.9, 4.5.1.9 and 3.5.1.9 of [7]. Upon successful execution of the command, the R-UIM shall temporarily store the data.

4.3.2.11 OTAPA Request/Response Messages

If Block ID = '0000 0000', the R-UIM updates EF_{PRL} (and EF_{CSSPR} if present) after the R-UIM receives and successfully executes a COMMIT command. If Block ID = '0000 0001', the R-UIM update, if present, EF_{EPRL} and EF_{CSSPR} after the R-UIM receives and successfully executes a COMMIT command.OTAPA Request/Response Messages

The network attempts to initiate OTAPA by sending an "OTAPA Request Message" containing the "start/stop" parameter. The ME in turn passes this to the R-UIM, along with a 32-bit ME-generated random number RANDSeed. If service n9 is allocated and activated and ME is assigned with MEID, the ME also passes pESN to the R-UIM. The R-UIM generates its own random number RAND_OTAPA which may be related to RANDSeed. Also, the R-UIM computes a value for AUTH_OTAPA as described in [7], section 3.3.7.

If the OTAPA feature is disabled by the user (as defined by the OTAPA_Enable bit in $EF_{OTAPASPC}$ and Sec. 3.2.2 [7]), the ME shall not send OTAPA REQUEST to the R-UIM.

4.3.2.12 Commit Command/Response Messages

The network sends a "Commit Request Message" to the R-UIM via the ME. The ME translates this to the R-UIM command COMMIT. The R-UIM responds with the Result Code which the ME forwards to the network via the "Commit Response Message". Upon successful execution of the command, the R-UIM shall move temporarily stored data to semi-permanent memory, i.e. to the appropriate EF(s) (as specified in sections 3.3.1.3, 3.5.1.6 and 4.5.1.7 of [7]).

4.3.2.13 PUZL Configuration Request/Response Messages

The network asks for PUZL data stored in a particular area of the R-UIM. The R-UIM responds with Block ID, Result Code, Block Length and Parameter Data (see sections 3.3.1.12, 3.5.1.12 and 4.5.1.12 of [7]).

4.3.2.14 PUZL Download Request/Response Messages

The network attempts to download PUZL data into the R-UIM. The data contains a Block ID, a Block Length and Parameter Data having 'Block Length' size. The R-UIM responds with the Block ID, Result Code, Identifier Present Flag, User Zone ID and User Zone System ID, as described in sections 3.3.1.13, 4.5.1.13 and 3.5.1.13 of [7] Upon successful execution of the command, the R-UIM shall temporarily store the data.

4.3.2.15 3GPD Configuration Request/Response Messages

The ME receives the 3GPD Configuration Request Message which requests configuration details of 'NUM_BLOCKS' of data with each block having a length of 'BLOCK_LEN'. In order that R-UIM command coding be simplified, the ME buffers the request into 'NUM_BLOCK' single block requests, then asks for configuration details for each block via the 3GPD CONFIGURATION REQUEST command to the R-UIM. For each block, the R-UIM responds with the Block ID, Block Length, Result Code and Parameter Data (see sections 3.3.1.14, 3.5.1.14 and 4.5.1.14 of [7]). The ME accumulates the set of block responses and sends a composite response to the network. If the 3GPD CONFIGURATION REQUEST command contains a BLOCK_ID for SimpleIP PAP SS Parameters, SimpleIP CHAP SS Parameters, MobileIP SS Parameters or HRPD Access Authentication CHAP SS Parameters, the R-UIM shall check if the Secure Mode is active. If the Secure Mode is not active, then the R-UIM shall return SW1='69' and SW2='82' (Security status not satisfied [55]).

4.3.2.16 3GPD Download Request/Response Messages

The ME receives the 3GPD Download Request Message which attempts to download 'NUM_BLOCKS' of data to the R-UIM, each block having a Block ID, Block Length and Parameter Data of length 'Block Length'. In order that R-UIM command coding be simplified, the ME buffers the request into NUM_BLOCK single block requests, then attempts to download each block via the 3GPD Download Request command to the R-UIM. The ME may query appropriate EF data to determine if adequate storage space exists in the R-UIM EFs to successfully complete the downloading operation, prior to issuance of multiple Download Request commands. For each execution of the 3GPD DOWNLOAD REQUEST command, the R-UIM returns the Block ID and Result Code (see sections 3.3.1.15, 3.5.1.15 and 4.5.1.15 of [7]). Upon successful execution of the command, the R-UIM shall temporarily store the data. The ME accumulates the set of block responses and sends a composite response to the network. If the 3GPD DOWNLOAD REQUEST command contains a BLOCK_ID for SimpleIP PAP SS Parameters, SimpleIP CHAP SS Parameters, MobileIP SS Parameters or HRPD Access Authentication CHAP SS Parameters, the R-UIM shall check if the Secure Mode is active. If the Secure Mode is not active, then the R-UIM shall return SW1='69' and SW2='82' (Security status not satisfied [55]).

4.3.2.17 Secure Mode Request/Response Messages

This is the command that causes the R-UIM to generate Secure Mode Ciphering Key (SMCK). The R-UIM shall use the SMCK as a key for encryption and decryption of all

PARAM-DATA of all Parameter Blocks sent and received by the R-UIM in the OTASP Data Messages while the Secure Mode is active.

The network can initiate the Secure Mode by sending Secure Mode Request Message to the ME with the START_STOP field set to '1'. Upon receipt of the Secure Mode Request Message with the START_STOP filed set to '1', the ME translates this to the SECURE MODE command. The R-UIM shall use RAND_SM received in this command and the SSD to compute the SMCK as described in [7], section 3.3.8.1 and then the R-UIM responds with Result Code, which the ME forwards to the network via the "Secure Mode Response Message". While the Secure Mode is active, the ME shall send a FRESH command to the R-UIM prior to sending any commands when it receives one of the following messages;

- Configuration Request Messages
- SSPR Configuration Request Message
- PUZL Configuration Request Message
- 3GPD Configuration Request Message
- Download Request Messages
- SSPR Download Request Message
- PUZL Download Request Message
- 3GPD Download Request Message
- MMD Configuration Request Message
- MMD Download Request Message
- MMS Configuration Request Message
- MMS Download Request Message
- System Tag Configuration Request Message
- System Tag Download Request Message

For the configuration request messages, the ME sends the FRESH command to the R-UIM to request a 15-bit FRESH value selection. This can be selected at random or can be set to a monotonically increasing counter. The R-UIM responds with the FRESH value.

For the download request messages, the ME sends the FRESH command to R-UIM to pass the FRESH value received from the network.

The network can terminate the Secure Mode by sending Secure Mode Request Message to the ME with the START_STOP field set to '0'. Upon receipt of the Secure Mode Request Message with the START_STOP filed set to '0', the ME translates this to the SECURE MODE command. The R-UIM responds with Result Code, which the ME forwards to the network via the "Secure Mode Response Message" (see sections 3.3.1.16, 3.5.1.16 and 4.5.1.16 of [7]).

4.3.2.18 Service Key Generation Request/Response Messages

This is the command that causes the R-UIM to generate Service keys, such as BCMCS, IMS, WLAN, etc. R-UIM shall generate an intermediate key based on the root key before using it to generate service keys. Details of this process are in [7], section 3.3.10. See also sections 3.3.1.21, 3.5.1.22 and 4.5.1.22 of [7].

4.3.2.19 MMD Configuration Request/Response Messages

The network asks for MMD data stored in a particular area of the R-UIM. The R-UIM responds with Block ID, Result Code, Block Length and Parameter Data (see sections 3.3.1.17, 3.5.1.18 and 4.5.1.18 of [7]).

4.3.2.20 MMD Download Request/Response Messages

The network attempts to download MMD data into the R-UIM. The data contains a Block ID, a Block Length and Parameter Data having 'Block Length' size. The R-UIM responds with the Block ID and Result Code as described in sections 3.3.1.18, 4.5.1.19 and 3.5.1.19 of [7]. Upon successful execution of the command, the R-UIM shall temporarily store the data.

4.3.2.21 MMS Configuration Request/Response Messages

The network asks for MMS data stored in a particular area of the R-UIM. The R-UIM responds with Block ID, Result Code, Block Length and Parameter Data (see sections 3.3.1.22, 3.5.1.23 and 4.5.1.23 of [7]).

4.3.2.22 MMS Download Request/Response Messages

The network attempts to download MMS data into the R-UIM. EF_{MMSICP} (MMS Issuer Connectivity Parameters) should be updated. The data contains a Block ID, a Block Length and Parameter Data having 'Block Length' size. The R-UIM responds with the Block ID and Result Code as described in sections 3.3.1.23, 4.5.1.24 and 3.5.1.24 of [7]. Upon successful execution of the command, the R-UIM shall temporarily store the data.

4.3.2.23 System Tag Configuration Request/Response Messages

The network asks for System Tag data stored in a particular area of the R-UIM. The R-UIM responds with Block ID, Result Code, Block Length and Parameter Data. Parameters are formatted as in sections 3.3.1.19, 3.5.1.20 and 4.5.1.20 of [7].

4.3.2.24 System Tag Download Request/Response Messages

The network attempts to download System Tag data into the R-UIM. The data contains a Block ID, a Block Length and Parameter Data having 'Block Length' size. The R-UIM responds with the Block ID, Result Code, Segment Offset and Segment Size, as described in sections 3.3.1.20, 3.5.1.21 and 4.5.1.21 of [7]. Upon successful execution of the command, the R-UIM shall temporarily store the data.

4.4 Description of Security-Related Commands

The commands BASE STATION CHALLENGE, UPDATE SSD and CONFIRM SSD are performed in sequence as described in Annex D. If either UPDATE SSD or CONFIRM SSD are received out of sequence, the card shall return SW1='98' and SW2='34' (Error, out of sequence). In this case, the ME shall abandon the sequence of commands and shall re-start the sequence of commands starting with Base Station Challenge if the ME performs the sequence of commands again. If the R-UIM receives a Base Station Challenge command, it shall re-start the command sequence. If T=0 protocol is used, APDU is mapped onto TPDU (see Section 9.1 in [17])

In the procedures described in Sections 4.4.1 through 4.4.5; RANDSSD, RANDSeed, RANDBS, AuthBS, RAND, RANDU, AUTHR and AUTHU are encoded with the highest-order octet first. ESN_ME is encoded with the lowest-order octet first to match the coding for $EF_{ESN\_MEID\_ME}$.

4.4.1 Update SSD

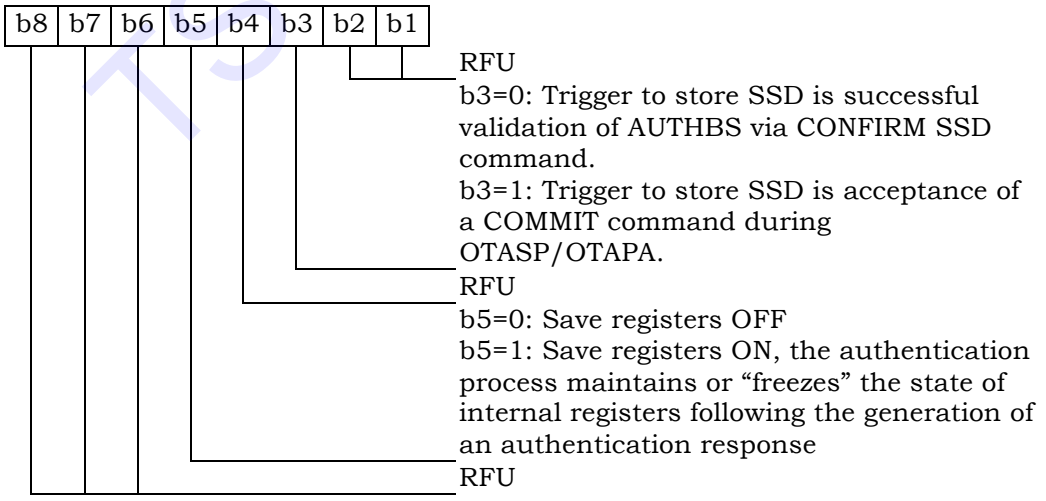
COMMAND	CLASS	INS	P1	P2	Lc	Le
UPDATE SSD	'A0'	'84'	'00'	'00'	'0F'	absent

Command parameters/data:

Octet(s)	Description	Length
1 – 7	RANDSSD	7 bytes
8	Process_Control	1 byte
9 – 15	ESN_ME	7 bytes

The input parameter Process_Control is coded as follows:

Octet 8:



Bit 3 of Process_Control specifies the trigger that causes the newly calculated value of SSD to become stored in semi-permanent memory. If b3 = '0', the trigger is a successful

validation of AUTHBS via a CONFIRM SSD command. If b3 = '1', the trigger is the acceptance of a COMMIT command during OTASP/OTAPA.

The use of bit 5 is only relevant to the AUTHENTICATE (Run CAVE) command, in which the generation of keys may follow the generation of an authentication response.If EF_{USGIND} bit 1 is set to '0', then the R-UIM shall use the value in the ESN_ME field as an input to the CAVE algorithm.

Otherwise, if the EF_{USGIND} bit 1 is set to '1', then the R-UIM shall ignore the value in the ESN_ME field.

The ESN_ME field is coded with the 4-byte ESN_ME which occupies Octets 9 to 12. Octets 13 – 15 shall be set to '00 00 00'.

Response parameters/data:

No response parameters are generated as a result of command execution. The appropriate SW1 and SW2 shall be returned.

4.4.2 BASE STATION CHALLENGE

COMMAND	CLASS	INS	P1	P2	Lc	Le
BASE STATION CHALLENGE	'A0'	'8A'	'00'	'00'	'04'	'04'

Command parameters/data:

Octet(s)	Description	Length
1 – 4	RANDSeed	4 bytes

Response parameters/data:

Octet(s)	Description	Length
1 – 4	RANDBS	4 bytes

4.4.3 CONFIRM SSD

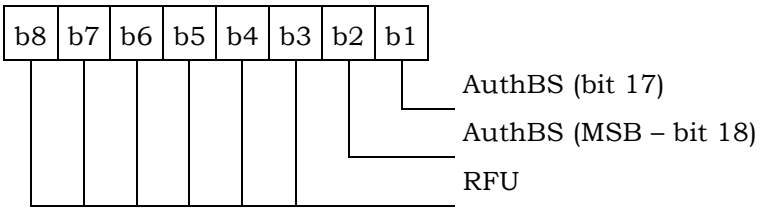
COMMAND	CLASS	INS	P1	P2	Lc	Le
CONFIRM SSD	'A0'	'82'	'00'	'00'	'03'	absent

Command parameters/data:

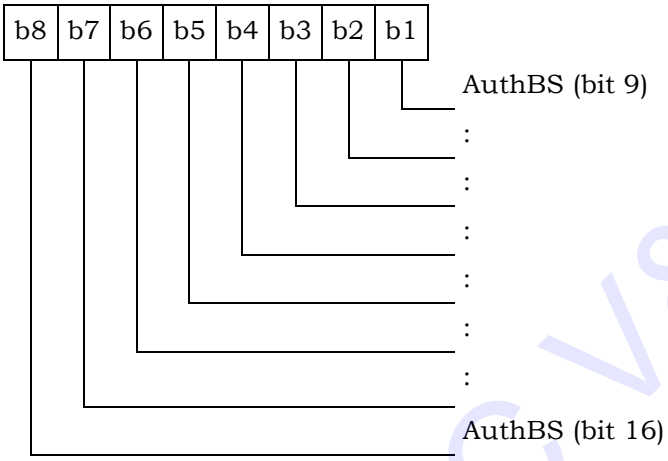
Octet(s)	Description	Length
1 – 3	AuthBS	3 bytes

AuthBS shall be coded as follows:

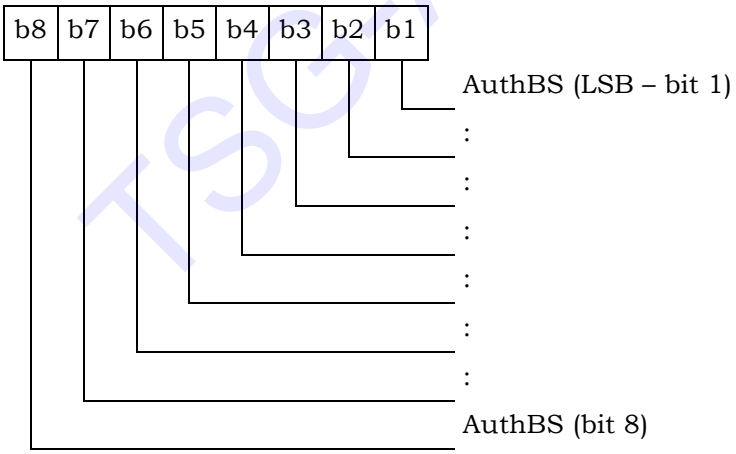
Octet 1:



Octet 2:



Octet 3:



Response parameters/data:

No response parameters are generated as a result of command execution. Successful comparison will cause SW1 to be set to '90' and SW2 to be set to '00'. Unsuccessful comparison will cause SW1 to be set to '98' and SW2 to be set to '04' (Authentication failed [17]).

If the ME is assigned an MEID and if bit1 of the EF_{USGIND} is set to '0', then the pESN value received in the UPDATE SSD command shall be used as the ESN input to the CAVE algorithm for the computation of AuthBS.

4.4.4 AUTHENTICATE

This command performs authentication functions.

COMMAND	CLASS	INS	P1	P2	Lc	Le
AUTHENTICATE	‘A0’	‘88’	P1	‘00’	‘XX’	‘YY’

P1 parameter defines the authentication command type:

P1	Meaning	XX	YY
‘00’	Run CAVE	‘11’	‘03’
‘01’	3G Access AKA	Variable	Variable
‘02’	EAP AKA	Variable	Variable

P1= ‘00’: 2G Authentication-Run CAVE

Command parameters/data:

Octet(s)	Description	Length
1	RANDTYPE (RAND/RANDU)	1 byte
2 – 5	RAND/RANDU	4 bytes
6	DigLength (expressed in bits)	1 byte
7 – 9	Digits	3 bytes
10	Process_Control	1 byte
11 – 17	ESN_ME	7 bytes

The parameter RANDTYPE is coded as follows:

‘0000 0000’ RAND (global random challenge)

‘0000 0001’ RANDU (unique random challenge)

All other values of RANDTYPE are reserved for future use.

If the RANDTYPE is set to RAND, then the RAND occupies octets 2-5. If the RANDTYPE is set to RANDU, then the RANDU occupies octets 3-5 and octet 2 is ignored.

If there are no digits for input to CAVE (e.g., for Registration or Unique Challenge), then DigLength = ‘00’ and Octets 7-9 = ’00 00 00’. If digits are included, bits b1 to b4 of Octet 9 encode the least significant digit, the next least significant digit is encoded in bits b5 to b8 of Octet 9, the next least significant digit is encoded in bits b1 to b4 of Octet 8 and so on to Octet 7. If less than 6 digits are input, then Octets 7-9 are zero padded. For example, if the digits are “123”, then

Byte 6 = ‘0000 1100’, (Note: 3 digits at 4 bits per digit is 12 bits)

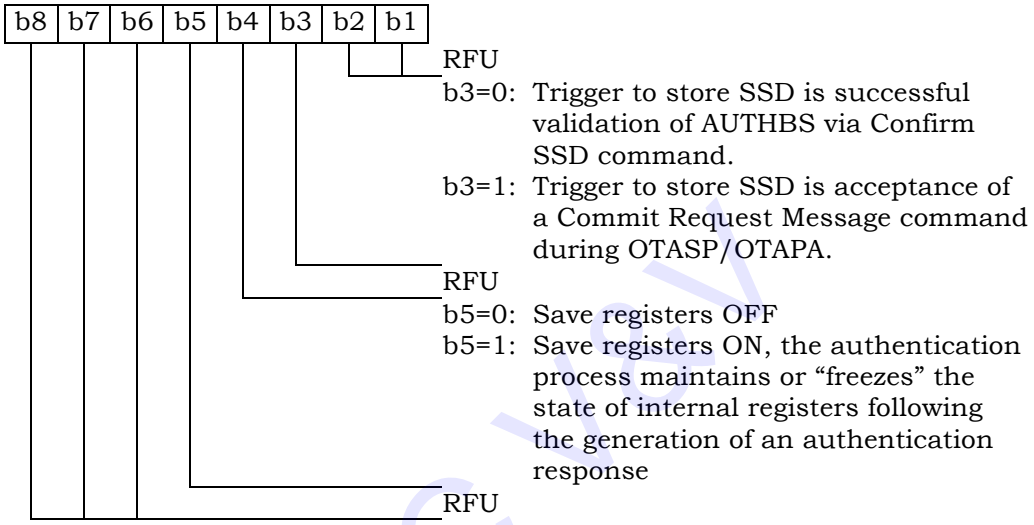
Byte 7 = ‘0000 0000’,

Byte 8 = ‘0000 0001’ and

Byte 9 = ‘0010 0011’.

The input parameter Process_Control is coded as follows:

Octet 10:



‘0’.

The ESN_ME field is coded with the 4-byte ESN_ME which occupies Octets 11 to 14. Octets 15 to 17 should be set to ‘00 00 00’. If EF_{USGIND} bit 1 is set to 0, then the R-UIM shall use the value in the ESN_ME field as the ESN input to the CAVE algorithm.

Response parameters/data:

Octet(s)	Description	Length
1 – 3	AUTHR/AUTHU	3 bytes

P1= ‘01’: 3G Access AKA

Upon receiving this AUTHENTICATE command, the R-UIM either generates the AKA security parameters: IK, CK, RES, UAK if supported, by using the Root Key or sends an AUTS if sequence number resynchronization is necessary. See section 2.3.12.5.2 of [5].

Command parameters/data:

Octet(s)	Description	Length
1-16	RANDA	16 bytes
17	Length of AUTN (L1)	1 byte
18-18+L1	AUTN	L1 bytes

Where AUTN = SQN⊕AK||AMF||MAC-A

Response parameters/data:

Octet(s)	Description	Length
1	Synchronization Failure Tag	1 byte
Either		
2 – 17	Cipher Key	16 bytes
18 – 33	Integrity Key	16 bytes
34	RES Length	1 byte
35 to 35+RES Length-1	RES	RES Length
Or		
2-15	AUTS	14 bytes

If the R-UIM detects the sequence numbers to be invalid, the R-UIM shall set synchronization failure tag to '00000001' and include AUTS. Otherwise, the R-UIM shall set synchronization failure tag to '00000000' and include CK, IK, RES Length and RES. All the other values are reserved.

If MACA comparison fails, the R-UIM returns status words SW1 = '98' and SW2 = '04' (Authentication failure [17]).

RES Length field shall be set to the length of RES, and it has to be greater or equal to 1.

IK, CK and UAK, if calculated, are stored in temporary memory until the CONFIRM_KEYS command is received.

P1= '02': EAP AKA (WLAN Authentication)~~WLAN Authentication – EAP AKA~~

Upon receiving this AUTHENTICATE command, the R-UIM either generates IK, CK, RES, UAK if supported, by using WLAN Root Key or sends an AUTS if sequence number resynchronization is necessary. See [42] and [59].

Command parameters/data:

Octet(s)	Description	Length
1-16	RANDA	16 bytes
17	Length of AUTN (L1)	1 byte
18-18+L1	AUTN	L1 bytes

Where AUTN = SQN⊕AK||AMF||MAC-A

Response parameters/data:

Octet(s)	Description	Length
1	Synchronization Failure Tag	1 byte
Either		
2 – 17	Cipher Key	16 bytes
18 – 33	Integrity Key	16 bytes
34	RES Length	1 byte
35 to 35+RES Length-1	RES	RES Length
or		
2-15	AUTS	14 bytes

If the R-UIM detects the sequence numbers to be invalid, the R-UIM shall set synchronization failure tag to '00000001' and include AUTS. Otherwise, the R-UIM shall set synchronization failure tag to '00000000' and include CK, IK, RES Length and RES. All the other values are reserved.

If MACA comparison fails, the R-UIM returns status words SW1 = '98' and SW2 = '04' (Authentication failure [17]).

RES Length field shall be set to the length of RES, and it has to be greater or equal to 1.

IK, CK and UAK, if calculated, are stored in temporary memory until the CONFIRM_KEYS command is received.

4.4.4.1 Advisory Note on the Use of Run CAVE

In early versions of R-UIM specifications, the AUTHENTICATE (Run CAVE) command was used to perform both the calculations of authentication responses and the generation of ciphering keys. As [14/15] systems continue to evolve, it became necessary to partition the tasks of authentication and cipher key generation among several commands.

The AUTHENTICATE (Run CAVE command as shown is used to generate authentication responses and to enable the calculation of ciphering keys upon the invocation of a subsequent command.

If ciphering keys are to be generated, the AUTHENTICATE (Run CAVE) command should carry the input parameter Process_Control with bit 5 set to ON ('1'). Once the authentication response has been ~~delivered via the Get Response command~~received by the ME, a cipher key generation command may be issued. This will perform key generation calculations that are based upon the "saved" parameters that were stored upon the execution of the AUTHENTICATE (Run CAVE) command with bit 5 of the Process_Control octet set to ON.

4.4.4.2 Use of Cipher Key Generation Command

The command GENERATE KEY/VPM may be invoked at any time following the AUTHENTICATE (Run CAVE) command with the "save" function ON. One or more instances of AUTHENTICATE (Run CAVE) command may be performed with the "save

registers" function OFF during the intervening time period, but the input parameters to the GENERATE KEY/VPM will be those values that were stored upon the most recent invocation of the AUTHENTICATE (Run CAVE) command with the "save registers" function turned ON. The response to GENERATE KEY/VPM will ~~provide~~ contain a fixed-length 64-bit key along with a VPM of ME-specified length to the ME ~~upon the execution of the Get Response command~~.

4.4.5 Generate Key/VPM

This command relies on the prior successful execution of the AUTHENTICATE (Run CAVE) command with the "save" function activated. If this has not occurred, the status words SW1='98' and SW2='34' (Error, out of sequence) shall be returned upon the invocation of this command.

COMMAND	CLASS	INS	P1	P2	Lc	Le
GENERATE KEY/VPM	'A0'	'8E'	'00'	'00'	'02'	*

Command parameters/data:

Octet(s)	Description	Length
1	First octet of VPM to be output	1 byte
2	Last octet of VPM to be output	1 byte

Details value:

Octet(s)		Description of the choice for the VPM to be output.	Length
1	2		
'XX'	'YY'	Retrieve the (YY-XX+1) length of the VPM to be output	(YY-XX+1) bytes
'FF'	'FF'	No VPM to be output	0 byte

If VPM output is present, then the range of 'XX' and 'YY' shall be between '00' and '40', and 'XX' ≤ 'YY'. If the entire VPM of length 520 bits (or 65 bytes) [20] is desired, 'XX' and 'YY' shall be set to, respectively, '00' and '40'.

Response parameters/data:

Octet(s)	Description	Length
1 – 8	Key	8 bytes
9 –	VPM octets from 'XX'	*

- The number of VPM octets varies as specified by command parameter.

4.5 Description of OTASP/OTAPA Commands

4.5.1 MS KEY REQUEST

The purpose of this command is described in 4.3.2.2.

COMMAND	CLASS	INS	P1	P2	Lc	Le
MS KEY REQUEST[6]	'A0'	'50'	'00'	'00'	*	'01'

Command parameters/data:

Octet(s)	Description	Length
1 – 20	RANDSeed	20 bytes
21	A-key Protocol Revision	1 byte
22	Parameter P Length	1 byte
23	Parameter G Length	1 byte
24 – X	Parameter P	Parameter P Length
X+1 to Y	Parameter G	Parameter G Length

*If A-key Protocol Revision is greater than '00000010', Parameter P Length and Parameter G Length shall be set to '00000000' and the Parameter P and G shall be omitted.
Details of command parameters are in [7], section 4.5.1.3, "MS Key Request Message".

Response parameters/data:

Octet(s)	Description	Length
1	Result Code	1 byte

Details of the response are in [7], sections 3.3.1.5, "MS Key Request Message Processing" and 3.5.1.3, "MS Key Response Message".

6 This command was previously called "Generate Public Key".

4.5.2 KEY GENERATION REQUEST

The purpose of this command is described in 4.3.2.3.

COMMAND	CLASS	INS	P1	P2	Lc	Le
KEY GENERATION REQUEST	'A0'	'52'	'00'	'00'	*	**

Command parameters/data:

Octet(s)	Description	Length
1	BS Result Length	1 byte
2 – Lc	BS Result	Lc – 1 bytes

- Note: Lc=Length of BS Result in octets + 1,

Details of command parameters are in [7], section 4.5.1.4.

Response parameters/data:

Octet(s)	Description	Length
1	Result Code	1 byte
2	MS Result Length	1 byte
3 – Le	MS Result	Le – 2 bytes

** Note: Le=Length of MS Result + 2

Details of the response are in [7], sections 3.3.1.6 and 3.5.1.4.

4.5.3 COMMIT

COMMAND	CLASS	INS	P1	P2	Lc	Le
COMMIT	'A0'	'CC'	'00'	'00'	empty	'01'

Response parameters/data:

Octet(s)	Description	Length
1	Result Code	1 byte

Details of the Commit Request and Response are in [7], sections 3.3.1.3, 4.5.1.6 and 3.5.1.6, respectively.

If one or more DOWNLOAD REQUEST commands with Block ID = '00' or '02' were received with an IMSI_M that has a zero value (all digits are zero), then the R-UIM shall set IMSI_M_PROGRAMMED to '0' in EF_{IMSI_M}. If IMSI_M has a non-zero value, the R-UIM shall set IMSI_M_PROGRAMMED to '1'.

If one or more DOWNLOAD REQUEST commands with Block ID = '03' were received with an IMSI_T that has a zero value (all digits are zero), then the R-UIM shall set IMSI_T_PROGRAMMED to '0' in EF_{IMSI_T}. If the IMSI_T has a non-zero value, the R-UIM shall set IMSI_T_PROGRAMMED to '1'.

4.5.4 VALIDATE

COMMAND	CLASS	INS	P1	P2	Lc	Le
VALIDATE	'A0'	'CE'	'00'	'00'	*	'02'

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Block Length	1 byte
3 – Lc	Param Data	Lc – 2 bytes

This command requests validation of a single block of data and forms a subset of the "Validation Request Message" as described in [7], section 4.5.1.10.

- Note: Lc = Length of Param Data + 2

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Result Code	1 byte

This response pertains to a single block of data and forms a subset of the "Validation Response Message" as described in [7], sections 3.3.1.10 and 3.5.1.10.

As defined in Sections 3.2.2.3 and 3.3.1.10 Validation Request Message Processing of [7], SP_LOCK_STATE is initially set at the start of an OTASP programming session and shall be set according to the following conditions:

1. It is set to '0' if
   a. the R-UIM does not support Service Programming Lock, or
   b. the R-UIM supports Service Programming Lock and bit 1 of EF_{SPCS} is set to '0'.
2. It is set to '1' otherwise.

Note that, during the OTASP session:

1. SP_LOCK_STATE cannot change from '0' to '1' and
2. SP_LOCK_STATE can change from '1' to '0' after the R-UIM receives and successfully executes a VALIDATE (Verify SPC) command as described in Annex E .

4.5.5 CONFIGURATION REQUEST

COMMAND	CLASS	INS	P1	P2	Lc	Le
CONFIGURATION REQUEST	'A0'	'54'	'00'	'00'	01	*

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte

This command requests configuration details of a single block of data and forms a subset of the "Configuration Request Message" as described in [7], section 4.5.1.1.

The ME shall not send a CONFIGURATION REQUEST with a Block ID = '04' (eHRPD_IMSI) to the R-UIM (this Block ID is used for the Network-MS interface and not needed for the ME-R-UIM interface). The mapping of CONFIGURATION REQUEST parameters for each Block ID to the various EFs are provided in the tables below. For Block ID = '00' (CDMA / Analog NAM) and '02' (CDMA NAM), the ME can derive MAX_SID_NID by 1) sending SELECT $EF_{CDMAHOME}$ and 2) setting MAX_SID_NID to the file size (bytes 3 to 4 from the response) divided by 5.

See Annex F for parameter to EF mapping.

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Block Length	1 byte
3	Result Code	1 byte
4 – Le	Param Data	Le – 3 bytes

- Note: Le = Length of Param Data + 3.

This response provides configuration details of a single block of data and forms a subset of the "Configuration Response Message" as described in [7], sections 3.3.1.1 and 3.5.1.1.

4.5.6 DOWNLOAD REQUEST

COMMAND	CLASS	INS	P1	P2	Lc	Le
DOWNLOAD REQUEST	'A0'	'56'	'00'	'00'	*	'02'

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Block Length	1 byte
3 – Lc	Param Data	Lc – 2 bytes

This command requests the download of a single block of data and forms a subset of the "Download Request Message" as described in [7], section 4.5.1.2.

- Note: Lc = Length of Param Data + 2

The ME shall not send a DOWNLOAD REQUEST with a Block ID = '04' (eHRPD_IMSI) to the R-UIM (this Block ID is used for the Network-MS interface and not needed for the ME-R-UIM interface). The mapping of the DOWNLOAD REQUEST parameters to the EFs where they are stored is described in section 4.5.5. If the received data includes SID/NID pairs (in $EF_{CDMAHOME}$), the R-UIM shall retain only the SID/NID pairs from the most recently received message.

Note: if Block ID = '00', '02' or '03', the DOWNLOAD REQUEST command in conjunction with COMMIT updates IMSI_M_PROGRAMMED in EF_{IMSI_M} or IMSI_T_PROGRAMMED in EF_{IMSI_T} as described in sections 4.5.3 and 4.5.5.

See Annex F for a description of parameter to EF mapping.

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Result Code	1 byte

This response pertains to a single block of data and forms a subset of the "Download Response Message" as described in [7], sections 3.3.1.2 and 3.5.1.2.

4.5.7 SSPR CONFIGURATION REQUEST

COMMAND	CLASS	INS	P1	P2	Lc	Le
SSPR CONFIGURATION REQUEST	'A0'	'EA'	'00'	'00'	'04'	*

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2 – 3	Request Offset	2 bytes
4	Request Max Size	1 byte

Note: If Block ID = '0000 0001' (Preferred Roaming List) then octets 2 through 4 are used as inputs for this command. For other Block IDs, octets 2 through 4 are ignored.

Details of command parameters are in [7], section 4.5.1.8, "SSPR Configuration Request Message".

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Result Code	1 byte
3	Block Length	1 byte
4 – Le	Param Data	Le – 3 bytes

- Note: Le=Length of Param Data + 3.

Details of the response are in [7], sections 3.3.1.8, "SSPR Configuration Request Message Processing" and 3.5.1.8, "SSPR Configuration Response Message". The PR_LISTS-P in [7] maps to EF_{PRL} if Block ID = '0000 0001'.

Note: If Block ID = '0000 0010' (Extended Preferred Roaming List Dimensions), EF_{EPRL} is not present and EF_{PRL} is present, then the R-UIM sets CUR_SSPR_P_REV to '01' in Param Data (which is PARAM_DATA in [7]) to return the PRL dimensions in the response as defined in Sec. 3.5.3.3 of [7].

4.5.8 SSPR DOWNLOAD REQUEST

COMMAND	CLASS	INS	P1	P2	Lc	Le
SSPR DOWNLOAD REQUEST	'A0'	'EC'	'00'	'00'	*	'05'

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Block Length	1 byte**
3 – Lc	Param Data	Block Length

* Note: Lc=Length of Command parameters/data.

** Note: Block Length = length of Param Data. The maximum value for Block Length is 253.

Details of the command parameters are in [7], section 4.5.1.9, "SSPR Download Request Message". While [7] defines a maximum PRL parameter block data size of 255 bytes, Lc has a maximum value of 255 and there is a 6 byte overhead consisting of Block ID, Block Length, Reserved, Last Segment, Segment Size and Segment Offset (See Sec. 4.5.3 of [7]). This results in a maximum PRL parameter block data size of 249 bytes that the ME can send to the R-UIM. The PR_LIST_{S-P} in [7] maps to EF_{PRL} if Block ID = '0000 0000' and to EF_{EPRL} if Block ID = '0000 0001'.

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Result Code	1 byte
3 – 4	Segment Offset (see note below)	0 or 2 bytes
5	Segment Size (see note below)	0 or 1 byte

- Details of the response are in [7], sections 3.3.1.9, "SSPR Download Request Message Processing" and 3.5.1.9, "SSPR Download Response Message".

- Note: If the Block ID is not '00000000' or '00000001', then the Segment Offset and Segment Size should not be included in the response.

4.5.9 OTAPA REQUEST

COMMAND	CLASS	INS	P1	P2	Lc	Le
OTAPA REQUEST	'A0'	'EE'	'XX'	'00'	'YY'	'06'

Depending on certain conditions, P1 is set to either '00' or to '01'.

P1 is set to '00' if *any* of the following conditions hold:

- ME is not assigned an MEID_ME;
- ME is assigned an MEID_ME but service n9 is not activated;
- EF_{USGIND} bit 1 is set to '1';

If P1 = '00'

Command parameters/data:

Octet(s)	Description	Length
1	Start/Stop	1 byte
2 – 5	RANDSeed	4 bytes

YY (Lc) = 5.

P1 is set to '01' if *all* of the following conditions hold:

- ME is assigned an MEID_ME;
- Service n9 is activated;
- EF_{USGIND} bit 1 is set to '0'.

If P1 = '01'

Command parameters/data:

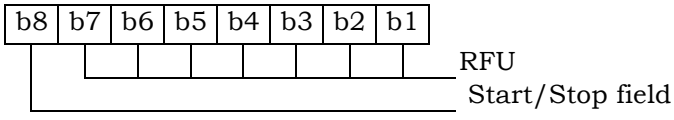
Octet(s)	Description	Length
1	Start/Stop	1 byte
2 – 5	RANDSeed	4 bytes
6-12	pESN	7 bytes

YY (Lc) = 12.

Note: The pESN is actually a four byte identifier which occupies Octets 6 to 9. Octets 10 – 12 should be set to '00 00 00'.

The Start/Stop parameter as defined in Section 4.5.1.11 of [7] shall be coded as follows:

Octet 1



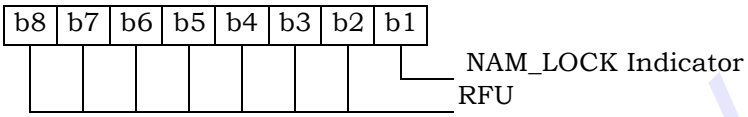
Response parameters/data:

Octet(s)	Description	Length
1	Result Code	1 byte
2	NAM_LOCK Indicator	1 byte
3 – 6	RAND OTAPA	0 or 4 bytes

* The RAND_OTAPA (octets 3-6) is returned if and only if the Result Code is '00', the NAM_LOCK_STATE is enabled (='1') and Start/Stop field was set to '1' (Start) in the OTAPA REQUEST command.

The NAM_LOCK Indicator parameter as defined in Section 3.5.1.11 of [7] shall be coded as follows:

Octet 2



Details of the response are in [7], sections 3.3.1.11 "OTAPA Request Message Processing" and 3.5.1.11, "OTAPA Response Message".

4.5.10 PUZL CONFIGURATION REQUEST

COMMAND	CLASS	INS	P1	P2	Lc	Le
PUZL CONFIGURATION REQUEST	'A0'	'F4'	'00'	'00'	*	*

Command parameters/data:

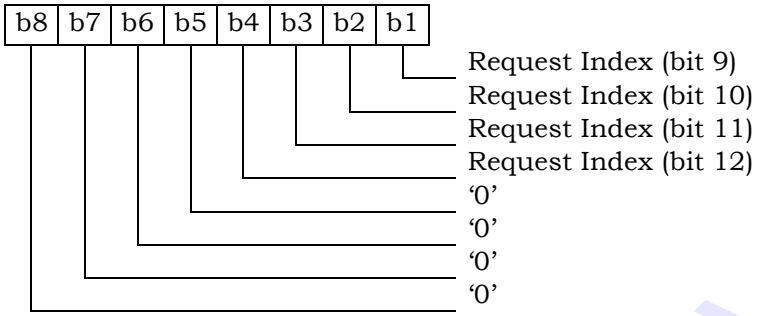
Octet(s)	Description	Length
1	Block ID ('0000 0000')	1 byte

Note: If Block ID = '0000 0001' (PUZL Priorities Parameter Block), then octets 2 through 4 are used as inputs for this command.

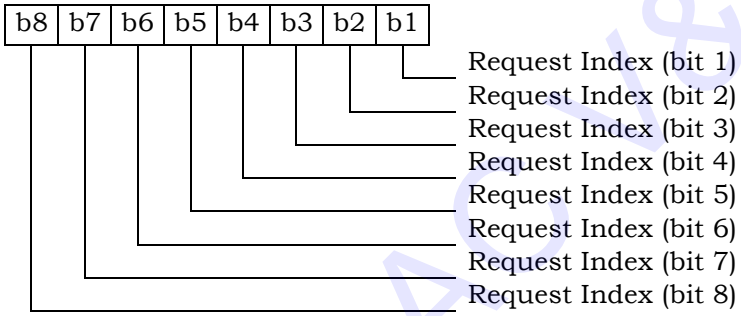
Octet(s)	Description	Length
1	Block ID (‘0000 0001’)	1 byte
2 – 3	Request Index	2 bytes
4	Request Max Entries	1 byte

The Request Index parameter as defined in [7] shall be coded as follows:

Octet 2



Octet 3

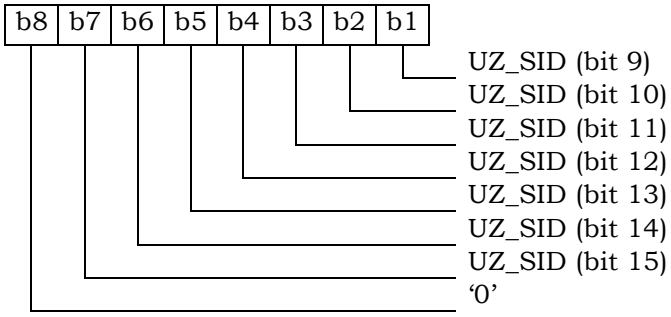


Note: If Block ID = ‘0000 0010’ (User Zone Parameter Block), then octets 2 through 8 are used as inputs for this command.

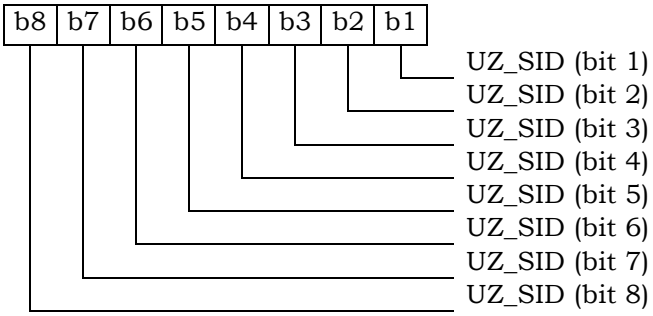
Octet(s)	Description	Length
1	Block ID (‘0000 0010’)	1 byte
2 – 3	UZ_ID	2 bytes
4 – 5	UZ_SID	2 bytes
6 – 7	Request Offset	2 bytes
8	Request Max Size	1 byte

The UZ_SID parameter as defined in [7] shall be coded as follows:

Octet 4

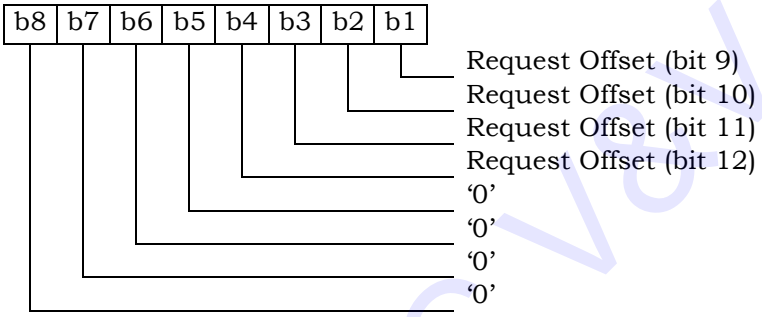


Octet 5

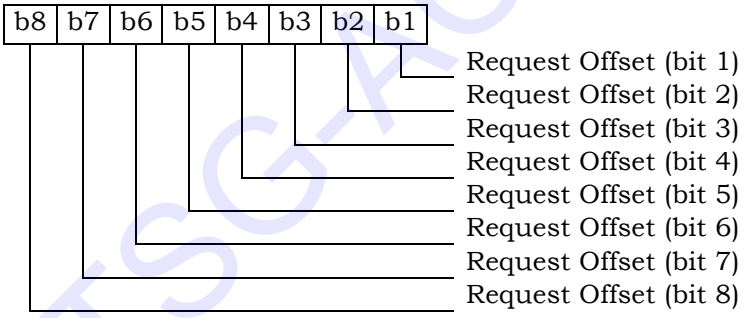


The Request Offset parameter as defined in [7] shall be coded as follows:

Octet 6



Octet 7



Note: If Block ID = ‘0000 0011’ (Preferred User Zone List Parameter Block), then octets 2 through 4 are used as inputs for this command.

Octet(s)	Description	Length
1	Block ID (‘0000 0011’)	1 byte
2 – 3	Request Index	2 bytes
4 – 5	Request Offset	2 bytes
6	Request Max Size	1 byte

Details of command parameters are in [7], section 4.5.1.12, “PUZL Configuration Request Message”.

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Result Code	1 byte
3	Block Length	1 byte
4 – Le	Param Data	Le – 3 bytes

* Note: Le=Length of Param Data + 3.

Details of the response are in [7], sections 3.3.1.12 "PUZL Configuration Request Message Processing" and 3.5.1.12, "PUZL Configuration Response Message".

4.5.11 PUZL DOWNLOAD REQUEST

COMMAND	CLASS	INS	P1	P2	Lc	Le
PUZL DOWNLOAD REQUEST	'A0'	'F6'	'00'	'00'	*	'05'

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Block Length	1 byte
3 – Lc	Param Data	Lc – 2 bytes

* Note: Lc=Length of Param Data + 2.

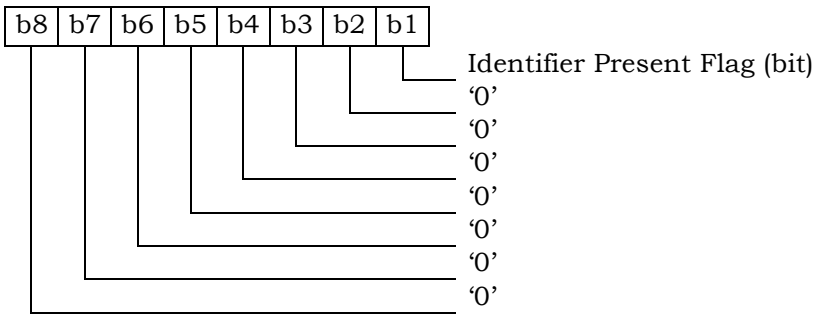
Details of the command parameters are in [7], section 4.5.1.13, "PUZL Download Request Message".

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Result Code	1 byte
3	Identifiers Present Flag	1 byte
4 – 5	UZ_ID	2 bytes
6 – 7	UZ_SID	2 bytes

The Identifiers Present Flag parameter as defined in [7] shall be coded as follows:

Octet 3

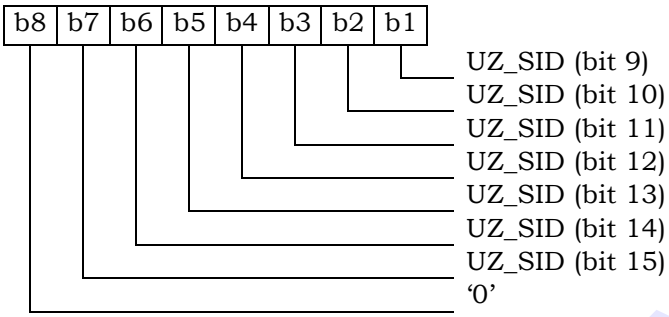


* The octets 4-7 are returned if the Identifiers Present Flag is set to '1'.

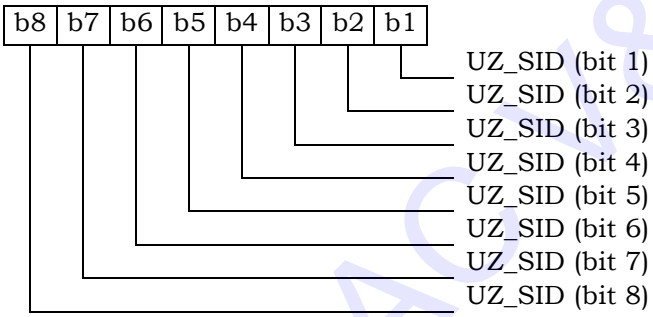
Details of the response are in [7], sections 3.3.1.13, "PUZL Download Request Message Processing" and 3.5.1.13, "PUZL Download Response Message".

The UZ_SID parameter as defined in [7] shall be coded as follows:

Octet 6



Octet 7



4.5.12 3GPD CONFIGURATION REQUEST

COMMAND	CLASS	INS	P1	P2	Lc	Le
3GPD CONFIGURATION REQUEST	'A0'	'FC'	'00'	'00'	01	*

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte

This command requests 3GPD configuration details of a single block of data and forms a subset of the "3GPD Configuration Request Message" as described in [7], section 4.5.1.15.

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Block Length	1 byte
3	Result Code	1 byte
4 – Le	Param Data	Le – 3 bytes

* Note: Le = Length of Param Data + 3.

This response provides 3GPD configuration details of a single block of data and forms a subset of the "3GPD Configuration Response Message" as described in [7], sections 3.3.1.14 and 3.5.1.14. If the Status Words received by the ME are SW1= '69' and SW2= '82' then the RESULT_CODE passed to the network by the ME shall be '33' (Rejected – Secure Mode not active) with a block length of zero. Otherwise the Result Code in the response shall be used.

4.5.13 3GPD DOWNLOAD REQUEST

COMMAND	CLASS	INS	P1	P2	Lc	Le
3GPD DOWNLOAD REQUEST	'A0'	'48'	'00'	'00'	*	'02'

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Block Length	1 byte
3 – Lc	Param Data	Lc – 2 bytes

This command requests the 3GPD download of a single block of data and forms a subset of the "3GPD Download Request Message" as described in [7], section 4.5.1.15.

* Note: Lc = Length of Param Data + 2.

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Result Code	1 byte

This response pertains to a single block of data and forms a subset of the "3GPD Download Response Message" as described in [7], sections 3.3.1.15 and 3.5.1.15. If the Status Words received by the ME are SW1= '69' and SW2= '82' then the RESULT_CODE passed to the network by the ME shall be '33' (Rejected – Secure Mode not active). Otherwise the Result Code in the response shall be used.

4.5.14 SECURE MODE

COMMAND	CLASS	INS	P1[7]	P2	Lc	Le
SECURE MODE	'A0'	'4A'	'00': start '01': stop	'See below'	'08' empty	'01'

[7] Note that the Start/Stop values used here differ from those used in [7].

P1= '00'

Command parameters/data:

Octet(s)	Description	Length
1 – 8	RAND_SM	8 bytes

Details of command parameters are in [7], section 4.5.1.16, "Secure Mode Request Message".

Response parameters/data:

Octet(s)	Description	Length
1	Result Code	1 byte

Details of response parameters are in [7], sections 3.3.1.16, "Secure Mode Request Message Processing" and 3.5.1.16, "Secure Mode Response Message".

P1= '01'

Command parameters/data:

No command parameters are generated.

P2 shall be used for "KEY_IN_USE" parameter as described in [7].

If KEY_IN_USE = '0000', then P2 = 0x00

If KEY_IN_USE = '0001', then P2 = 0x01.

Response parameters/data:

Octet(s)	Description	Length
1	Result Code	1 byte

Details of response parameters are in [7], sections 3.3.1.16, "Secure Mode Request Message Processing" and 3.5.1.16, "Secure Mode Response Message".

4.5.15 FRESH

COMMAND	CLASS	INS	P1	P2	Lc	Le
FRESH	'A0'	'4C'	'00': put '01': get	'00'	'02' empty	Empty '02'

P1= '00'

Command parameters/data:

Octet(s)	Description	Length
1 – 2	Crypto-Sync	2 bytes

Response parameters/data:

No response parameters are generated as a result of command execution. Successful generation will cause SW1 to be set to '90' and SW2 to be set to '00'. Unsuccessful generation will cause SW1 to be set to '98' and SW2 to be set to '04' (Authentication failed [17]).

P1= ‘01’

Command parameters/data:

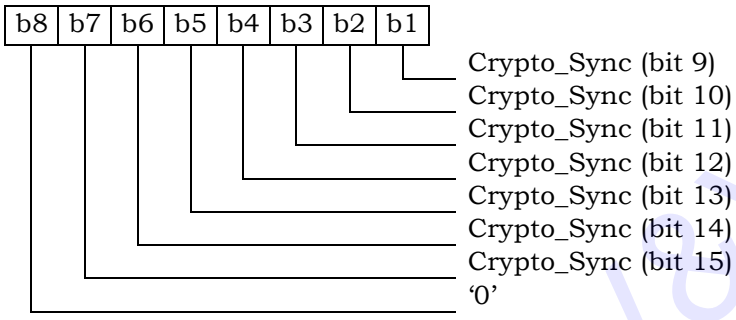
No command parameters are generated.

Response parameters/data:

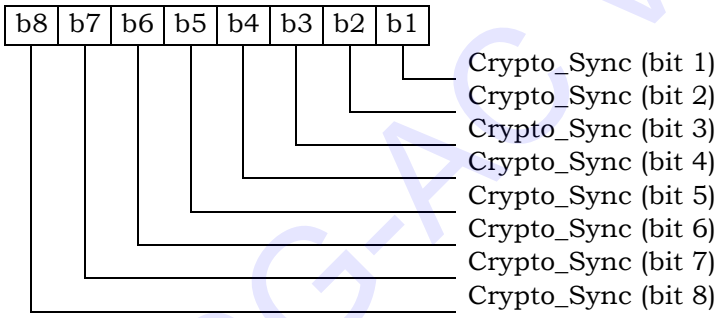
Octet(s)	Description	Length
1 – 2	Crypto-Sync	2 bytes

The Crypto-Sync parameter as defined in [7] shall be coded as follows:

Octet 1



Octet 2



4.5.16 SERVICE KEY GENERATION REQUEST

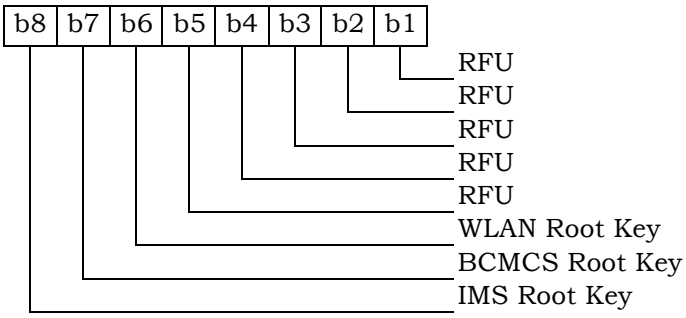
COMMAND	CLASS	INS	P1	P2	Lc	Le
SERVICE KEY GENERATION REQUEST	‘A0’	‘4E’	‘00’	‘00’	‘02’	‘01’

Command parameters/data:

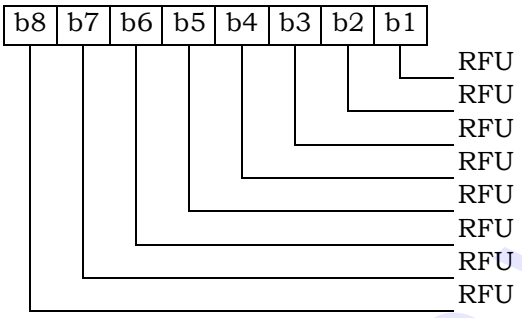
Octet(s)	Description	Length
1-2	KEY_ID	2 bytes

The bitmap of KEY_ID defined in Table 4.5.1.22-1 of [7] shall be coded as follows:

Octet 1:



Octet 2:



Response parameters/data:

Octet(s)	Description	Length
1	Result Code	1 byte

Details of response parameters are in [7].

4.5.17 MMD CONFIGURATION REQUEST

COMMAND	CLASS	INS	P1	P2	Lc	Le
MMD CONFIGURATION REQUEST	'A0'	'C4'	'00'	'00'	'01'	*

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte

This command requests configuration details of a single block of data and forms a subset of the "MMD Configuration Request Message" as described in [7], section 4.5.1.18.

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Block Length	1 byte
3	Result Code	1 byte
4 – Le	Param Data	Le – 3 bytes

* Note: Le=Length of Param Data + 3.

Details of the response are in [7], sections 3.3.1.17, "MMD Configuration Request Message Processing" and 3.5.1.18, "MMD Configuration Response Message".

4.5.18 MMD DOWNLOAD REQUEST

COMMAND	CLASS	INS	P1	P2	Lc	Le
MMD DOWNLOAD REQUEST	'A0'	'C6'	'00'	'00'	*	'02'

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Block Length	1 byte
3 – Lc	Param Data	Lc – 2 bytes

* Note: Lc=Length of Param Data + 2.

Details of the command parameters are in [7], section 4.5.1.19, "MMD Download Request Message".

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Result Code	1 byte

Details of the response are in [7], sections 3.3.1.18, "MMD Download Request Message Processing" and 3.5.1.19, "MMD Download Response Message".

4.5.19 MMS CONFIGURATION REQUEST

COMMAND	CLASS	INS	P1	P2	Lc	Le
MMS CONFIGURATION REQUEST	'A0'	'42'	'00'	'00'	'01'	*

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte

This command requests configuration details of a single block of data and forms a subset of the "MMS Configuration Request Message" as described in [7], section 4.5.1.23.

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Block Length	1 byte
3	Result Code	1 byte
4 – Le	Param Data	Le – 3 bytes

* Note: Le=Length of Param Data + 3.

Details of the response are in [7], sections 3.3.1.22, "MMS Configuration Request Message Processing" and 3.5.1.23, "MMS Configuration Response Message".

4.5.20 MMS DOWNLOAD REQUEST

COMMAND	CLASS	INS	P1	P2	Lc	Le
MMS DOWNLOAD REQUEST	'A0'	'46'	'00'	'00'	*	'02'

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Block Length	1 byte
3 – Lc	Param Data	Lc – 2 bytes

* Note: Lc=Length of Param Data + 2.

Details of the command parameters are in [7], section 4.5.1.24, "MMS Download Request Message".

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Result Code	1 byte

Details of the response are in [7], sections 3.3.1.23, "MMS Download Request Message Processing" and 3.5.1.24, "MMS Download Response Message".

4.5.21 SYSTEM TAG CONFIGURATION REQUEST

COMMAND	CLASS	INS	P1	P2	Lc	Le
SYSTEM TAG CONFIGURATION REQUEST	'A0'	'C8'	'00'	'00'	'04'	*

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2 – 3	Request Offset	2 bytes
4	Request Max Size	1 byte

Note:

If Block ID = '0000 0010' (Group Tag List), '0000 0100' (Specific Tag List), or '0000 0110' (Call Prompt List), then octets 2 through 4 are used as inputs for this command. For other Block IDs, octets 2 through 4 are ignored.

Details of command parameters are in [7], section 4.5.1.20, "System Tag Configuration Request Message".

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Result Code	1 byte
3	Block Length	1 byte
4 – Le	Param Data	Le - 3 bytes

* Note: Le=Length of Param Data + 3.

Details of the response are in [7], sections 3.3.1.19, "System Tag Configuration Request Message Processing" and 3.5.1.20, "System Tag Configuration Response Message".

4.5.22 SYSTEM TAG DOWNLOAD REQUEST

COMMAND	CLASS	INS	P1	P2	Lc	Le
SYSTEM TAG DOWNLOAD REQUEST	'A0'	'CA'	'00'	'00'	*	'05'

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Block Length	1 byte
3 – Lc	Param Data	Lc - 2 bytes

* Note: Lc=Length of Param Data + 2.

Details of the command parameters are in [7], section 4.5.1.21, "System Tag Download Request Message".

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Result Code	1 byte
3 – 4	Segment Offset	2 bytes
5	Segment Size	1 byte

Note: If the BLOCK_ID = '0000 0001' (Group Tag List), '0000 0010' (Specific Tag List), or '0000 0011' (Call Prompt List), then octets 3 through 5 are used. For other Block IDs, octets 3 through 5 are ignored.

Details of the response are in [7], sections 3.3.1.20, “System Tag Download Request Message Processing” and 3.5.1.21, “System Tag Download Response Message”.

4.6 ESN and MEID Management Command

If T=0 protocol is used, APDU is mapped onto TPDU. (See Section 9.1 in [17])

4.6.1 Store ESN_MEID_ME

COMMAND	CLASS	INS	P1	P2	Lc	Le
STORE ESN_MEID_ME	'A0'	'DE'	'XX'	'00'	'08'	'01'

The STORE ESN_MEID_ME command stores to the R-UIM: the ESN_ME (P1 = '00') or the MEID_ME (P1 = '01').

P1 is set to '00' if any of the following condition holds:

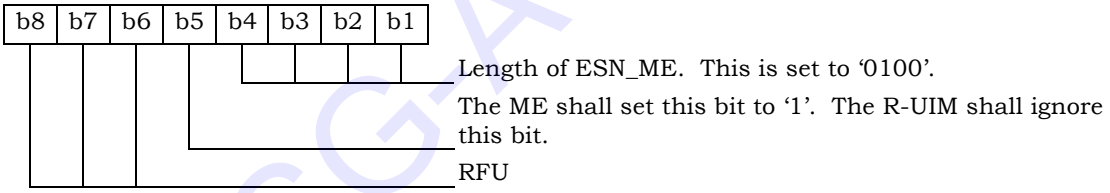
- ME is not assigned with an MEID_ME;
- ME is assigned with an MEID_ME but service n9 is not activated;

Command parameters/data (P1 = '00'):

Octet(s)	Description	Length
1	ESN_ME Length	1 byte
2 – 5	ESN_ME	4 bytes
6 – 8	RESERVED	3 bytes

During the ME and R-UIM initialization process, the ME shall invoke the "STORE ESN_MEID_ME" command to store its ESN_ME in $EF_{ESN\_MEID\_ME}$.

Octet 1:



Octet 2 – 5:

ESN_ME is encoded with the lowest-order byte first to match the coding for $EF_{ESN\_MEID\_ME}$.

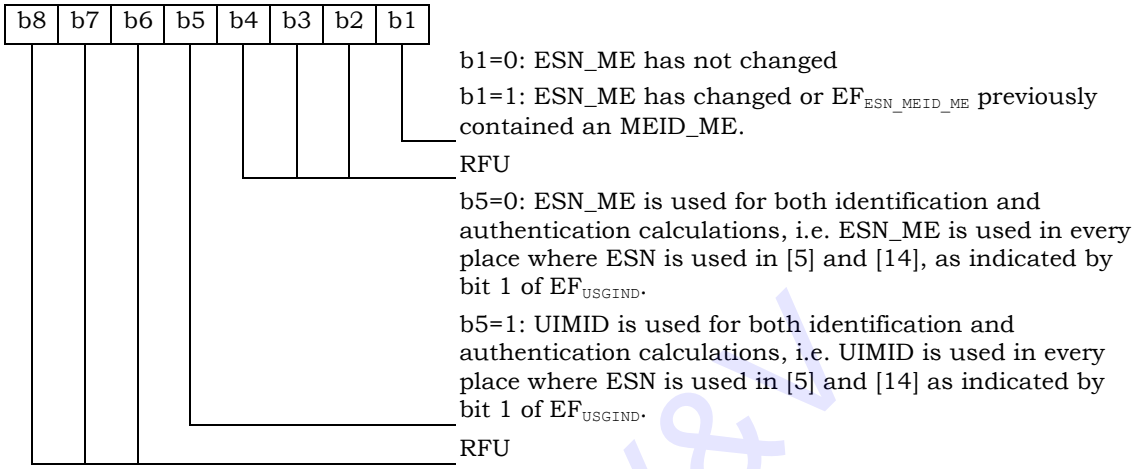
Octet 6 – 8:

The RESERVED field is set to '00 00 00'.

Response parameters/data:

Octet	Description	Length
1	Change Flag, Usage Indicator	1 byte

Octet 1:



P1 is set to '01' if all of the following condition holds:

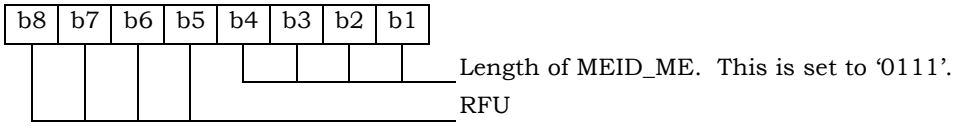
- Service n9 is allocated and activated.
- ME is assigned an MEID_ME.

Command parameters/data (P1 = '01'):

Octet(s)	Description	Length
1	MEID_ME Length	1 byte
2 – 8	MEID_ME	7 bytes

During the ME and R-UIM initialization process, the ME shall invoke the "STORE ESN_MEID_ME" command to store its MEID_ME in $EF_{ESN\_MEID\_ME}$.

Octet 1:



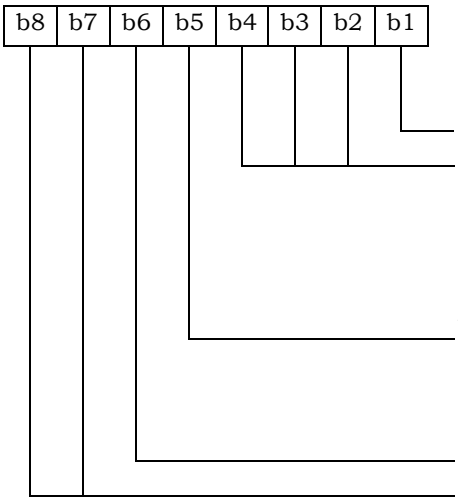
Octet 2 – 8:

MEID_ME is encoded with the lowest-order byte first to match the coding for $EF_{ESN\_MEID\_ME}$.

Response parameters/data:

Octet	Description	Length
1	Change Flag, Usage Indicator	1 byte

Octet 1:



b1=0: MEID_ME has not changed

b1=1: MEID_ME has changed or $EF_{ESN\_MEID\_ME}$ previously contained an ESN_ME.

RFU

b5=0: pESN is used for both identification and authentication calculations, i.e. pESN is used in every place where ESN is used in [5] and [14], as indicated by bit 1 of EF_{USGIND}.

b5=1: UIMID is used for both identification and authentication calculations, i.e. UIMID is used in every place where ESN is used in [5] and [14] , as indicated by bit 1 of EF_{USGIND}.

b6=0: MEID_ME is used for MS identification, as indicated by bit 2 of EF_{USGIND}.

b6=1: SF_EUIMID is used for MS identification, as indicated by bit 2 of EF_{USGIND}.

RFU

4.7 Description of Packet Data Security-Related Functions

This section describes the interface between the ME and R-UIM when the R-UIM performs service authentication and access authentication functions for 3G packet data service. Currently [23] defines Simple IP and Mobile IP as the two access methods for service authentication. Simple IP refers to a service in which an access provider network assigns an IP address and supplies an IP routing address to an MS. When using Simple IP, the network may request either Point-to-Point Challenge Handshake Authentication Protocol (PPP CHAP) or Point-to-Point Password Authentication Protocol (PPP PAP) to authenticate the user. Mobile IP refers to a service where the network provides the user with IP routing service to a public IP network and/or secure IP routing service to private networks. When using Mobile IP, the network authenticates the user by Mobile IP mobile-home authentication and Mobile IP challenge/response authentication.

[29] defines access authentication used for HRPD. Access authentication is a procedure in which the Access Terminal (AT) is authenticated by the AN-AAA (Access Network Authentication, Authorization and Accounting entity).

The following figure shows the authentication model for both the packet data service and voice services.

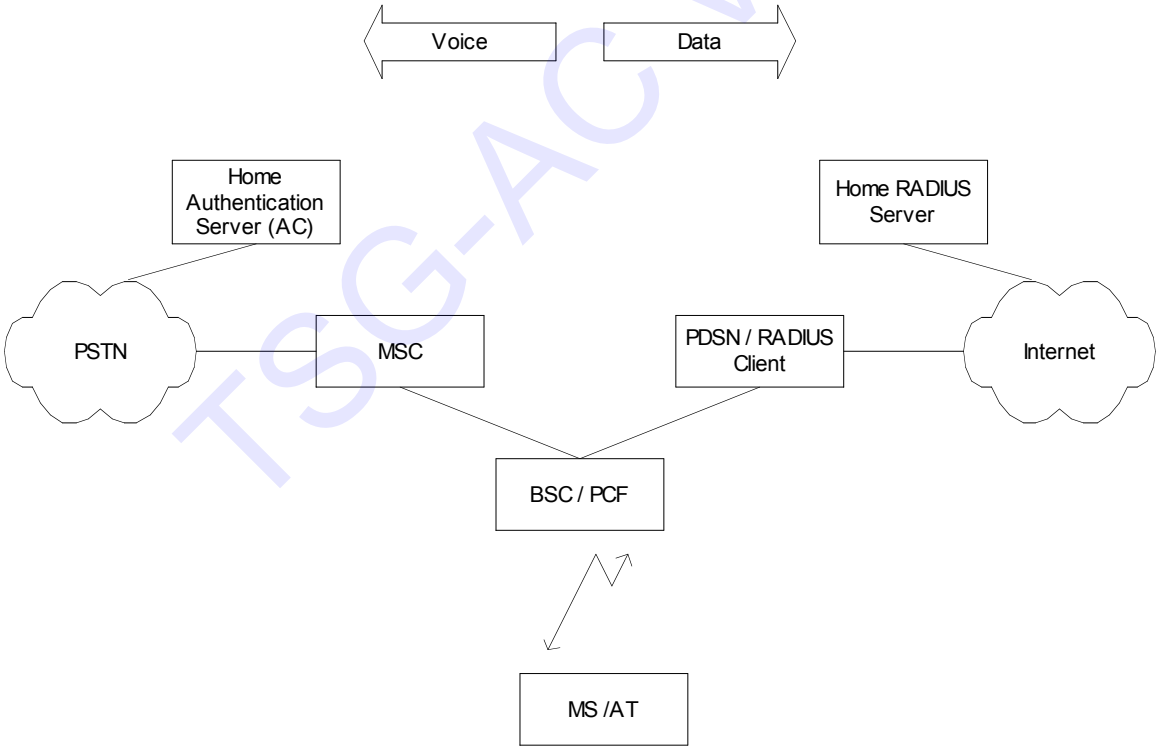


Figure 7. Authentication Models

4.7.1 Managing Shared Secrets

The R-UIM stores and manages the Shared Secrets (SS) used in Simple IP and Mobile IP operation for packet data authentication calculations. The network can update the Shared Secrets on the R-UIM using secure mode OTASP/OTAPA messages.

4.7.2 Performing Simple IP Authentication

As shown in the Figure below (COMPUTE IP AUTHENTICATION (CHAP)), to start the Simple IP authentication process, the network (PDSN) sends a CHAP-Challenge to the mobile station along with the same CHAP-ID sent by the mobile station in the access request. The mobile equipment (ME) will forward this information to the R-UIM with the NAI-Entry-Index used in the access request using the COMPUTE IP AUTHENTICATION (CHAP) command. This NAI-Entry-Index determines the SS to be used in the calculation of the CHAP-Response. The R-UIM computes the CHAP-Response and passes it to the ME to be subsequently forwarded to the network. If the CHAP-Response sent by the MS matches the network's calculated CHAP-Response, the network will send back an Access-Accept granting service.

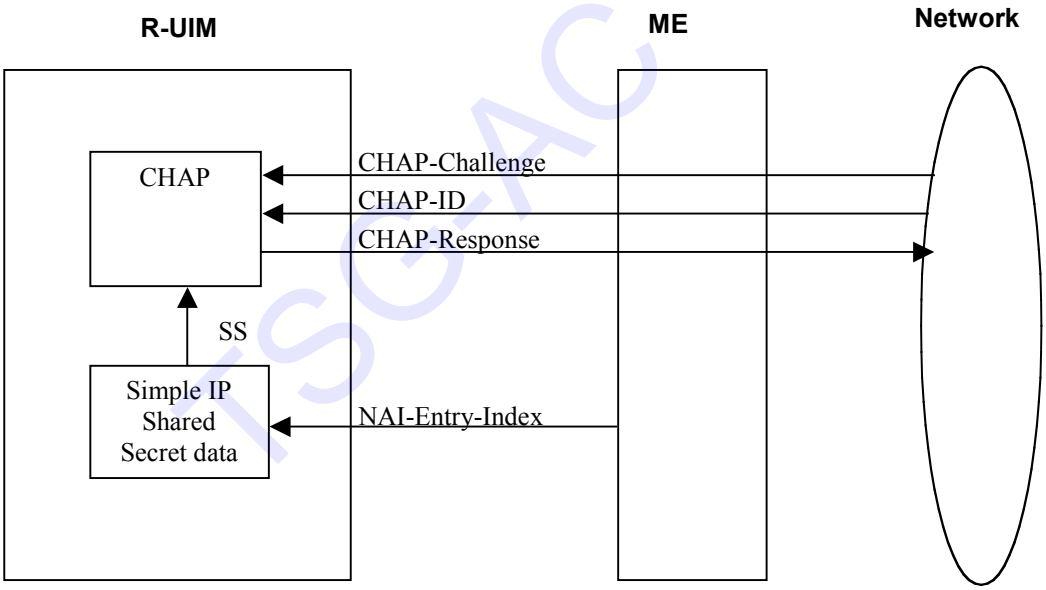


Figure 8. COMPUTE IP AUTHENTICATION (CHAP)

4.7.3 Performing Mobile IP Authentication

For a mobile station that uses Mobile IP, the PDSN shall begin transmission of an operator configurable number of Agent Advertisements immediately following establishment of PPP or upon reception of an Agent Solicitation message from the mobile station. Mobile IP authentication takes place after the ME receives the agent advertisement message with a challenge from the host.

An overview of the Computation of MN-AAA Authenticator is given in the following figure.

To authenticate, the mobile station shall start by sending a Mobile IP registration request message (MIP-RRQ) to the network as defined in [23]. This message shall include various extensions that allow authentication data to be carried from the mobile station to the PDSN. The PDSN then sends the authentication data to a RADIUS server by use of an Access Request message. Once the Authentication is successful, the RADIUS server responds either with an Access Accept message to grant service or with an Access Reject to refuse service.

The MIP_RRQ message shall include the following extensions as specified in [23] in the order given:

1. MN-NAI Extension [25]
2. MN-HA Authentication Extension [24]
3. MN-FA Challenge Extension [27]
4. MN-AAA Extension [27]

The mobile station shall use a static Home Agent (HA) address.

To calculate the MN-HA Authentication extension, the ME sends the COMPUTE IP AUTHENTICATION (MN-HA Authenticator) to the R-UIM with the following information:

- the NAI-Entry-Index to indicate the NAI used in the request,
- the protected fields of the MIP-RRQ (Registration Message) (refer to [24]).

The protected fields are:

- the UDP payload,
- all prior Extensions in their entirety and
- the Type, Length and SPI of this Extension.

The R-UIM returns the MN-HA-Authenticator by hashing the MN-HA Shared Secret indicated by the associated NAI with the protected fields in the registration message.

Since the RADIUS protocol defined in [26] cannot carry attributes greater than 253 in size, the preceding Mobile IP data, type, subtype (if present), length and SPI are hashed before the MN-AAA Authenticator can be generated. This is achieved by using the COMPUTE IP AUTHENTICATION (MIP-RRQ Hash). In this command the ME sends the preceding MIP-RRQ data to the R-UIM and the R-UIM calculates the Hash of this data. The Hash is not returned to the ME.

Subsequently the CHALLENGE from the network and the NAI-Entry-Index identifying the secret the mobile station shares with the home RADIUS server shall be sent to the R-UIM in the COMPUTE IP AUTHENTICATION (MN-AAA Authenticator) command.

The R-UIM computes the MN-AAA Authenticator according to [27], and returns to the ME, to be sent in the MIP-RRQ message to the network.

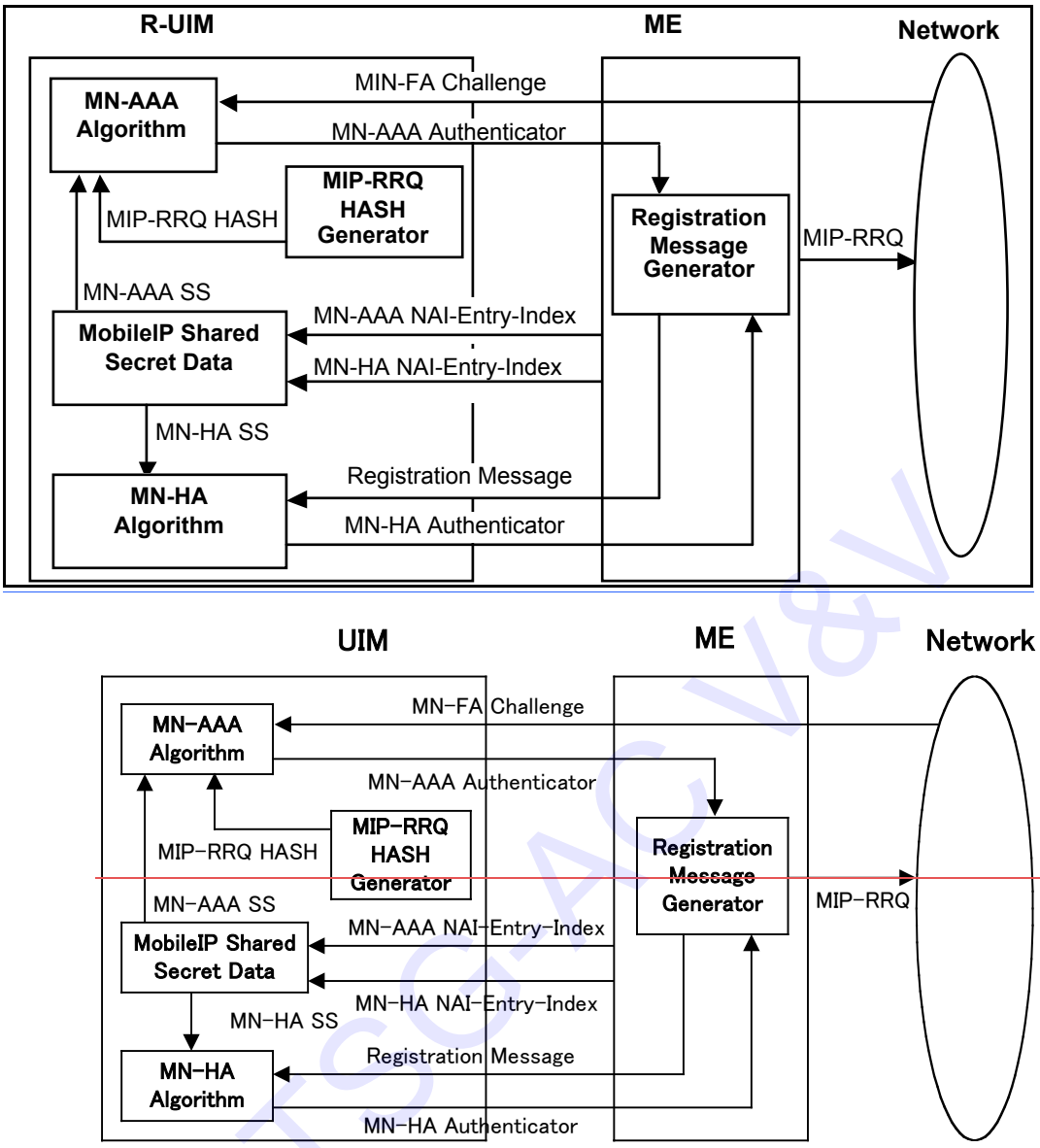


Figure 9. Computation of MN-AAA Authenticator

4.7.4 HRPD Access Authentication

For access authentication, the AT and the network AN initiate Point-to-Point Protocol (PPP) and Link Control Protocol (LCP) negotiations. If the access authentication feature is used, the AN always proposes CHAP as a PPP option in an initial LCP Configure-Request during the PPP establishment. The AN generates a random challenge and sends it to the AT in a CHAP-Challenge message.

The mobile equipment (ME) will forward this information to the R-UIM using the COMPUTE IP AUTHENTICATION (HRPD Access Authentication) command. The R-UIM computes the CHAP-Response and passes it to the ME to be subsequently forwarded to the network. If

the CHAP-Response sent by the AT matches the network's calculated CHAP-Response, the AN will return an indication of CHAP access authentication success to the AT.

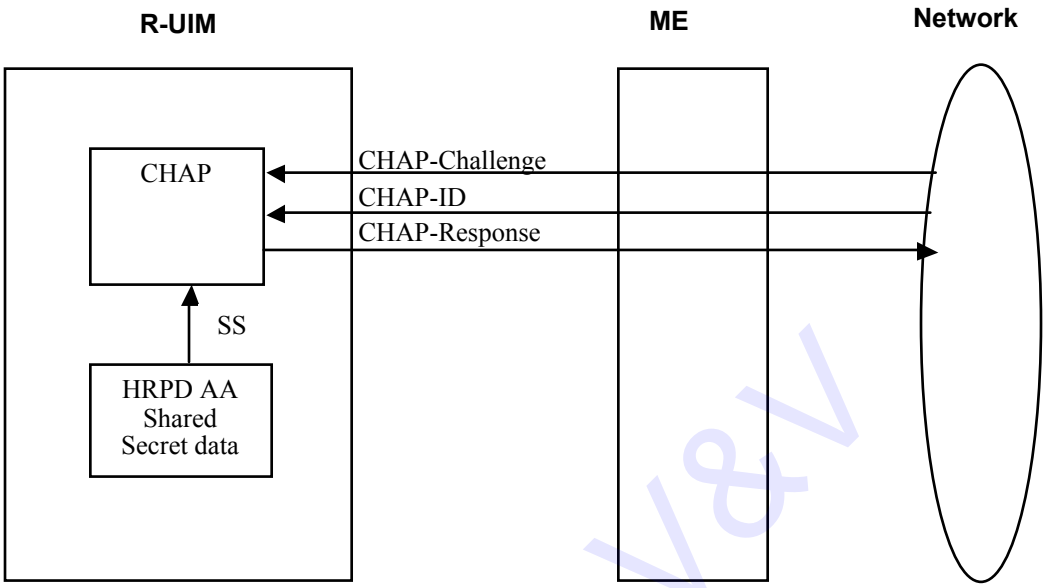


Figure 10. HRPD Access Authentication Command

4.8 Description of Packet Data Security-Related Commands

4.8.1 COMPUTE IP AUTHENTICATION

This command computes responses and authenticators for use in Simple IP, Mobile IP and HRPD Access Authentication.

COMMAND	CLASS	INS	P1	P2	Lc	Le
COMPUTE IP AUTHENTICATION	'80'	'80'	P1	P2	Lc	Le

P1 parameter defines the COMPUTE IP AUTHENTICATION command type:

P1	CLASS
00	CHAP
01	MN-HA Authenticator
02	MIP-RRQ Hash
03	MN-AAA Authenticator
04	HRPD Access Authentication

The MS must perform the COMPUTE IP AUTHENTICATION (MN-HA Authenticator), COMPUTE IP AUTHENTICATION (MIP-RRQ Hash) and COMPUTE IP AUTHENTICATION (MN-AAA Authenticator) commands in sequence. If either COMPUTE IP AUTHENTICATION (MIP-RRQ Hash) or COMPUTE IP AUTHENTICATION (MN-AAA Authenticator) are received

out of sequence, the R-UIM shall return SW1='98' and SW2='34' (Error, out of sequence). In this case, the ME shall abandon the sequence of commands and shall re-start the sequence of commands starting with COMPUTE IP AUTHENTICATION (MN-HA Authenticator) if the ME performs the sequence of commands again. However, the MS can execute the COMPUTE IP AUTHENTICATION (MN-HA Authenticator) command any number of times before the COMPUTE IP AUTHENTICATION (MIP-RRQ Hash) and COMPUTE IP AUTHENTICATION (MN-AAA authenticator) commands.

4.8.1.1 CHAP

This COMPUTE IP AUTHENTICATION command generates the CHAP response.

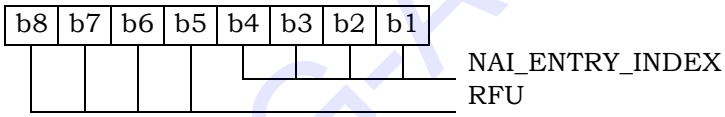
COMMAND	**CLASS**	**INS**	**P1**	**P2**	**Lc**	**Le**
COMPUTE IP AUTHENTICATION	'80'	'80'	'00'	'00'	*	'10'

Command parameters/data:

Octet(s)	**Description**	**Length**
1	CHAP_ID	1 byte
2	NAI-Entry-Index	1 byte
3 – X	CHAP-Challenge	Lc - 2 byte

CHAP-ID: CHAP Identifier as specified in [23] and [26].

NAI-Entry-Index: The Simple IP NAI-Entry-Index indicates the Shared Secret to use from the Simple IP CHAP SS Parameters block. The field carries the 4-bit NAI_ENTRY_INDEX defined in Sec. 3.5.8.10 of [7].



CHAP-Challenge: Challenge received from the network used in computing the CHAP-Response. The length of the CHAP-Challenge depends upon the method used to generate the octets, and is independent of the hash algorithm used.

*Lc = Length of CHAP-Challenge + 2.

Response parameters/data:

Octet(s)	**Description**	**Length**
1 – 16	CHAP-Response	16 bytes

The R-UIM calculates the CHAP-Response as follows:

CHAP-Response = Algo (CHAP-ID || CHAP-SS || CHAP-Challenge)

CHAP-SS: Simple IP CHAP Shared Secret associated with the given NAI-Entry-Index

Algo: The operator shall choose the function for one-way hashing. MD5 is defined as the hashing function, but the operator may choose another hashing function.

4.8.1.2 MN-HA Authenticator

This COMPUTE IP AUTHENTICATION command computes the MN-HA Authenticator. If the maximum length of the Registration-Message exceeds 254 bytes, this command shall chain successive blocks of registration data with a maximum size of 254 bytes each. Valid block sequences are a) Single block or b) First Block, zero or more Next Blocks and a Last Block. If a block used within the command is received out of sequence the card shall return SW1='98' and SW2='34' (Error, out of sequence) and the command shall be considered cancelled by the R-UIM and the ME. In this case, the ME shall abandon the sequence of commands and shall re-start the sequence of commands starting with COMPUTE IP AUTHENTICATION (MN-HA Authenticator) command if the ME performs the sequence of commands again.

COMMAND	**CLASS**	**INS**	**P1**	**P2**	**Lc**	**Le**
COMPUTE IP AUTHENTICATION	'80'	'80'	'01'	*	*	*

P2 contains chaining information as follows:

P2	**Block**
'00'	First Block
'01'	Next Block
'02'	Single Block
'03'	Last Block

*Le : Absent for P2 = '00' or '01'

16 bytes for P2 = '02' or '03'

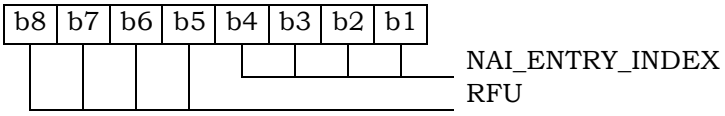
The command data depends on the value of P2:

P2 = '00' or '02':

Command parameters/data:

Octet(s)	**Description**	**Length**
1	NAI-Entry-Index	1 byte
2 – X	Registration-Data	Lc - 1 bytes

NAI-Entry-Index: The Mobile IP NAI-Entry-Index field indicates the MN-HA Shared Secret to be used from the Mobile IP SS Parameters block. The field carries the 4-bit NAI_ENTRY_INDEX defined in Sec. 3.5.8.11 of [7].



Registration-Data: Protected fields from the registration message pursuant to [24]. The protected fields contain: the UDP payload, all prior Extensions in their entirety and the Type, Length and SPI of this Extension (See Section 4.7.3). Maximum length of the Registration-Data is 254 octets per block.

P2 = ‘01’ or ‘03’:

Command parameters/data:

Octet(s)	Description	Length
1 – X	Registration-Data	Lc bytes

Registration-Data: See above under P2=‘00’ or ‘02’.

The response depends on the chaining information P2:

P2 = ‘00’ or ‘01’

Response: NONE

P2 = ‘02’ or ‘03’

Response parameters/data:

Octet(s)	Description	Length
1 – 16	MN-HA Authenticator	16 bytes

The R-UIM calculates the MN-HA Authenticator response as follows:

MN-HA Authenticator = Algo (MN-HA SS || Registration-Message|| MN-HA SS)

MN-HA SS: MN-HA Shared Secret associated with the given NAI-Entry-Index.

Registration-Message: The complete Registration-Message containing the Registration-Data blocks in the consecutive command messages.

Algo: The operator shall choose the function for one-way hashing. MD5 is defined as the hashing function, but the operator may choose another hashing function.

4.8.1.3 MIP-RRQ Hash

This COMPUTE IP AUTHENTICATION command calculates the MIP-RRQ Hash. As the preceding MIP-RRQ data can exceed 247 bytes, it shall be sent to the R-UIM in one or several successive blocks, depending on its actual length. Valid block sequences are a) Single block or b) First Block, zero or more Next Blocks and a Last Block. If a command block is received out of sequence the card shall return SW1=‘98’ and SW2=‘34’ (Error, out of sequence) and the command shall be considered cancelled by the R-UIM and the ME. In this case, the ME shall abandon the sequence of commands and shall re-start the sequence of commands starting with MN-HA Authenticator if the ME performs the sequence of commands again.

COMMAND	CLASS	INS	P1	P2	Lc	Le
COMPUTE IP AUTHENTICATION	‘80’	‘80’	‘02’	*	*	absent

P2 contains chaining information as follows:

P2	Block
'00'	First Block
'01'	Next Block
'02'	Single Block
'03'	Last Block

The command data depends on the value of P2:

P2 = '00' or '01':

Command parameters/data:

Octet(s)	Description	Length
1 – X	Preceding MIP-RRQ Data	Lc bytes

P2 = '02' or '03':

Command parameters/data:

Octet(s)	Description	Length
1 – X	Preceding MIP-RRQ Data	Lc - 8 bytes
X+1 – X+8	MN-AAA Extension Header	8 bytes

Preceding MIP-RRQ Data: The mobile IP registration request preceding the MN-AAA EXTENSION. Maximum length of the Preceding MIP-RRQ Data is 255 for the first and next blocks and 247 octets for the last or single blocks.

MN-AAA Extension Header: Type, Length and SPI fields of the MN-AAA EXTENSION.

Response parameters/data:

NONE

The R-UIM will calculate the MIP-RRQ Hash as follows:

MIP-RRQ Hash: Algo (PRECEDING-MIP-RRQ || MN-AAA Extension Header)

PRECEDING-MIP-RRQ: The complete preceding mobile IP registration request, containing the Preceding MIP-RRQ Data from the consecutive MIP-RRQ Hash options.

Algo: The operator shall choose the function for one-way hashing. MD5 is defined as the hashing function, but the operator may choose another hashing function.

4.8.1.4 MN-AAA Authenticator

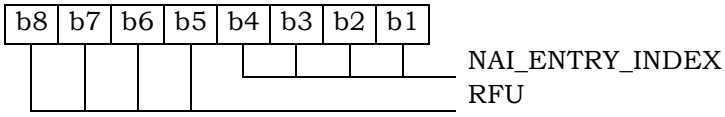
This COMPUTE IP AUTHENTICATION command computes the MN-AAA Authenticator.

COMMAND	CLASS	INS	P1	P2	Lc	Le
COMPUTE IP AUTHENTICATION	‘80’	‘80’	‘03’	‘00’	*	‘10’

Command parameters/data:

Octet(s)	Description	Length
1	NAI-Entry-Index	1 byte
2 – X	Challenge	Lc-1 bytes

NAI-Entry-Index: The Mobile IP NAI-Entry-Index field indicates the MN-AAA Shared Secret to be used from the Mobile IP SS Parameters block. The field carries the 4-bit NAI_ENTRY_INDEX defined in Sec. 3.5.8.11 of [7].



Challenge: Challenge in the MN-FA Challenge Extension. See [27]. If the ME receives a challenge greater than 237 bytes, it will send the highest-order byte and least significant 237 bytes to the R-UIM. If the challenge has fewer than 238 bytes, this R-UIM shall include the high-order byte in the computation twice, but ensures that the challenge is used exactly as is. Additional padding is never used to increase the length of the challenge.

*Lc = Length of Challenge + 1 bytes

Response parameters/data:

Octet(s)	Description	Length
1 – 16	MN-AAA Authenticator	16 bytes

The R-UIM will calculate the response as follows:

MN-AAA Authenticator = Algo (Highest Order byte from Challenge || MN-AAA SS || MIP-RRQ Hash || Least Significant bytes of Challenge up to 237 bytes)

MN-AAA SS: MN-AAA Shared Secret associated with the given NAI-Entry-Index.

Algo: The operator shall choose the function for one-way hashing. MD5 is defined as the hashing function, but the operator may choose another hashing function.

4.8.1.5 HRPD Access Authentication

This COMPUTE IP AUTHENTICATION command generates the CHAP response used for HRPD access authentication.

COMMAND	CLASS	INS	P1	P2	Lc	Le
COMPUTE IP AUTHENTICATION	‘80’	‘80’	‘04’	‘00’	*	‘10’

Command parameters/data:

Octet(s)	Description	Length
1	CHAP_ID	1 byte
2 -X	CHAP-Challenge	Lc - 1 byte

CHAP-ID: CHAP Identifier as specified in [23] and [26].

CHAP-Challenge: Challenge received from the network used in computing the CHAP-Response. The length of the CHAP-Challenge depends upon the method used to generate the octets, and is independent of the hash algorithm used.

*Lc = Length of CHAP-Challenge + 1.

Response parameters/data:

Octet(s)	Description	Length
1 – 16	CHAP-Response	16 bytes

The R-UIM calculates the CHAP-Response as follows:

CHAP-Response = Algo (CHAP-ID || CHAP-SS || CHAP-Challenge)

CHAP-SS: HRPD Access Authentication Shared Secret

Algo: The operator shall choose the function for one-way hashing. MD5 is defined as the hashing function, but the operator may choose another hashing function.

4.9 Descriptions of BCMCS Commands

For complete details, refer to [36] and [58].

The following commands are used for BCMCS key management. The R-UIM shall implement these commands whenever the BCMCS service is allocated in the CDMA Service Table. This assumes that a BCMCS Root key is securely stored in the R-UIM.

COMMAND	CLASS	INS	P1	P2	Lc	Le
BCMCS	‘A0’	‘58’	P1	P2	Lc	Le

P1 parameter defines the BCMCS command type:

P1	CLASS
‘00’	Retrieve SK
‘01’	Update BAK
‘02’	Delete BAK
‘03’	Retrieve SRTP SK
‘04’	Generate Authorization Signature
‘05’	BCMCS Authentication

4.9.1 RETRIEVE SK

4.9.1.1 BCMCS Command description

This command is used by the terminal to ask the R-UIM to calculate the BCMCS Short Term Key (SK) associated with a particular BCMCS Flow Identifier (BCMCS_Flow_ID). For this computation, the R-UIM uses the Broadcast Access Key (BAK) identified by the Broadcast Access Key Identifier (BAK_ID).

Input:

- Service Type = '01' corresponding to "3GPP2 BCMCS"
- BCMCS_Flow_ID
- BAK_ID
- SK_RAND

Output:

- SK

4.9.1.2 Command parameters/data:

Code	Value
CLA	A0
INS	'58'
P1	'00'
P2	'00'
Lc	Length of the subsequent data field
Data	Service Type, BCMCS_Flow_ID, BAK_ID, SK_RAND
Le	'12'

The command data contains:

-A Service Type byte: '01' ("3GPP2 BCMCS")

-Three TLV objects for BCMCS_Flow_ID, BAK_ID, SK_RAND

Note: Coding of Tag Field inside BCMCS TLV Objects is defined in Annex B

Command data:

Byte(s)	Description	Length
1	Service Type = '01' (3GPP2 BCMCS)	1
2-A+1	BCMCS_Flow_ID TLV	A
A+2-A+B+1	BAK_ID TLV	B
A+B+2-A+B+C+1	SK_RAND TLV	C
NOTE: The tags inside TLV objects in the command are specified in Annex B of this document.		

Response parameters/data:

Byte(s)	Description	Length
1 – 18	SK TLV	18
NOTE: The tags inside TLV objects in the response are specified in Annex B of this document.		

4.9.2 Update BAK

4.9.2.1 BCMCS Command description

This command asks the R-UIM to perform a BCMCS BAK update.

Input:

- Service Type = '01' corresponding to "3GPP2 BCMCS"
- BCMCS_Flow_ID
- BAK_ID
- BAK_Expire
- TK_RAND
- Encrypted BAK

Output: None

4.9.2.2 Command parameters/data:

Code	Value
CLA	A0
INS	'58'
P1	'01'
P2	'00'
Lc	Length of the subsequent data field
Data	Service Type, BCMCS_Flow_ID, BAK_ID, BAK_Expire, TK_RAND, Encrypted BAK
Le	Absent

Command data:

Byte(s)	Description	Length
1	Service Type = '01' (3GPP2 BCMCS)	1
2-A+1	BCMCS_Flow_ID TLV	A
A+2-A+B+1	BAK_ID TLV	B
A+B+2 – A+B+C+1	BAK_Expire TLV	C
A+B+C+2 – A+B+C+D+1	TK_RAND TLV	D
A+B+C+D+2 – A+B+C+D+17	Encrypted BAK	16
NOTE: The tags inside TLV objects in the command are specified in Annex B of this document.		

Response Data: None

4.9.3 Delete BAK

4.9.3.1 BCMCS Command description

This command asks the R-UIM to perform a BCMCS BAK deletion in order to free memory. This command should not be used as a means for ending a user's subscription.

Input:

- Service Type = '01' corresponding to "3GPP2 BCMCS"
- BCMCS_Flow_ID
- BAK_ID

Output:

None.

4.9.3.2 Command parameters/data:

Code	Value
CLA	A0
INS	'58'
P1	'02'
P2	'00'
Lc	Length of the subsequent data field
Data	Service Type, BCMCS_Flow_ID, BAK_ID
Le	Absent

Command data:

Byte(s)	Description	Length
1	Service Type = '01' (3GPP2 BCMCS)	1
2-A+1	BCMCS_Flow_ID TLV	A
A+2-A+B+1	BAK_ID TLV	B
NOTE: The tags inside TLV objects in the command is specified in Annex B of this document.		

Response Data: None

The following diagnostics shall be indicated in the command response by the following Status Words:

- SW1= '94', SW2='02' (Invalid BAK ID).
- SW1='94', SW2='04' (Invalid BCMCS Flow ID).

4.9.4 Retrieve SRTP SK

4.9.4.1 BCMCS Command description

This command is used by the terminal to ask the R-UIM to calculate the BCMCS SRTP Short Term Key (SK) associated with a particular BCMCS Flow Identifier (BCMCS_Flow_ID). For this computation, the R-UIM uses the Broadcast Access Key (BAK) identified by the BCMCS_Flow_ID, Broadcast Access Key Identifier (BAK_ID), SK_RAND and Packet Index.

Input:

- Service Type = '01' corresponding to "3GPP2 BCMCS"
- BCMCS_Flow_ID
- BAK_ID
- SK_RAND
- Packet Index

Output:

- SRTP SK

4.9.4.2 Command parameters/data:

Code	Value
CLA	A0
INS	'58'
P1	'03'
P2	'00'
Lc	Length of the subsequent data field
Data	Service Type, BCMCS_Flow_ID, BAK_ID, SK_RAND, Packet Index
Le	'12'

The command data contains:

-Three TLV objects for BAK_ID, SK_RAND and Packet Index

Command data:

Byte(s)	Description	Length
1	Service Type = '01' (3GPP2 BCMCS)	1
2 - A+1	BCMCS_Flow_ID TLV	A
A+2 - A+B+1	BAK_ID TLV	B
A+B+2-A+B+C+1	SK_RAND TLV	C
A+B+C+2-A+B+C+D+1	Packet Index TLV	D
NOTE: The tags inside TLV objects in the command are specified in Annex B of this document.		

Response parameters/data

Byte(s)	Description	Length
1 – 18	SRTP SK TLV	18
NOTE: The tag inside TLV object in the response is specified in Annex B of this document.		

4.9.5 Generate Authorization Signature

4.9.5.1 BCMCS Command description

This command is used by the terminal to ask the R-UIM to calculate the authorization signature associated with a particular BCMCS Flow Identifier (BCMCS_Flow_ID). For this computation, the R-UIM uses the Broadcast Access Key (BAK) identified by the Broadcast Access Key Identifier (BAK_ID) and timestamp.

Input:

- Service Type
- BCMCS_Flow_ID

- BAK_ID
- Timestamp

Output:

- Auth Signature

4.9.5.2 Command parameters/data:

Code	Value
CLA	A0
INS	‘58’
P1	'04'
P2	'00'
Lc	Length of the subsequent data field
Data	Service Type, BCMCS_Flow_ID, BAK_ID, Timestamp
Le	'06'

The command data contains:

-Three TLV objects for BCMCS_Flow_ID, BAK_ID, and Timestamp.

Command data:

Byte(s)	Description	Length
1	Service Type = ‘01’ (3GPP2 BCMCS)	1
2-A+1	BCMCS_Flow_ID TLV	A
A+2-A+B+1	BAK_ID TLV	B
A+B+2-A+B+C+1	Timestamp TLV	C
NOTE: The tags inside TLV objects in the command are specified in Annex B of this document.		

Response parameters/data

Byte(s)	Description	Length
1 – 6	Auth Signature TLV	6
NOTE: The tag inside TLV object in the response is specified in Annex B of this document.		

4.9.6 BCMCS Authentication

4.9.6.1 BCMCS Command description

This command is used by the terminal to ask the R-UIM to calculate the BCMCS digest response for information acquisition. For this computation, the R-UIM uses the BCMCS Root Key.

Input:

- RAND
- Challenge

Output:

- Digest Response

4.9.6.2 Command parameters/data:

Code	Value
CLA	A0
INS	'58'
P1	'05'
P2	'00'
Lc	Length of the subsequent data field
Data	RAND, Challenge
Le	'12'

The command data contains:

-Two TLV objects for RAND, and Challenge.

Command data:

Byte(s)	Description	Length
1	Service Type = '01' (3GPP2 BCMCS)	1
2-A+1	RAND TLV	A
A+2-A+B+1	Challenge TLV	B
NOTE: The tags inside TLV objects in the command are specified in Annex B of this document.		

Response parameters/data

Byte(s)	Description	Length
1 – 18	Digest Response TLV	18
NOTE: The tag inside TLV object in the response is specified in Annex B of this document.		

4.10 Descriptions of Application Authentication Commands

The ME will select the authentication mechanism based on the capability of the R-UIM card and the server, and send an Authenticate Command to the card to generate the response and optionally session keys. Successful authentication calculation will cause SW1 to be set to '90' and SW2 to be set to '00'. Unsuccessful calculation will cause SW1 to be set to '98' and SW2 to be set to '04' (Authentication failed [17]).

For complete details on MMS, refer to [37], [39], [40] and [41]. For complete details on MMD, refer to [45]

4.10.1 Application Authentication

R-UIM generates response and optional 1 or 2 sets of session keys.

COMMAND	CLASS	INS	P1	P2	Lc	Le
APPLICATION AUTHENTICATION	'A0'	'5A'	'00'	'00'	'xx'	'xx'

Command parameters/data:

Octet(s)	Description	Length
1	Authentication Mechanism & Algorithm	1 byte
2	Application ID	1 byte
3-4	Length of Realm (Service or Host Name)	2 bytes
5 to A+4	Realm (Service or Host Name)	A bytes
A +5 to A+6	Length of Server Nonce	2 bytes
A +7 to A + B+6	Server Nonce	B bytes
A+ B+7 to A + B +8	Length of Client Nonce	2 bytes
A + B+9 to A+B+C+8	Client Nonce	C bytes

The coding for authentication mechanism & algorithm is defined according to the following table:

Table 10. Authentication mechanism

Binary Value	Authentication Mechanism
'00000000'	CRAM-MD5
'00000001'	HTTP Digest (MD5)
'00000010'	HTTP Digest (MD5-sess)
'00000011'	HTTP Digest (AKAv1-MD5)
'00000100'	HTTP Digest (AKAv1-MD5-sess)

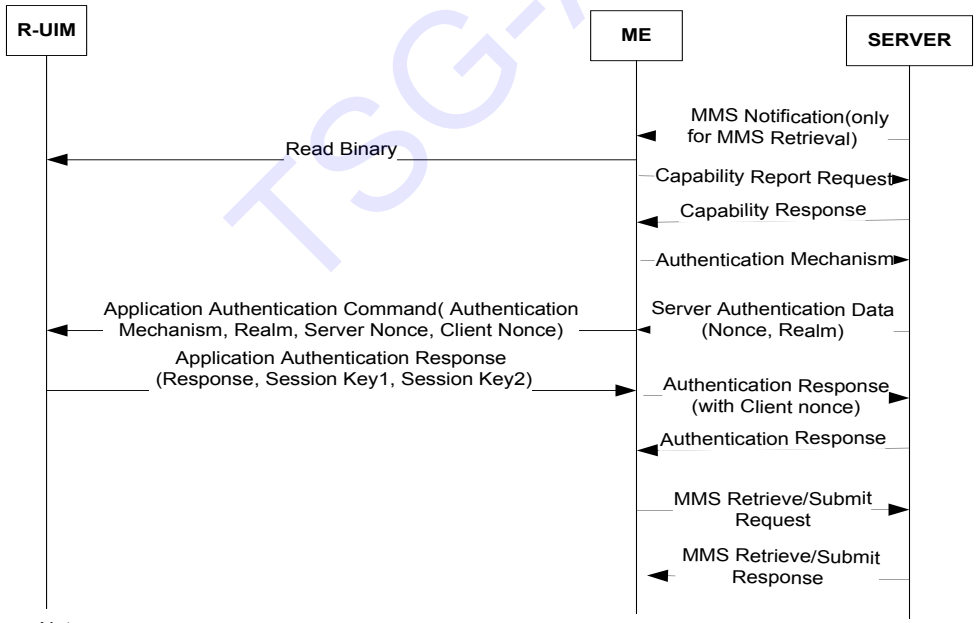
‘00000101’	SASL DIGEST
‘00000110’	SASL OTP
‘00000111’	SASL GSSAPI
‘00001000’-‘11111111’	Reserved

Response parameters/data:

Octet(s)	Description	Length
1	Response Length	1 bytes
2 to X+1	Response	X bytes
X+2 to X+ 3	SessionKey1 Length	2 bytes
X+ 4 to X+ Y+3	SessionKey1	Ybytes
X+ Y+4 to X+ Y+5	SessionKey2 Length	2 bytes
X+ Y+6 to X+ Y+ Z+5	SessionKey2	Z bytes

It is up to different authentication mechanism algorithm to determine if session keys are needed and if so, how many session keys should be returned. For example, SASL Digest returns 2 session keys, HTTP Digest (MD5-session) returns 1 session key and HTTP Digest (MD5) returns no session key. If no session key is to be returned by the R-UIM, the R-UIM shall set the corresponding session key length to 0.

The following is a call flow for MMS message retrieval:



Note:
Capability Report/Response/Authentication Mechanismare all optional; that is, either none of them are used, or all of them are used. The carrier determines to use them or not.

4.11 Description of AKA-related Functions

In order to support AKA, the R-UIM shall support the requirement defined in Section 2.2.2 of [42] and section 2.1.2.3 of [59]. The following AKA-related parameters are stored in the R-UIM.

- Root Key
- Cipher and Integrity Keys (CK, IK)
- SQN_{MS}
- UAK (if supported)

4.11.1 Authentication and key agreement procedure

This section gives an overview of the authentication mechanism and cipher and integrity key generation that are invoked by the network. For complete details, refer to [5], [20], [42] and [59]. The mechanism achieves mutual authentication by the user and the network showing knowledge of a secret root key that is shared between the R-UIM and the Authentication Center. In addition, the R-UIM keeps track of a counter SQN_{MS} to support network authentication. SQN_{MS} denotes the highest sequence number the R-UIM has ever accepted.

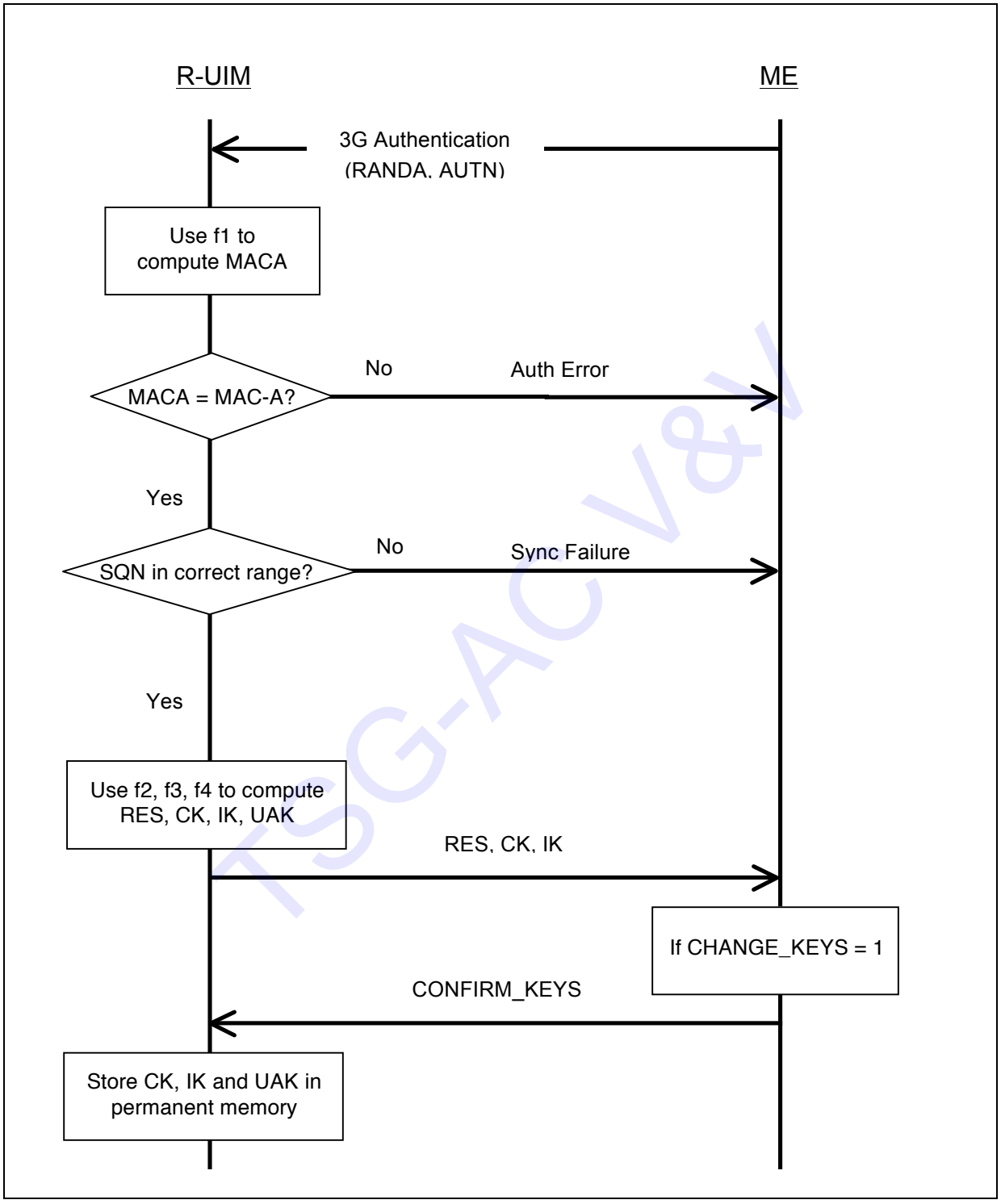
The R-UIM first computes the anonymity key AK = f5 (RANDA) and retrieves the SQN = (SQN⊕AK) ⊕AK

Then the R-UIM computes MACA = f1 (SQN ||RAND||AMF) as defined in [20]. This value is compared with the MAC-A value included in AUTN.

The R-UIM keeps track of a counter SQN_{MS} to support network authentication. SQN_{MS} denotes the highest sequence number the R-UIM has ever accepted. If the R-UIM detects the sequence numbers to be invalid, the R-UIM shall set synchronization failure tag to '00000001' and include AUTS.

Where AUTS = ConSeq (SQN_{MS}) || MACS;

ConSeq(SQN_{MS}) = SQN_{MS}⊕ f5*(RAND) is the concealed value of the counter SQN_{MS} in the R-UIM and MACS = f1*(SQN_{MS} || RAND||AMF);


R-UIM
ME
3G Authentication
(RANDA, AUTN)
Use f1 to
compute MACA
MACA = MAC-A?
No
Auth Error
Yes
SQN in correct range?
No
Sync Failure
Yes
Use f2, f3, f4 to compute
RES, CK, IK, UAK
RES, CK, IK
If CHANGE_KEYS = 1
CONFIRM_KEYS
Store CK, IK and UAK in
permanent memory

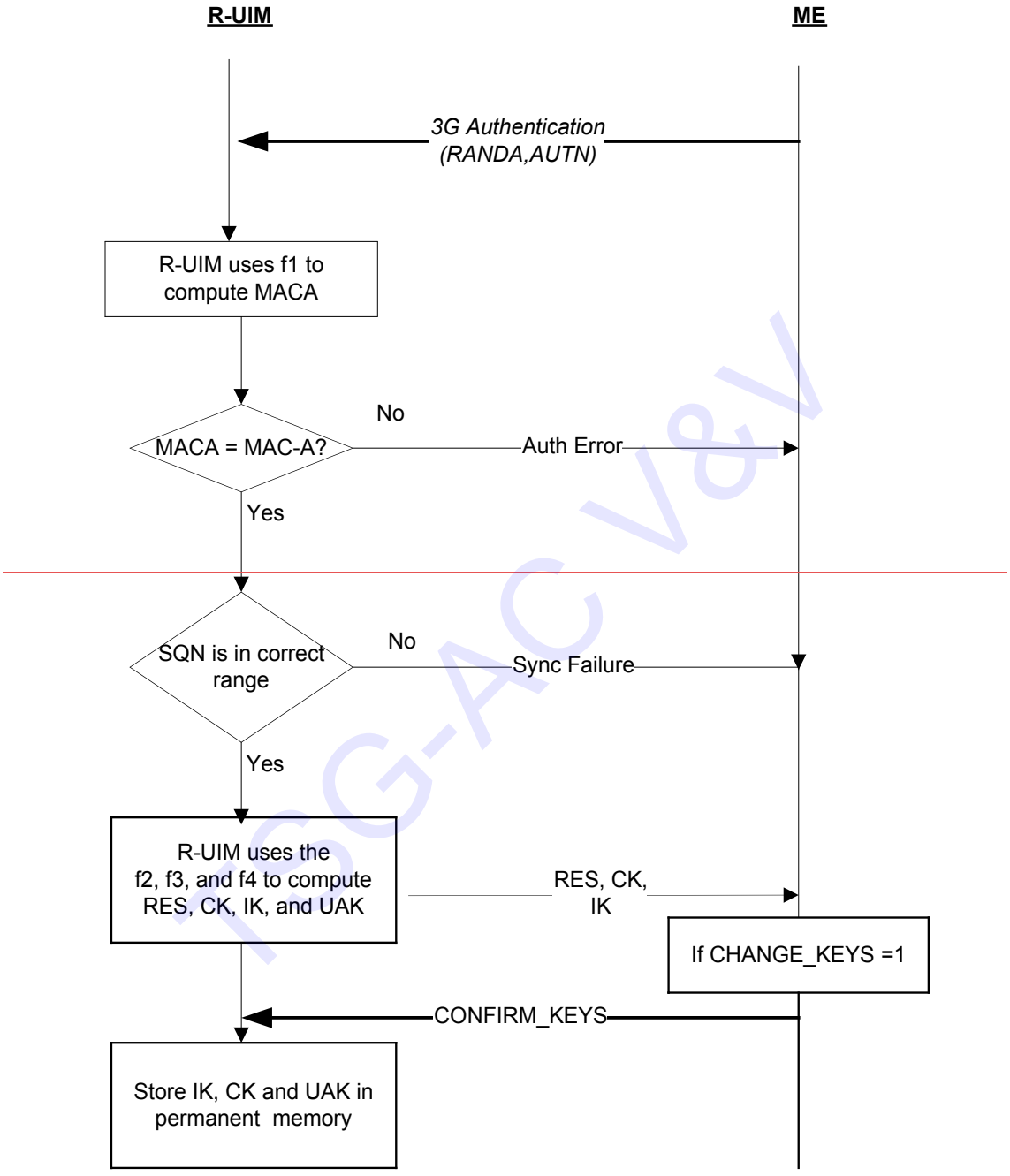


Figure 11. AKA Procedures

4.11.2 Cryptographic Functions

The names and parameters of the cryptographic functions supported by the R-UIM are defined in [42] and [59].

4.11.3 3G Access AKA Command description

The command is used during the procedure for authenticating the R-UIM to its network and vice versa. In addition, a cipher key, an integrity key, and UAK if supported, are calculated. For the execution of the command the R-UIM uses the root key, which is stored in the R-UIM.

4.11.4 UMAC Generation Description

If UAK is supported by the R-UIM, the R-UIM uses UAK to convert MAC-I, into UMAC. If UMAC is successfully generated, the R-UIM responds to the ME by setting the Success Tag to '1' and including the UMAC in the response to the ME. Otherwise, the R-UIM sets the Success Tag to '0' and omits the UMAC.

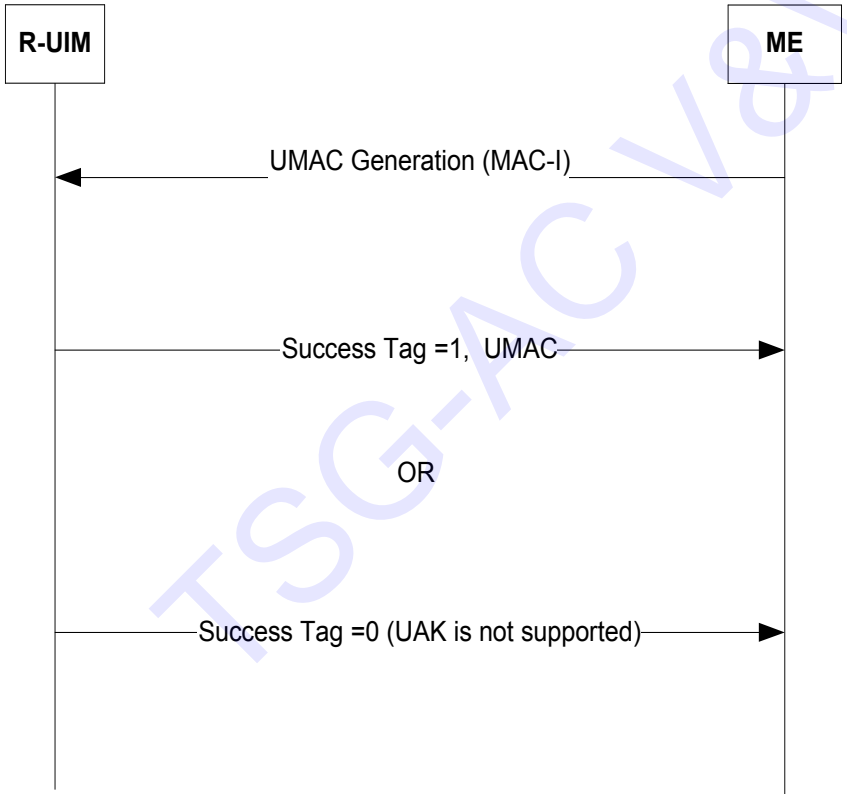


Figure 12. UMAC Generation

4.11.5 Restoration of 3G keys

The CK and IK for 3G circuit-switched authentication are generated and updated by the 3G Access AKA AUTHENTICATE command and sent to the ME in response. After receipt of the CONFIRM_KEYS command the CK and IK are stored in EF_{3GCIK}. The ME shall delete the CK and IK from memory after power-off as well as after removal of the R-UIM. Upon powering on or detecting the insertion of a new R-UIM, if service n30 is allocated and activated, then the ME shall read the EF_{3GCIK} and restore CK and IK.~~The CK and IK are generated during~~

~~AKA, and updated through AKA. The CK and IK are stored in the R-UIM and a copy is stored in the ME. The CK and IK are sent from the R-UIM to the ME upon request from the ME. The ME shall delete the CK and IK from memory after power off as well as after removal of the R-UIM. Upon powering on, the ME shall check the R-UIM revision and service table, if AKA is supported and activated, then the ME shall read the EF_{3GCIK} from the R-UIM and restore them.~~

4.11.6 CONFIRM_KEYS Command description

The command is used during the procedure for 3G authentication. The (IK, CK) pair and the UAK that was calculated by the R-UIM when it received the 3G Access AKA or EAP AKA Authentication command ~~is~~ are now stored in semi-permanent memory. If the previous AUTHENTICATE command was 3G Access AKA then IK and CK shall be stored in EF_{3GCIK}.

4.12 Description of AKA commands

4.12.1 UMAC Generation

This command converts MAC-I into UMAC using UAK.

COMMAND	CLASS	INS	P1	P2	Lc	Le
UMAC GENERATION	'A0'	'5E'	'00'	'00'	'04'	'xx'

Command parameters/data:

Octet(s)	Description	Length
1-4	MAC-I	4 bytes

Response parameters/data:

Octet(s)	Description	Length
1	SUCCESS TAG	1 byte
2 – 5	UMAC	0 or 4 bytes

If the R-UIM generates UMAC successfully, the R-UIM shall set success tag to '00000001', and include the UMAC. If the R-UIM does not support UAK, the R-UIM shall set success tag to '00000000' and omit the UMAC. All the other values are reserved.

4.12.2 CONFIRM_KEYS

COMMAND	CLASS	INS	P1	P2	Lc	Le
CONFIRM_KEYS	'A0'	'5C'	'00'	'00'	empty	empty

Command parameters/data:

The command has no parameters

Response parameters/data:

No response parameters are generated as a result of this command.

5 ADDITIONAL AIR INTERFACE PROCEDURES

5.1 Registration Procedure

5.1.1 R-UIM Removal and Insertion

Upon the removal of an R-UIM from a powered-on ME, the ME shall clear its temporary memory of R-UIM related parameters.

Upon the insertion of an R-UIM into a powered-on ME, the ME shall perform the following:

- Perform ME/R-UIM initialization tasks;
- Update its NAM parameters to those stored on the R-UIM; for any service available and activated in the R-UIM, the parameters available from the R-UIM shall be used.
- Perform the actions defined in 6.6.5.5.1.1 of [14] or 2.6.5.5.1.1 of [5]; and
- Enter the System *Determination Substate* with a power-up indication as described in [5] or [14].

5.1.2 Procedure when ESN Changes with TMSI Assigned

When the ME detects that an R-UIM is inserted, it will use the STORE ESN_MEID_ME command to inform the R-UIM of the ESN or MEID of the ME. If bit 1 of octet 1 of the response parameters/data to the STORE ESN_MEID_ME command is set to '1', $REG\_ENABLED_S$ is equal to YES and there is a TMSI assigned in the R-UIM (the bits of the $TMSI\_CODE_{s\text{-}p}$ field of EF_{TMSI} are not all set to '1'), the ME shall perform the following:

- Store the value USE_TMSI_s in a temporary variable;
- Set USE_TMSI_s to '0';
- Initiate a power up registration regardless of the state of $POWER\_UP\_REG_S$ and $REGISTERED_S$; and
- Restore the value of USE_TMSI_s from the temporary variable.

If the registration fails due to access attempt failure or if the registration is cancelled due to initiation of an origination by the user or detection of a page match (see section 6.6.3.6 of [14] and section 2.6.3.6 of [5]), the ME shall delete the TMSI in the R-UIM by setting all bits of the $TMSI\_CODE_{s\text{-}p}$ field of EF_{TMSI} to '1'.

5.2 NAM Parameters when no R-UIM is inserted into the ME

When no R-UIM is inserted into the ME, the ME shall use the following default set of NAM parameters, from Section 3.1 of [7]:

- $IMSI\_M\_CLASS_p$ shall be set to 0.
- MCC_M_p, $IMSI\_M\_11\_12_p$, and $IMSI\_M\_S_p$ shall be set to coded value of the IMSI_M with the four least-significant digits set to ESN_p, converted directly from binary to decimal, modulo 10000. The other digits shall be set to 0.

- IMSI_M_ADDR_NUM$_p$ shall be set to '000'.
- IMSI_T_CLASS$_p$ shall be set to 0.
- MCC_T$_p$, IMSI_T_11_12$_p$, and IMSI_T_S$_p$ shall be set to the coded value of the IMSI_T with the four least-significant digits set to ESN$_p$, converted directly from binary to decimal, modulo 10000. The other digits shall be set to 0.
- IMSI_T_ADDR_NUM$_p$ shall be set to '000'.
- ACCOLC$_p$ shall be set as specified in 2.3.5 of [5] or 6.3.5 of [14].
- HOME_SID$_p$, if present, shall be set to 0.
- All other indicators of the selected NAM may be set to manufacturer-defined default values. All configuration indicator values shall be set within their valid range (see F.3 of [5] or [14]).

MEs may perform any function allowable by applicable standards, including system accesses when no R-UIM is inserted into the ME.

5.3 IMSI-Related Parameters in the ME when no IMSI is Programmed in the R-UIM

When the IMSI_M_PROGRAMMED bit of the IMSI_M EF is set to '0', the ME shall use the following values associated with IMSI_M in lieu of the values programmed in EF$_{IMSI_M}$:

- IMSI_M_CLASS$_p$ shall be set to 0.
- MCC_M$_p$, IMSI_M_11_12$_p$, and IMSI_M_S$_p$ shall be set to the coded value of the IMSI_M with the four least-significant digits set to ESN$_p$, converted directly from binary to decimal, modulo 10000. The other digits shall be set to 0.
- IMSI_M_ADDR_NUM$_p$ shall be set to '000'.
- ACCOLC$_p$ shall be set as specified in Section 2.3.5 of [5] or Section 6.3.5 of [14].

When the IMSI_T_PROGRAMMED bit of EF$_{IMSI_T}$ is set to '0', the ME shall use the following values for IMSI_T in lieu of the values programmed in EF$_{IMSI_T}$:

- IMSI_T_CLASS$_p$ shall be set to 0.
- MCC_T$_p$, IMSI_T_11_12$_p$, and IMSI_T_S$_p$ shall be set to the coded value of the IMSI_T with the four least-significant digits set to ESN$_p$, converted directly from binary to decimal, modulo 10000. The other digits shall be set to 0.
- IMSI_T_ADDR_NUM$_p$ shall be set to '000'.

5.4 VOID

6 BCMCS PROCEDURES

For complete details, refer to [36] and [58].

6.1 Functionalities of R-UIM and ME

6.1.1 R-UIM

- Generate TK from BCMCS Root Key and TK_RAND, then decrypt BAK using TK
- Compute SK from BAK and SK_RAND and pass SK to ME
- Store BCMCS Root Key, BAK, BCMCS_Flow_ID, BAK_ID and BAK_Expire
- When necessary, generate Auth-Key from BCMCS Root Key and calculate digest response
- When necessary, generate SRTP session Encryption Key using AES
- Generate authorization signature from BAK and timestamp by using EHMAC algorithm (BAK Hash)

6.1.2 ME

- Use SK to decrypt BCMCS content
- Determine whether to issue RetrieveSK command by checking BAK_ID and SK_RAND
- Initiate BAK Request to the network and issue a BCMCS (Update BAK) command
- Can store BCMCS_FLOW_ID, BAK_ID, BAK_EXPIRE, SK and SK_RAND
- Determine the expiration status of BAK and send a BCMCS (Delete BAK) command when necessary

6.2 Key Management

If service n39 is allocated, a secret list of current BAK values (BAK) and secret list of updated BAK values (UpdatedBAK) shall be securely maintained in the R-UIM (not accessible to the ME). When the ME sends a BCMCS (Update BAK) command, the R-UIM shall create a new entry in $EF_{UpBAKPARA}$ and put the decrypted BAK into a record in the secret list of updated BAK values (UpdatedBAK) corresponding to $EF_{UpBAKPARA}$.

When the ME sends a BCMCS (Delete BAK) command, the R-UIM shall search for the given (BCMCS_Flow_ID, BAK_ID) pair in $EF_{BAKPARA}$. If such a record is found, it shall erase (fill up with 'FF') the record corresponding to the BAK in the BAK secret list (BAK). If this search is unsuccessful, the R-UIM shall look for the (BCMCS_Flow_ID, BAK_ID) pair in $EF_{UpBAKPARA}$. If the record is found, the R-UIM shall remove the record in $EF_{UpBAKPARA}$ and corresponding BAK in the Updated BAK secret list (UpdatedBAK) identified by BCMCS_Flow_ID and BAK_ID.

When ME sends a BCMCS (Retrieve SK) command, if BCMCS_Flow_ID and BAK_ID are found in $EF_{BAKPARA}$, R-UIM shall use the corresponding BAK from the secret BAK list (BAK) to generate SK. Otherwise, if the ID pair matches any record in $EF_{UpBAKPARA}$, R-UIM shall copy the 3 parameters (BCMCS_Flow_ID, BAK_ID, BAK_Expire) into the $EF_{BAKPARA}$, copy the corresponding BAK from the Updated BAK secret list (UpdatedBAK) to the BAK secret list (BAK) and use this BAK to generate SK. If none of the precedent procedures apply (BCMCS_Flow_ID and BAK_ID unavailable), the R-UIM shall reply with an appropriate error using status words SW1='6A', SW2='88' (Referenced data or reference data not found)[55].

Annex A (INFORMATIVE): SUGGESTED CONTENTS OF THE EFs AT PRE-PERSONALIZATION

A general outline of the R-UIM files defined in this specification is in Table 11.

1. All values are sized in Bytes unless otherwise noted.
2. Default Values are specified when available and are intended to be guidelines only. In some cases, operators must specify explicit parameter values as no logical default exists. In the case where the parameter values are necessary, valid values and/or ranges are listed.
3. Default and Parameter values are for general quick reference only and not intended to specify details. Refer to the corresponding file for details.
4. Default Values and Parameter Values are specified in Hexadecimal, unless otherwise noted.
5. GSM-specific files are not included.
6. 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.
7. The term 'Ø' used in the column "Access – Read – Update – Invalidate – Rehabilitate" means 'Not applicable'.

Table 11. Summary of R-UIM Files

File Name	*File ID*	*File Type*	*Access – Read – Update – Invalidate – Rehabilitate*	*Size in Bytes*	*Mandatory or Optional*	*Default Values (D) and/or Parameter Values (P) in Bytes*
Authentication – NAM Parameters and Operational Parameters						
A-Key	-	-	Never – Never – Ø – Ø	8	M	Specified by Operator
Root Key	-	-	Never – Never – Ø – Ø	16	M	Specified by Operator
BCMCS Root Key	-	-	Never – Never – Ø – Ø	16	O	Specified by Operator
IMS Root Key	-	-	Never – Never – Ø – Ø	16	O	Specified by Operator
WLAN Root Key	-	-	Never – Never – Ø – Ø	16	O	Specified by Operator
SSD	-	-	Never – Never – Ø – Ø	16	M	-
EF_{COUNT}	3F00/7F25/6F21	CY	CHV1 – CHV1 – ADM – ADM	2	M	D = '00 00'
BAK	-	-	Never – Never – Ø – Ø	16	O	Specified by Operator
UpdatedBAK	-	-	Never – Never – Ø – Ø	16	O	Specified by Operator
SharedSecret	-	-	Never – Never – Ø – Ø	Variable	O	Specified by Operator
UAK	-	-	Never – Never – Ø – Ø	16	O	Specified by Operator
SQN_{MS}	-	-	Never – Never – Ø – Ø	6	O	-

File Name	*File ID*	*File Type*	*Access – Read – Update – Invalidate – Rehabilitate*	*Size in Bytes*	*Mandatory or Optional*	*Default Values (D) and/or Parameter Values (P) in Bytes*
NAM Parameters and Operational Parameters						
EF_{IMSI_M}	3F00/7F25/6F22	TR	CHV1 – ADM – ADM – CHV1	10	M	P = Specified by Operator or D='00…00'
EF_{IMSI_T}	3F00/7F25/6F23	TR	CHV1 – ADM – ADM – CHV1	10	M	P = Specified by Operator or D='00…00'
EF_{TMSI}	3F00/7F25/6F24	TR	CHV1 – CHV1 – ADM – CHV1	16	M	D = '00 00 00 00 00 00 00 00 00 00 FF FF FF FF 00 00 00'
EF_{AH}	3F00/7F25/6F25	TR	CHV1 – CHV1 – ADM – ADM	2	O	P = Specified by Operator or D = '00 00'
EF_{AOP}	3F00/7F25/6F26	TR	CHV1 – CHV1 – ADM – ADM	1	O	-
EF_{ALOC}	3F00/7F25/6F27	TR	CHV1 – CHV1 – ADM – ADM	7	O	-
$EF_{CDMAHOME}$	3F00/7F25/6F28	LF	CHV1 – CHV1 – ADM – ADM	5	M	P = Specified by Operator or D = '00 00 00 00 00'
EF_{ZNREGI}	3F00/7F25/6F29	LF	CHV1 – CHV1 – ADM – ADM	8	M	D = '00 00 00 00 00 00 00 00'
EF_{SNREGI}	3F00/7F25/6F2A	TR	CHV1 – CHV1 – ADM – ADM	7	M	-
$EF_{DISTREGI}$	3F00/7F25/6F2B	TR	CHV1 – CHV1 – ADM – ADM	8	M	D = '00 00 00 00 00 00 00 00'
EF_{ACCOLC}	3F00/7F25/6F2C	TR	CHV1 – ADM – ADM – ADM	1	M	P = '00' to '0F' derived from IMSI_M / IMSI_T
EF_{TERM}	3F00/7F25/6F2D	TR	CHV1 – CHV1 – ADM – ADM	1	M	Specified by Operator P = '00' to '07'
EF_{SSCI}	3F00/7F25/6F2E	TR	CHV1 – CHV1 – ADM – ADM	1	O	Specified by Operator P = '00' to '07'
EF_{ACP}	3F00/7F25/6F2F	TR	CHV1 – CHV1 – ADM – ADM	7	O	Specified by Operator
EF_{PRL}	3F00/7F25/6F30	TR	CHV1 – ADM – ADM – ADM	Variable	M	Specified by Operator
EF_{RUIMID}	3F00/7F25/6F31	TR	ALW – Never – Never-Never	8	M	Specified by R-UIM Manufacturer
EF_{CST}	3F00/7F25/6F32	TR	CHV1 – ADM – ADM – ADM	Variable	M	Specified by Operator
EF_{SPC}	3F00/7F25/6F33	TR	ADM – ADM – ADM – ADM	3	M	D = '00 00 00' or P = '00 00 00' to '99 99 99'
$EF_{OTAPASPC}$	3F00/7F25/6F34	TR	CHV1 – CHV1 – ADM – ADM	1	M	Specified by Operator or D = '00'
$EF_{NAMLOCK}$	3F00/7F25/6F35	TR	CHV1 – CHV1 – ADM – ADM	1	M	Specified by Operator
EF_{OTA}	3F00/7F25/6F36	TR	CHV1 – ADM – ADM – ADM	Variable	M	P = Defined in [7]
EF_{SP}	3F00/7F25/6F37	TR	CHV1 – CHV1 – ADM – ADM	1	M	Specified by Operator
$EF_{ESN\_MEID\_ME}$	3F00/7F25/6F38	TR	ALW – ADM – ADM – ADM	8	M	D ='00…00'

File Name	*File ID*	*File Type*	*Access – Read – Update – Invalidate – Rehabilitate*	*Size in Bytes*	*Mandatory or Optional*	*Default Values (D) and/or Parameter Values (P) in Bytes*
$EF_{Revision}$	3F00/7F25/6F39	TR	ALW – ADM – ADM – ADM	1	M	D = '04'
EF_{RUIM_PL}	3F00/7F25/6F3A	TR	ALW – CHV1 – ADM – ADM	Variable	M	D = 'FF… FF'
EF_{SMS}	3F00/7F25/6F3C	LF	CHV1 – CHV1 – ADM – ADM	Variable	O	D = '00 FF…FF'
EF_{SMSP}	3F00/7F25/6F3D	LF	CHV1 – CHV1 – ADM – ADM	Variable	O	D = 'FF…FF'
EF_{SMSS}	3F00/7F25/6F3E	TR	CHV1 – CHV1 – ADM – ADM	Variable	O	D = 'FF…FF'
EF_{SSFC}	3F00/7F25/6F3F	TR	CHV1 – CHV1 – ADM – ADM	Variable	O	Specified by Operator
EF_{SPN}	3F00/7F25/6F41	TR	ALW – ADM – ADM – ADM	35	O	Specified by Operator
EF_{USGIND}	3F00/7F25/6F42	TR	CHV1 – ADM – ADM – ADM	1	M	Specified by Operator
EF_{AD}	3F00/7F25/6F43	TR	ALW – ADM – ADM – ADM	Variable	M	D = '00…00'
EF_{MDN}	3F00/7F25/6F44	LF	CHV1 – CHV1 – ADM – ADM	11	O	Specified by Operator
EF_{MAXPRL}	3F00/7F25/6F45	TR	CHV1 – ADM – ADM – ADM	2 or 4	M	Specified by Operator
EF_{SPCS}	3F00/7F25/6F46	TR	CHV1 – Never – Never-Never	1	M	P = If EF 6F33 is set to default value then D = '00' otherwise D = '01'
EF_{ECC}	3F00/7F25/6F47	TR	ALW – ADM – ADM – ADM	Variable	O	D = 'FF'
$EF_{ME3GPDOPC}$	3F00/7F25/6F48	TR	CHV1 – CHV1 – ADM – ADM	1	O	D = '00'
$EF_{3GPDOPM}$	3F00/7F25/6F49	TR	CHV1 – ADM – ADM – ADM	1	O	Specified by Operator
EF_{SIPCAP}	3F00/7F25/6F4A	TR	CHV1 – ADM – ADM – ADM	4	O	Specified by Operator
EF_{MIPCAP}	3F00/7F25/6F4B	TR	CHV1 – ADM – ADM – ADM	5	O	Specified by Operator
EF_{SIPUPP}	3F00/7F25/6F4C	TR	CHV1 – ADM – ADM – ADM	Variable	O	Specified by Operator
EF_{MIPUPP}	3F00/7F25/6F4D	TR	CHV1 – ADM – ADM – ADM	Variable	O	Specified by Operator
EF_{SIPSP}	3F00/7F25/6F4E	TR	CHV1 – CHV1 – ADM – ADM	1	O	Specified by Operator
EF_{MIPSP}	3F00/7F25/6F4F	TR	CHV1 – CHV1 – ADM – ADM	Variable	O	Specified by Operator
$EF_{SIPPAPSS}$	3F00/7F25/6F50	TR	CHV1 – CHV1 – ADM – ADM	Variable	O	Specified by Operator
Simple IP CHAP SS	-	-	Never – Never – Ø – Ø	Variable	O	Specified by Operator
Mobile IP SS	-	-	Never – Never – Ø – Ø	Variable	O	Specified by Operator
Shared Secret	-	-	Never – Never – Ø – Ø	Variable	O	Specified by Operator
EF_{PUZL}	3F00/7F25/6F53	TR	CHV1 – ADM – ADM – ADM	Variable	O	Specified by Operator
$EF_{MAXPUZL}$	3F00/7F25/6F54	TR	CHV1 – ADM – ADM – ADM	5	O	Specified by Operator
EF_{MECRP}	3F00/7F25/6F55	TR	CHV1 – CHV1 – ADM – ADM	3	M	D = '00 00 00'
$EF_{HRPDCAP}$	3F00/7F25/6F56	TR	CHV1 – ADM – ADM – ADM	2	O	Specified by Operator
$EF_{HRPDUPP}$	3F00/7F25/6F57	TR	CHV1 – ADM – ADM – ADM	Variable	O	Specified by Operator
HRPD AA CHAP SS	-	-	Never – Never – Ø – Ø	Variable	O	Specified by Operator
EF_{CSSPR}	3F00/7F25/6F58	TR	CHV1 – ADM – ADM – ADM	1	O	D = 'FF'

File Name	*File ID*	*File Type*	*Access – Read – Update – Invalidate – Rehabilitate*	*Size in Bytes*	*Mandatory or Optional*	*Default Values (D) and/or Parameter Values (P) in Bytes*
EF_{ATC}	3F00/7F25/6F59	TR	CHV1 – ADM – ADM – ADM	1	O	Specified by Operator
EF_{EPRL}	3F00/7F25/6F5A	TR	CHV1 – ADM – ADM – ADM	Variable	O	Specified by Operator
$EF_{BCSMScfg}$	3F00/7F25/6F5B	TR	CHV1 – ADM – ADM – ADM	1	O	Specified by Operator
$EF_{BCSMSpref}$	3F00/7F25/6F5C	TR	CHV1 – CHV1 – ADM – ADM	1	O	D = 'FF'
$EF_{BCSMStable}$	3F00/7F25/6F5D	LF	CHV1 – ADM – ADM – ADM	Variable	O	D = '00 FF…FF'
EF_{BCSMSp}	3F00/7F25/6F5E	LF	CHV1 – CHV1 – ADM – ADM	2	O	D = 'FF FF'
EF_{IMPI}	3F00/7F25/6F5F	TR	CHV1 – ADM – ADM – ADM	Variable	O	Specified by Operator
EF_{DOMAIN}	3F00/7F25/6F60	TR	CHV1 – ADM – ADM – ADM	Variable	O	Specified by Operator
EF_{IMPU}	3F00/7F25/6F61	LF	CHV1 – ADM – ADM – ADM	Variable	O	Specified by Operator
EF_{PCSCF}	3F00/7F25/6F62	LF	CHV1 – ADM – ADM – ADM	Variable	O	Specified by Operator
$EF_{BAKPARA}$	3F00/7F25/6F63	LF	CHV1 – ADM – ADM – ADM	Variable	O	Specified by Operator
$EF_{UpBAKPARA}$	3F00/7F25/6F64	CY	CHV1 – ADM – ADM – ADM	Variable	O	Specified by Operator
EF_{MMSN}	3F00/7F25/6F65	LF	CHV1 – CHV1 – ADM – ADM	Variable	O	D='00 00 00 FF…FF'
EF_{EXT8}	3F00/7F25/6F66	LF	CHV1 – CHV1 – ADM – ADM	Variable	O	D='FF…FF'
EF_{MMSICP}	3F00/7F25/6F67	TR	CHV1 – ADM – ADM – ADM	Variable	O	D='FF…FF'
EF_{MMSUP}	3F00/7F25/6F68	LF	CHV1 – CHV1 – ADM – ADM	Variable	O	D='FF…FF'
EF_{MMSUCP}	3F00/7F25/6F69	TR	CHV1 – CHV1/2 – ADM – ADM	Variable	O	D= 'FF…FF'
$EF_{AuthCapability}$	3F00/7F25/6F6A	LF	CHV1 – ADM – ADM – ADM	Variable	O	D= '00…00'
EF_{3GCIK}	3F00/7F25/6F6B	TR	CHV1 – ADM – ADM – ADM	32	O	Specified by Operator
EF_{DCK}	3F00/7F25/6F6C	TR	CHV1 – CHV1 – ADM – ADM	20	O	Specified by Operator
EF_{GID1}	3F00/7F25/6F6D	TR	CHV1 – ADM – ADM – ADM	N	O	Specified by Operator
EF_{GID2}	3F00/7F25/6F6E	TR	CHV1 – ADM – ADM – ADM	N	O	Specified by Operator
$EF_{CDMACNL}$	3F00/7F25/6F6F	TR	CHV1 – ADM – ADM – ADM	7N	O	Specified by Operator
EF_{HOME_TAG}	3F00/7F25/6F70	TR	CHV1 – ADM – ADM – ADM	X	M	Specified by Operator
EF_{GROUP_TAG}	3F00/7F25/6F71	TR	CHV1 – ADM – ADM – ADM	GROUP_TAG_ LIST_SIZE	M	Specified by Operator
$EF_{SPECIFIC\_TAG}$	3F00/7F25/6F72	TR	CHV1 – ADM – ADM – ADM	SPEC_TAG_ LIST_SIZE	M	Specified by Operator
EF_{CALL_PROMPT}	3F00/7F25/6F73	TR	CHV1 – ADM – ADM – ADM	CALL_PRMPT _LIST_SIZE	M	Specified by Operator
EF_{SF_EUIMID}	3F00/7F25/6F74	TR	ALW – Never – Never-Never	7	O	Specified by R-UIM Manufacturer
$EF_{AppLabels}$	3F00/7F25/6F92	TR	CHV1 – ADM – ADM – ADM	Variable	O	Specified by Operator
EF_{Model}	3F00/7F25/6F90	TR	CHV1 – CHV1 – ADM – ADM	126	O	D='FF…FF'
EF_{RC}	3F00/7F25/6F91	TR	ALW – ADM – ADM – ADM	Variable	O	Specified by Operator
EF_{SMSCAP}	3F00/7F25/6F76	TR	CHV1 – ADM – ADM – ADM	4	O	Specified by Operator
$EF_{MIPFlags}$	3F00/7F25/6F78	TR	CHV1 – ADM – ADM – ADM	1	O	Specified by Operator

File Name	*File ID*	*File Type*	*Access – Read – Update – Invalidate – Rehabilitate*	*Size in Bytes*	*Mandatory or Optional*	*Default Values (D) and/or Parameter Values (P) in Bytes*
$EF_{3GPDUPPExt}$	3F00/7F25/6F7D	TR	CHV1 – ADM – ADM – ADM	Variable	O	Specified by Operator
$EF_{IPV6CAP}$	3F00/7F25/6F77	TR	CHV1 – ADM – ADM – ADM	21	O	Specified by Operator
$EF_{TCPConfig}$	3F00/7F25/6F79	TR	CHV1 – ADM – ADM – ADM	2	O	Specified by Operator
EF_{DGC}	3F00/7F25/6F7A	TR	CHV1 – ADM – ADM – ADM	3	O	Specified by Operator
$EF_{WAPBrowserCP}$	3F00/7F25/6F7B	TR	CHV1 – ADM – ADM – ADM	Variable	O	Specified by Operator
$EF_{WAPBrowserBM}$	3F00/7F25/6F7C	TR	CHV1 – CHV1 – ADM – ADM	Variable	O	D='FF…FF'
$EF_{MMSConfig}$	3F00/7F25/6F7E	TR	CHV1 – ADM – ADM – ADM	8	O	Specified by Operator
EF_{JDL}	3F00/7F25/6F7F	TR	CHV1 – ADM – ADM – ADM	Variable	O	Specified by Operator

1

Annex B (INFORMATIVE): BCMCS-RELATED TAG VALUES

Tag	Name of Data Element	Usage
'80'	BCMCS Flow ID TLV object	BCMCS command
'81'	BAK ID TLV object	BCMCS command
'82'	RAND, SK RAND or TK RAND TLV objects	BCMCS command
'83'	BAK Expire TLV object	BCMCS command
'84'	Packet Index TLV object	BCMCS command
'85'	SK TLV object or SRTP SK TLV object	BCMCS command
'86'	Timestamp TLV object	BCMCS command
'87'	Auth Signature TLV object	BCMCS command
'88'	Challenge TLV object	BCMCS command
'89'	Digest Response TLV object	BCMCS command

Annex C (INFORMATIVE): ESN AND MEID CONFIGURATIONS

The various possibilities of configuring the ESN and MEID are described here.

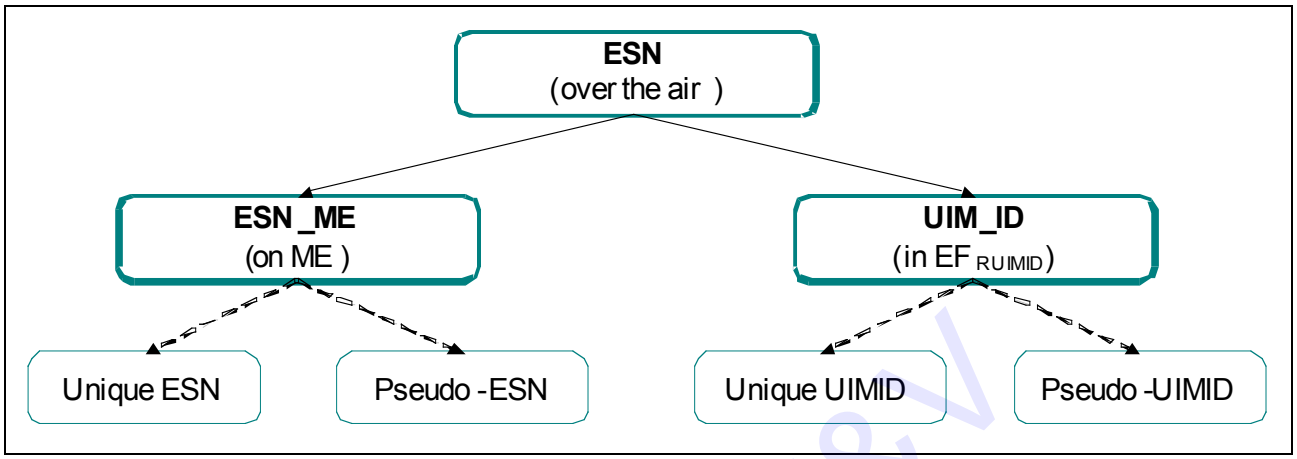


Figure 13. ESN Configurations

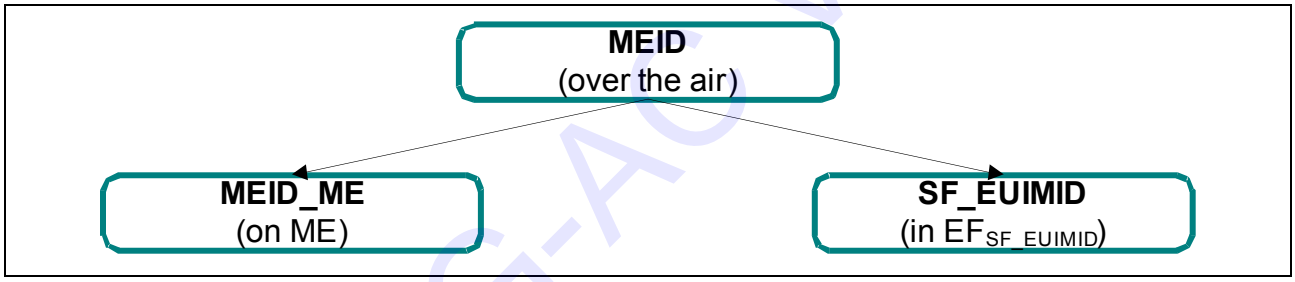
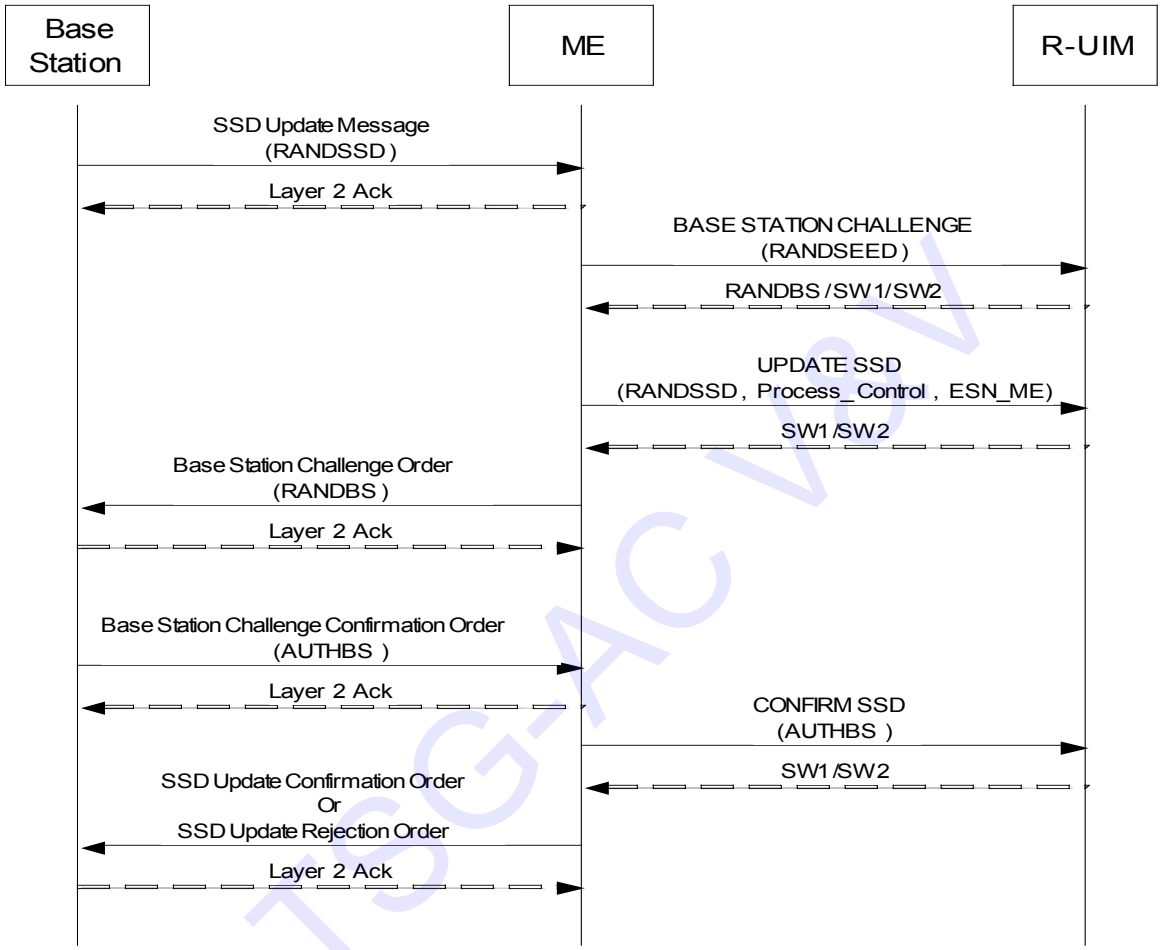


Figure 14. MEID Configurations

Annex D (INFORMATIVE): CALL-FLOW FOR SSD UPDATE

The entire call flow for SSD Update is described below. Note the differences in command names and sequence order in the call flow between the ME and R-UIM versus that between the Base Station and ME (defined in [5]).



Annex E (INFORMATIVE): SP_LOCK_STATE IN THE R-UIM

This Annex describes SP_LOCK_STATE in the R-UIM during an OTASP session as a function of:

i) the successfully executed commands,

ii) EF_{SPC} and

iii) bit 1 of EF_{SPCS}.

It is based on Sec. 3.2.2.3 and Sec. 3.3.1.10 Validation Request Message of [7].

The tables below show different types of OTAPA sessions. The columns EF_{SPC}, EF_{SPCS} and SP_LOCK_STATE represent the values after a command is successfully executed. In the following tables, '*nn nn nn*' represents any non-default value for SPC and "Any" represents either '0' or '1'.

The following table shows SP_LOCK_STATE for an OTAPA session with a default SPC and where the SPC is not changed.

Table 12. SPC_LOCK_STATE, default SPC and no change to SPC

Step	Command	EF_{SPC}	EF_{SPCS}	SP_LOCK_STATE
0	[Initial Conditions]	'00 00 00'	'00'	Any
1	OTAPA REQUEST • Start/Stop = '80' (Start) • Other parameters as required	'00 00 00'	'00'	0
…	(Any X ≥ 0 Commands)	'00 00 00'	'00'	0
2+X	OTAPA REQUEST • Start/Stop = '00' (Stop)	'00 00 00'	'00'	Any

The following table shows SP_LOCK_STATE for an OTAPA session that changes SPC from the default value to a non-default value. Note that SP_LOCK_STATE does not become 1 during this OTAPA session that changes SPC from the default to non-default value. SP_LOCK_STATE becomes 1 during the next OTAPA session where the R-UIM is configured with a non-default SPC (see the subsequent table "SPC_LOCK_STATE, non-default SPC and change SPC to default SPC") since SP_LOCK_STATE is determined by the SPC value in EF_{SPC} at the "Start" of an OTASP session.

Table 13. SPC_LOCK_STATE, default SPC and changing SPC to a non-default SPC

Step	Command	EF_{SPC}	EF_{SPCS}	SP_LOCK_STATE
0	[Initial Conditions]	'00 00 00'	'00'	Any

Step	Command	EF_{SPC}	EF_{SPCS}	SP_LOCK_STATE
1	OTAPA REQUEST • Start/Stop = '80' (Start) • Other parameters as required	'00 00 00'	'00'	0
…	(Any X ≥ 0 Commands)	'00 00 00'	'00'	0
2+X	VALIDATE • Block ID = '01' (Change SPC) • Block Length = '03' • Param Data = '*nn nn nn*'	'00 00 00'	'00'	0
…	(Any Y ≥ 0 Commands)	'00 00 00'	'00'	0
3+X+Y	COMMIT	'*nn nn nn*'	'01'	0
…	(Any Z ≥ 0 Commands)	'*nn nn nn*'	'01'	0
4+X+Y +Z	OTAPA REQUEST • Start/Stop = '00' (Stop)	'*nn nn nn*'	'01'	Any

The following table shows SP_LOCK_STATE for an OTAPA session that changes SPC from a non-default value to the default value.

Table 14. SPC_LOCK_STATE, non-default SPC and change SPC to default SPC

Step	Command	EF_{SPC}	EF_{SPCS}	SP_LOCK_STATE
0	[Initial Conditions]	'*nn nn nn*'	'01'	Any
1	OTAPA REQUEST • Start/Stop = '80' (Start) • Other parameters as required	'*nn nn nn*'	'01'	1
…	(Any W ≥ 0 Commands)	'*nn nn nn*'	'01'	1
2+W	VALIDATE • Block ID = '00' (Verify SPC) • Block Length = '03' • Param Data = '*nn nn nn*'	'*nn nn nn*'	'01'	0
…	(Any X ≥ 0 Commands)	'*nn nn nn*'	'01'	0
3+W+X	VALIDATE • Block ID = '01' (Change SPC) • Block Length = '03' • Param Data = '00 00 00'	'00 00 00'	'01'	0
…	(Any Y ≥ 0 Commands)	'00 00 00'	'01'	0
4+W+X +Y	COMMIT	'00 00 00'	'00'	0

Step	Command	EF_{SPC}	EF_{SPCS}	SP_LOCK_STATE
…	(Any Z ≥ 0 Commands)	'00 00 00'	'00'	0
5+W+X +Y+Z	OTAPA REQUEST • Start/Stop = '00' (Stop)	'00 00 00'	'00'	Any

1

Annex F (INFORMATIVE): CONFIGURATION REQUEST AND DOWNLOAD REQUEST PARAMETER TO EF MAPPING

Mapping with Block ID = '00' (CDMA/Analog NAM)

Parameters for Block ID = '00'	**EF for Storage**
IMSI_M_CLASS, IMSI_M_ADDR_NUM, MCC_M, IMSI_M_11_12, IMSI_M_S*	EF_{IMSI_M}
HOME_SIDp	EF_{AH} (if present)
Extended Address	EF_{AOP} (if present)
CDMA Home SID (SIDp) and CDMA Home NID (NIDp) (Single or multiple pairs)	$EF_{CDMAHOME}$
Access Overload Class	EF_{ACCOLC}
MOB_TERM_FOR_NIDp, MOB_TERM_FOR_SIDp, MOB_TERM_HOMEp	EF_{TERM}
SCM*, MOB_P_REV**, LOCAL_CONTROL_ANALOG	EF_{MECRP}

* Although IMSI_M_PROGRAMMED in EFIMSI_M is not part of the DOWNLOAD REQUEST message, it is set according to Sec. 4.5.3.

** SCM and MOB_P_REV are needed for CONFIGURATION REQUEST but not needed for DOWNLOAD REQUEST.

Mapping with Block ID = '01' (Mobile Directory Number)

Parameters for Block ID = '01'	**EF for Storage**
MDN	EF_{MDN}

Mapping with Block ID = '02' (CDMA NAM)

Parameters for Block ID = '02'	**EF for Storage**
IMSI_M_CLASS, IMSI_M_ADDR_NUM, MCC_M, IMSI_M_11_12, IMSI_M_S*	EF_{IMSI_M}
CDMA Home SID (SIDp) and CDMA Home NID (NIDp) (Single or multiple pairs)	$EF_{CDMAHOME}$
Access Overload Class	EF_{ACCOLC}
MOB_TERM_FOR_NIDp, MOB_TERM_FOR_SIDp, MOB_TERM_HOMEp	EF_{TERM}
Bit 6 of SCM (Slotted Mode)**, MOB_P_REV*, LOCAL_CONTROL_CDMA	EF_{MECRP}

* Although IMSI_M_PROGRAMMED in EF_{IMSI_M} is not part of the DOWNLOAD REQUEST message, it is set according to Sec. 4.5.3.

** Bit 6 of SCM and MOB_P_REV are needed for CONFIGURATION REQUEST but not needed for DOWNLOAD REQUEST.

Mapping with Block ID = '03' (IMSI_T)

Parameters for Block ID = '03'	**EF for Storage**
IMSI_T_CLASS, IMSI_T_ADDR_NUM, MCC_T, IMSI_T_11_12, IMSI_T_S	EF_{IMSI_T}

* Although IMSI_T_PROGRAMMED in EF_{IMSI_T} is not part of the DOWNLOAD REQUEST message, it is set according to Sec. 4.5.3.