3GPP2 C.S0023-0

3rd Generation Partnership Project 2 "3GPP2"

Date: June 9, 2000

Removable User Identity Module (R-UIM) for cdma2000 Spread Spectrum Systems

Intentionally left blank.

1 Introduction

1.1 General Description

This document contains the requirements for the Removable User Identity Module (R-UIM). It is an extension of Subscriber Identity Module (SIM), per latest GSM 11.11 capabilities, to enable operation in a [11/14/15] radiotelephone environment. Examples of this environment include, but are not limited to, analog, [11, 14] -based CDMA, and the [1] 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 [11/14/15] 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 [11/14/15] environment.

This standard specifies security-related procedures and commands, along with data and information storage items that permit basic operation in the [11/14/15] environment. Later versions are expected to also address the delivery of [11/14/15] user features and services via the R-UIM.

Although the focus of this document is compatibility with [11/14/15], the scope of this document may later be expanded to include compatibility with other [15] -related technologies such as TDMA and AMPS.

1.2 Terms

AC. See Authentication Center.

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.

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.)

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

CRC. See Cyclic Redundancy Code.

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. R-UIM Dedicated File.

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

EF. R-UIM Elementary File.

Electronic Serial Number (ESN) . A 32-bit number assigned by the mobile station manufacturer, uniquely identifying the mobile station equipment.

ESN. See Electronic Serial Number.

HLR. See Home Location Register.

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.

IMSI. See International Mobile Subscriber Identity.

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

IMSI_O. The operational value of IMSI used by the mobile station for operation with the base station.

IMSI_T. "True" IMSI not associated with MIN, 15-digits or fewer.

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

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.

LSB. Least significant bit.

M/O. Mandatory/Optional.

MCC. See Mobile Country Code

ME. Mobile Equipment.

MF. R-UIM 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.

MIN. See Mobile Identification Number.

MNC. See Mobile Network Code.

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. 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.

NAM. See Number Assignment Module.

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.

NID. See Network Identification.

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

OTAF. See Over-the-Air Service Provisioning Function.

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

OTAPA. See Over-the-Air Parameter Administration.

OTASP. See Over-the-Air Service Provisioning.

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.

Phase. Revision level of the R-UIM.

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.

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.

R-UIM. Removable UIM.

Service Option. A service capability of the system. Service options may be applications such as voice, data, or facsimile. See [10].

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.

SID. See System Identification.

SIM. Subscriber Identity Module.

SPASM. See Subscriber Parameter Administration Security Mechanism.

SPC. Service Programming Code.

SSD. See Shared Secret Data.

SSPR. See System Selection for Preferred Roaming.

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-p}).

TMSI. Temporary Mobile Station Identity.

UCS2. Universal Multiple-Octet Coded Character Set.

UIM. User Identity Module.

VPM. Voice Privacy Mask.

1.3 References

1. C.S0001-A, “Introduction to cdma2000 Spread Spectrum Systems”, March 2000.

2. C.S0002-A, “Physical Layer Standard for cdma2000 Spread Spectrum Systems”, March 2000.

3. Reserved.

4. C.S0004-A, “Signaling Link Access Control (LAC) Standard for cdma2000 Spread Spectrum Systems”, March 2000.

5. C.S0005-A, “Upper Layer (Layer 3) Signaling Standard for cdma2000 Spread Spectrum Systems”, March 2000.

6. Reserved.

7. C.S0016-0, “Over-the-Air Service Provisioning of Mobile Stations in Spread Spectrum Systems”, June 1998.

8. C.S0016-0, “Short Message Service for Spread Spectrum Systems”, April 1999.

9. ITU-T Recommendation E.212, “Identification Plan for Land Mobile Stations”, 1988.

10. C.R1001-0, “Administration of Parameter Value Assignments for TIA/EIA Wideband Spread Spectrum Standards”, December 1999.

11. TIA/EIA/IS-95-A, Mobile Station – Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular Systems”, May 1995.

12. TIA/EIA/IS-683 Annex A, OTASP CCA, March 1996.

13. TIA/EIA/IS-683 A Annex A, OTASP CCA, January 1998.

14. TIA/EIA-95-B, “Mobile Station - Base Station Compatibility Standard for Dual-Mode Wideband Spread Cellular Systems”, December 1998.

15. ANSI-TIA/EIA-41, “Cellular Radio-Telecommunications Intersystem Operations”, 1997.

16. TIA/EIA-91, “Mobile Station - Base Station Compatibility Standard for 800 MHz Analog Cellular”, October 1994.

17. GSM 11.11; “Digital cellular telecommunications system (Phase 2+); “Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) Interface”.

18. GSM 11.12; “Digital cellular telecommunications system (Phase 2); Specification of the 3 volt Subscriber Identity Module – Mobile Equipment (SIM-ME) Interface”.

19. GSM 11.18; “Digital cellular telecommunications system; Specification of the 1.8 volt Subscriber Identity Module – Mobile Equipment (SIM-ME) Interface”.

(Note: References [17], [18], and [19] are to the latest published version of the ETSI documents.)

2 Physical, Electrical, and Logical Interfaces

2.1 Physical Interface

The physical characteristics of the R-UIM shall follow the definitions specified in the sections of GSM 11.11 shown in Table 2.1-1 Physical Characteristics.

Table 2.1-1 Physical Characteristics

Section of GSM 11.11	Title
4	*Physical Characteristics*
4.1	Format and Layout
4.1.1	ID-1 SIM
4.1.2	Plug-In SIM, including Annex A (Normative)
4.2	Temperature range for card operations
4.3	Contacts
4.3.1	Provision of contacts
4.3.2	Activation and deactivation
4.3.3	Inactive contacts
4.3.4	Contact pressure
4.4	Precedence (Informative)
4.5	Static protection

2.2 Electrical Interface

The electrical characteristics of the R-UIM shall follow the definitions specified in the sections of GSM 11.11 shown in Table 2.2-1.

Table 2.2-1 Electronic Signals and Transmission Protocols

Section of GSM 11.11	Title
5	*Electronic Signals and Transmission Protocols*
5.1	Supply voltage Vcc (contact C1)
5.2	Reset (RST) (contact C2)
5.3	Programming voltage Vpp (contact C6)
5.4	Clock CLK (contact C3)
5.5	I/O (contact C7)
5.6	States
5.7	Baudrate
5.8	Answer To Reset (ATR)
5.8.1	Structure and contents
5.8.2	PPS procedure
5.8.3	Speed enhancement
5.9	Bit/character duration and sampling time
5.10	Error handling
Annex A	*Plug-In SIM*

2.3 Logical Interface

The logical interface of the R-UIM shall follow the definitions specified in the sections of GSM 11.11 shown in Table 2.3-1. The Dedicated file ID for CDMA (used for EFs in section 3.4) is 7F25.

Table 2.3-1 Logical Model

Section of GSM 11.11	Title
6	*Logical Model*
6.1	General description
6.2	File identifier
6.3	Dedicated files
6.4	Elementary files
6.4.1	Transparent EF
6.4.2	Linear fixed EF
6.4.3	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 Authentication and key generation procedure

See section 4.1 and section 4.2.

2.4.2 Algorithms and processes

The algorithm used by the R-UIM is CAVE (see section 4.1 and section 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 GSM 11.11 shown in Table 2.4-1

Table 2.4-1 File access conditions

Section of GSM 11.11	Title
7	File Access Conditions

2.5 Function Description

The functions of the R-UIM shall follow the definitions specified in the sections of GSM 11.11 shown in Table 2.5-1. For [15], the following functions from section 4 are used: Base Station Challenge, Update SSD, Run CAVE, and Generate Key/VPM.

Table 2.5-1 Description of the Functions

Section of GSM 11.11	Title
8	*Description of The Functions*
8.1	SELECT
8.2	STATUS
8.3	READ BINARY
8.4	UPDATE BINARY
8.5	READ RECORD
8.6	UPDATE RECORD
8.7	SEEK
8.8	INCREASE
8.9	VERIFY CHV
8.10	CHANGE CHV
8.11	DISABLE CHV
8.12	ENABLE CHV
8.13	UNBLOCK CHV
8.14	INVALIDATE
8.15	REHABILITATE
8.17	SLEEP
8.18	TERMINAL PROFILE
8.19	ENVELOPE
8.20	FETCH
8.21	TERMINAL RESPONSE

2.6 Command Description

The commands used with the R-UIM shall follow the definitions specified in the sections of GSM 11.11 shown in Table 2.6-1. The commands used to run CAVE are specified in section 4.4.

Table 2.6-1 Description of the Commands (Part 1 of 2)

Section of GSM 11.11	**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

Table 2.6-1 Description of the Commands (Part 2 of 2)

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
9.4.1	Responses to commands which are correctly executed
9.4.2	Responses to commands which are postponed
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

2.7 Content of EFs

The content of the EFs of the R-UIM shall include the sections of GSM 11.11 shown in Table 2.7-1.

Table 2.7-1 Content of EFs

Section of GSM 11.11	**Title**
10.1	Contents of the EFs at the MF level
10.1.1	EF_{ICCID} (ICC Identification)
10.2	DFs at the GSM application level
10.5	Contents of files at the telecom level
10.5.1	EF_{ADN} (Abbreviated dialling numbers)
10.5.2	EF_{FDN} (Fixed dialling numbers)
10.5.5	EF_{MSISDN}
10.5.8	EF_{LND} (Last number dialled)
10.5.9	EF_{SDN} (Service Dialling Numbers)
10.5.10	EF_{EXT1} (Extension1)
10.5.11	EF_{EXT2} (Extension2)
10.5.12	EF_{EXT3} (Extension3)
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

* The number stored in EF_{MSISDN} is used as the MDN in [15] systems.

2.8 Application Protocol

The application protocol of the R-UIM shall follow the definitions specified in the sections of GSM 11.11 shown in Table 2.8-1.

Table 2.8-1 Application Protocol

Section of GSM 11.11	Title
11	*Application protocol*
11.1	General procedures
11.1.1	Reading an EF
11.1.2	Updating an EF
11.1.3	Increasing an EF
11.2.5	Administrative information request
11.2.6	SIM service table request
11.2.7	SIM revision request
11.2.8	SIM Presence Detection and Proactive Polling

2.9 R-UIM Application Toolkit

(Reserved)

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

Figure 3 depicts the multi-mode R-UIM file structure.

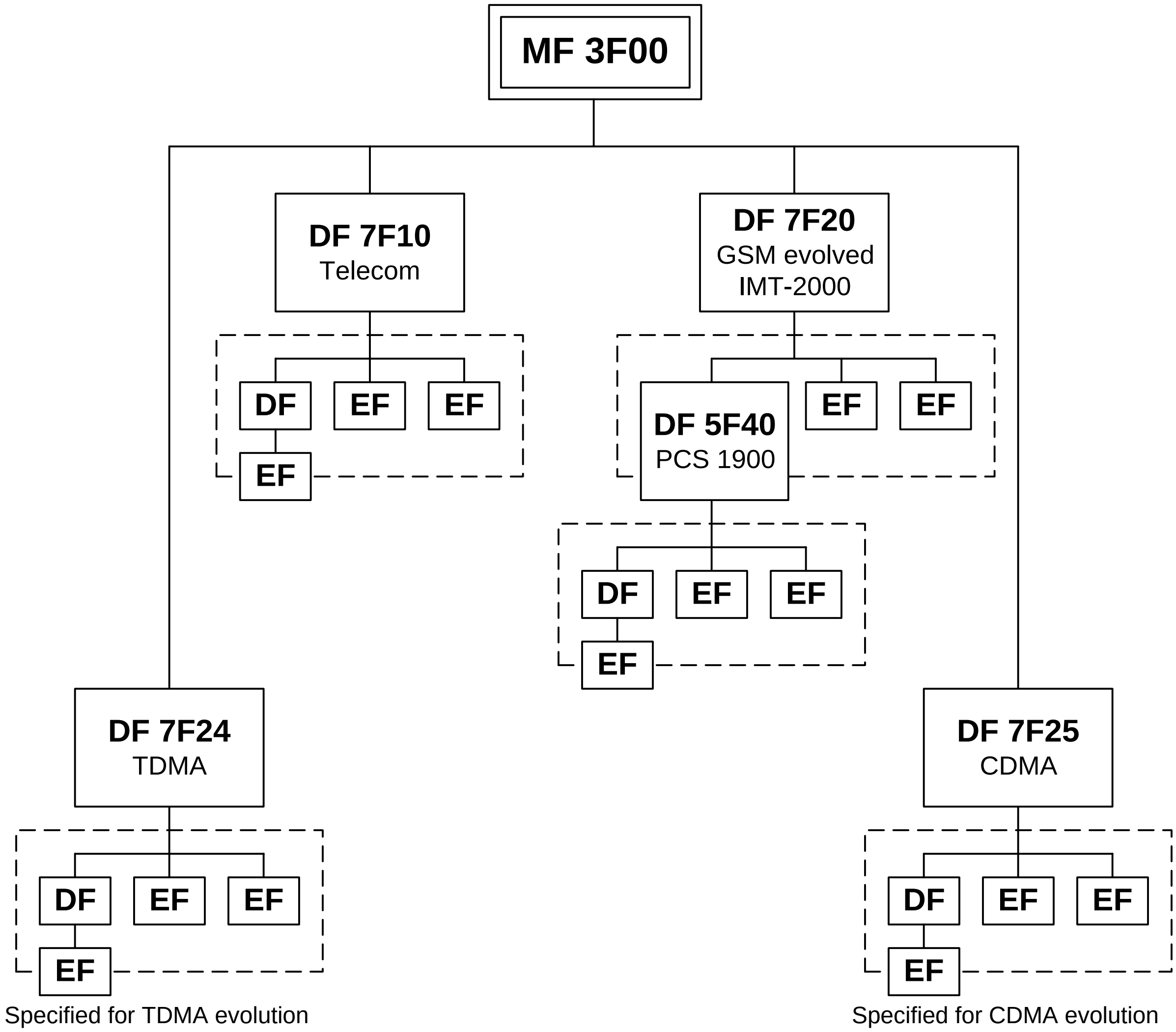

Figure 3 Dedicated File Structure

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 [11/14]-based CDMA, and the [1] family of standards.

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 3.4.

The first byte identifies the type of file. The numbering scheme for DFs and Efs is inherited from GSM 11.11 as:

- '3F': Master File;
- '7F': 1^{st} level Dedicated File;
- '5F': 2^{nd} level Dedicated File;
- '2F': Elementary File under the Master File;
- '6F': Elementary File under the 1^{st} level Dedicated File;
- '4F': Elementary File under the 2^{nd} 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.

Dedicated Files:

- administrative use:
  '7F 4X', '5F1X', '5F2X'
- operational use:
  '7F 10' ($DF_{TELECOM}$), '7F 20' (DF_{GSM}), '7F 21' ($DF_{DCS1800}$), '7F 22' ($DF_{IS\text{-}41}$), '7F 23' ($DF_{FP\text{-}CTS}$), '7F 24' ($DF_{TIA/EIA\text{-}136}$), '7F 25' ($DF_{TIA/EIA\text{-}95}$), and '7F 2X', where X ranges from '6' to 'F'.
- reserved under '7F10':
  '5F50' ($DF_{GRAPHICS}$)
- reserved under '7F20':
  '5F30' ($DF_{IRIDIUM}$), '5F31' ($DF_{Globalstar}$), '5F32' (DF_{ICO}), '5F33' (DF_{ACeS}), '5F3X', where X ranges from '4' to 'F' for other MSS.
  '5F40'($DF_{PCS\text{-}1900}$), '5F4Y' where Y ranges from '1' to 'F';
  '5F5X' where X ranges from '0' to 'F';
  '5F60'(DF_{CTS}), '5F6Y' where Y ranges from '1' to 'F';
  '5F70' (DF_{SoLSA}), '5F7Y' where Y ranges from '1' to 'F';
  '5FYX' 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', '5F2X'
  '6F 1X' in the DFs '7F 10', '7F 20', '7F 21', ‘7F 25’;
  '4F 1X' in all 2^{nd} 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 2nd level DFs.
  '2F 1X' in the MF '3F 00'.

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.

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

[11/14] and [1] store parameters in several different types of memory. Variables stored in permanent memory use the subscript p. Variables stored in semi-permanent memory use the subscript s-p. 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 Call Count

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

<table>
<tr><td colspan="2">Identifier: ‘6F21’</td><td colspan="2">Structure: cyclic</td><td>Mandatory</td></tr>
<tr><td colspan="2">File size: 2 bytes</td><td colspan="3">Update Activity: high</td></tr>
<tr><td colspan="5">Access Conditions:

READ CHV
UPDATE CHV
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1-2</td><td colspan="2">COUNT_{s-p}</td><td>M</td><td>2 bytes</td></tr>
</table>

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

3.4.2 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: CHV UPDATE: ADM INVALIDATE: ADM REHABILITATE: ADM			

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.

Byte 1:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                          |<--------'0'=Class 0, '1'=Class 1
 |<--------------------------------------------->|<-----------------RFU
```

Byte 2, byte 3, byte 4, byte 5, and byte 6 are encoded as described in [14], section 6.3.1.1, "Encoding of IMSI_M_S and IMSI_T_S."

Byte 2:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                          |<-------LSB of IMSI_M_S2
 |<--------------------------------------------->|<---------------IMSI_M_S2 bits in ascending order
```

Byte 3:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                            |<-------Next-MSB of IMSI_M_S2
```

```
                                              |<---------------MSB of IMSI_M_S2
  |<------------------------------------>|<--------------------------RFU
```

Byte 4:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                      |<-------LSB of IMSI_M_S1

  |<---------------------------------------------->|<---------------IMSI_M_S1 bits in ascending order
```

Byte 5:

b8	b7	b6	b5	b4	b3	b2	b1

```
  |<------------------------------------------------------>|<----IMSI_M_S1 bits in ascending order
```

Byte 6:

b8	b7	b6	b5	b4	b3	b2	b1

```
        |<--------------------------------------------->|<---IMSI_M_S1 bits in ascending order

  |<--------------------------------------------------------------MSB of IMSI_M_S1
```

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

Byte 7:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                      |<-------LSB of IMSI_M_11_12
              |<------------------------------>|<------------------middle bits of IMSI_M_11_12
        |<-------------------------------------------------------MSB of IMSI_M_11_12
  |<--------------------------------------------------------------RFU
```

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

Byte 8:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                      |<-------LSB of IMSI_M_ADD_NUM
                                              |<---------------middle bit of IMSI_M_ADD_NUM
                                        |<----------------------MSB of IMSI_M_ADD_NUM
        |<-------------------->|<---------------------------------RFU
  |<----------------------------------------------------------------IMSI_M_PROGRAMMED indicator
```

IMSI_M_PROGRAMMED shall be set to '1' if an IMSI_M has been programmed (IMSI_M would contain a MIN for systems that comply with [11]); if an IMSI_M has not been programmed, it shall be set to '0'.

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

Byte 9:

b8	b7	b6	b5	b4	b3	b2	b1

|<-------LSB of MCC_M

|<--->|<----------------MCC_M bits in ascending order

Byte 10:

b8	b7	b6	b5	b4	b3	b2	b1

|<-------Next-MSB of MCC_M

|<----------------MSB of MCC_M

|<--------------------------------------->|<-------------------------RFU

For R-UIM applications in systems that comply with [11], 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 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 CHV UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
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 [14] Sections are identical to those described in Section 3.4.3above, 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 TMSI

This EF stores the Temporary Mobile Station Identity (TMSI). TMSI is assigned by the serving network and consists of 4 components, Assigning TMSI Length, ASSIGNING_TMSI_ZONE$_{s\text{-}p}$, TMSI_CODE$_{s\text{-}p}$, and TMSI_EXP_TIME$_{s\text{-}p}$.

Identifier: '6F24'	Structure: transparent		Mandatory
File size: 16 bytes	Update Activity: high		
Access Conditions: READ CHV UPDATE CHV INVALIDATE ADM REHABILITATE CHV			
Bytes	**Description**	**M/O**	**Length**
1	Assigning TMSI Length	M	1 byte
2-9	ASSIGNING_TMSI_ZONE$_{s\text{-}p}$	M	8 bytes
10-13	TMSI_CODE$_{s\text{-}p}$	M	4 bytes
14-16	TMSI_EXP_TIME$_{s\text{-}p}$	M	3 bytes

Byte 1:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                   |<-------LSB of TMSI Length
                                       |<----->|-------------middle bits of TMSI Length
                                 |<----------------------MSB of TMSI Length
  |<------------------->|<----------------------------RFU
```

Bytes 2 through 9 store the (up to) 8 octet TMSI Zone as described in Sections 6.3.15, 6.3.15.1, and 6.3.15.2 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 [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 [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 Analog Home SID

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

Identifier: ‘6F25’	Structure: transparent	Mandatory
File size: 2 bytes	Update Activity: low	
Access Conditions:		
READ	CHV	
UPDATE	CHV	
INVALIDATE	ADM	
REHABILITATE	ADM	

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

Byte 1:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                  |<-------LSB of SID
 |<------------------------------------------->|<----------------SID bits in ascending order
```

Byte 2:

b8	b7	b6	b5	b4	b3	b2	b1

```
                |<------------------------------------>|<---------SID bits in ascending order
       |<--------------------------------------------------MSB of SID
 |<----------------------------------------------------------RFU
```

3.4.6 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	Mandatory
File size: 1 byte		Update Activity: low	
Access Conditions: READ UPDATE INVALIDATE REHABILITATE		 CHV CHV ADM ADM	
Bytes	Description	M/O	Length
1	Analog Operational Parameters (Ex_p, LCM, GID)	M	1 byte

Byte 1:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                      |<-------Extended Address
                                                |<----------------Local Use Mark
                      |<--------------------->|<-----------------------Group ID
 |<----->|<--------------------------------------------------RFU
```

3.4.7 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		Mandatory
File size: 7 bytes			Update Activity: high	
Access Conditions: READ CHV UPDATE CHV 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

Byte 1:

b8	b7	b6	b5	b4	b3	b2	b1

|<-------LSB of $NXTREG_{s-p}$

|<-->|<----------------$NXTREG_{s-p}$ bits in ascending order

Byte 2:

b8	b7	b6	b5	b4	b3	b2	b1

|<--->|<-------$NXTREG_{s-p}$ bits in ascending order

Byte 3:

b8	b7	b6	b5	b4	b3	b2	b1

|<-------------------->|<--------$NXTREG_{s-p}$ bits in ascending order

|<------------------------------------MSB of $NXTREG_{s-p}$

|<------------->|---RFU

Byte 4:

b8	b7	b6	b5	b4	b3	b2	b1

|<-------LSB of SID_{s-p}

|<--->|<----------------SID_{s-p} bits in ascending order

Byte 5:

b8	b7	b6	b5	b4	b3	b2	b1

|<------------------------------------>|<-------SID_{s-p} bits in ascending order

|<---MSB of SID_{s-p}

|<--RFU

Byte 6:

b8	b7	b6	b5	b4	b3	b2	b1

|<-------LSB of $LOCAID_{s-p}$

|<--->|<----------------$LOCAID_{s-p}$ bits in ascending order

Byte 7:

b8	b7	b6	b5	b4	b3	b2	b1

|<------------>|<---------$LOCAID_{s-p}$ bits in ascending order

|<----------------------------MSB of $LOCAID_{s-p}$

|<----------->|<-------------------------------------RFU

|<--PUREG

3.4.8 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

File size: 5 x N bytes, N = number of records	Update Activity: low

Access Conditions:

READ	CHV
UPDATE	CHV
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1-2	CDMA home SID (SID_p)	M	2 bytes
3-4	CDMA home NID (NIDp)	M	2 bytes
5	Band Class	M	1 byte

Byte 1:

B8	b7	b6	b5	b4	b3	b2	b1

|<-------LSB of SID
|<--->|<----------------SID bits in ascending order

Byte 2:

B8	b7	b6	b5	b4	b3	b2	b1

|<--------------------------------------->|<--------SID bits in ascending order
|<---MSB of SID
|<---RFU

Byte 3:

B8	b7	b6	b5	b4	b3	b2	b1

|<-------LSB of NID
|<--->|<-------------NID bits in ascending order

Byte 4:

B8	b7	b6	b5	b4	b3	b2	b1

|<-->|<-------NID bits in ascending order
|<--MSB of NID

Byte 5:

B8	b7	b6	b5	b4	b3	b2	b1

|<--------------------->|<-------Band Class
|<---------------------------->|---------------------------------- RFU

3.4.9 CDMA Zone-Based Registration Indicators

This EF stores eight entries in the zone-based registration list “ZONE_LIST.” Each stored element includes a REG_ZONE, a corresponding SID, NID pair, a Band Class/Frequency Block identifier, and a ZONE_TIMER. Details are described in [14] Sections 6.3.4, 6.6.5.1.5, and 6.6.5.5, titled “Registration Memory”, “Zone-Based Registration”, and “Registration Procedures”, respectively.

Identifier: ‘6F29”	Structure: transparent	Mandatory
File size: 65 bytes	Update Activity: high	
Access Conditions: READ CHV UPDATE CHV INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	TOTAL_ZONES	M	1 byte
2-3	REG_ZONE	M	2 bytes
4-5	SID	M	2 bytes
6-7	NID	M	2 bytes
8	Frequency Block	M	1 byte
9	Band Class/ZONE_TIMER	M	1 byte
	………		
58-59	REG_ZONE	M	2 bytes
60-61	SID	M	2 bytes
62-63	NID	M	2 bytes
64	Frequency Block	M	1 byte
65	Band Class/ZONE_TIMER	M	1 byte

Byte 1:

b8	b7	b6	b5	b4	b3	b2	b1

|<-------LSB of TOTAL_ZONES
|<--------------Middle bit of TOTAL_ZONES
|<--------------MSB of TOTAL_ZONES
|<------------------------------>|--------------------------RFU

Byte 2:

b8	b7	b6	b5	b4	b3	b2	b1

|<-------LSB of REG_ZONE
|<-->|<----------------REG_ZONE bits in ascending order

Byte 3:

b8	b7	b6	b5	b4	b3	b2	b1

|<----------->|<---------REG_ZONE bits in ascending order
|<------------------------------MSB of REG_ZONE
|<--------------------->|--RFU

Byte 4:

b8	b7	b6	b5	b4	b3	b2	b1

|<-------LSB of SID
|<--->|<---------------SID bits in ascending order

Byte 5:

b8	b7	b6	b5	b4	b3	b2	b1

|<------------------------------------->|<----------SID bits in ascending order
|<--MSB of SID
|<--RFU

Byte 6:

b8	b7	b6	b5	b4	b3	b2	b1

|<-------LSB of NID
|<--->|<---------------NID bits in ascending order

Byte 7:

b8	b7	b6	b5	b4	b3	b2	b1

|<--->|<---------NID bits in ascending order
|<--MSB of NID

Byte 8:

b8	b7	b6	b5	b4	b3	b2	b1

|<------------->|<----------------Frequency Block: ‘000’ Block A
|<------------------------------>|<--RFU ‘001’ Block B
‘010’ Block C
‘011’ Block D
‘100’ Block E
‘101’ Block F
All others RFU

Note: Frequency Block entry is ignored if Band Class is ‘00000’.

Byte 9:

b8	b7	b6	b5	b4	b3	b2	b1

|<------------->|<------ZONE_TIMER
|<------------------------------>|<------------------------------Band Class
‘00000’ 800 MHz
‘00001’ PCS
: as defined in [2]
‘00111’ 700 MHz

Bytes 10-57 are used for the second through seventh registration zones in the zone list. Bytes 58-65 are used for the eighth zone in the zone list. Bytes 10-65 are coded the same as bytes 2-9.

3.4.10 CDMA System/Network Registration Indicators

This EF stores its SID, NID List on the R-UIM. This is described in [14] Sections 6.3.4 and 6.6.5.1.5, titled “Registration Memory”, and “Zone-Based Registration”, respectively.

<table>
<tr><td colspan="2">Identifier: ‘6F2A”</td><td colspan="1">Structure: transparent</td><td>Mandatory</td></tr>
<tr><td colspan="2">File size: 6N + 1 bytes</td><td colspan="2">Update Activity: high</td></tr>
<tr><td colspan="4">Access Conditions:
READ CHV
UPDATE CHV
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td>Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td>N, Size of SID/NID List</td><td>M</td><td>1 byte</td></tr>
<tr><td>2-3</td><td>SID, first entry</td><td>M</td><td>2 bytes</td></tr>
<tr><td>4-5</td><td>NID, first entry</td><td>M</td><td>2 bytes</td></tr>
<tr><td>6</td><td>Frequency Block, first entry</td><td>M</td><td>1 byte</td></tr>
<tr><td>7</td><td>Band Class/ZONE_TIMER, first entry</td><td>M</td><td>1 byte</td></tr>
<tr><td></td><td>………</td><td></td><td></td></tr>
<tr><td>6N-4, 6N-3</td><td>SID</td><td>M</td><td>2 bytes</td></tr>
<tr><td>6N-2, 6N-1</td><td>NID</td><td>M</td><td>2 bytes</td></tr>
<tr><td>6N</td><td>Frequency Block</td><td>M</td><td>1 byte</td></tr>
<tr><td>6N + 1</td><td>Band Class/ZONE_TIMER</td><td>M</td><td>1 byte</td></tr>
</table>

Byte 2:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                        |<-------LSB of SID
  |<--------------------------------------------->|<---------------SID bits in ascending order
```

Byte 3:

b8	b7	b6	b5	b4	b3	b2	b1

```
                  |<-------------------------------------->|<----------SID bits in ascending order
         |<---------------------------------------------------MSB of SID
  |<------------------------------------------------------------RFU
```

Byte 4:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                        |<-------LSB of NID
  |<--------------------------------------------->|<---------------NID bits in ascending order
```

Byte 5:

b8	b7	b6	b5	b4	b3	b2	b1

```
         |<---------------------------------------------->|<---------NID bits in ascending order
  |<------------------------------------------------------------MSB of NID
```

Byte 6:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                         |<------------->|<----------------Frequency Block:   '000' Block A
|<------------------------------->|<------------------------------------------RFU              '001' Block B
                                                                                              '010' Block C
                                                                                              '011' Block D
                                                                                              '100' Block E
                                                                                              '101' Block F
                                                                                              All others RFU
```

Note: Frequency Block entry is ignored if Band Class is ‘00000’.

Byte 7:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                          |<------------->|<------ZONE_TIMER
|<------------------------------->|<-------------------------------Band Class
                                                             '00000' 800 MHz
                                                             '00001' PCS
                                                                 :              as defined in [2]
                                                             '00111' 700 MHz
```

Bytes 8 to 6N+1 are coded the same as bytes 2-7.

3.4.11 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 CHV UPDATE CHV 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

The parameters for Distance-Based Registration are described in [14], Section 6.6.5.1.4.

Byte 1:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                          |<---------LSB of BASE_LAT_REG
 |<--------------------------------------->|<-------------BASE_LAT_REG bits in ascending order
```

Byte 2:

b8	b7	b6	b5	b4	b3	b2	b1

```
 |<------------------------------------------------->|<----BASE_LAT_REG bits in ascending order
```

Byte 3:

b8	b7	b6	b5	b4	b3	b2	b1

```
                       |<--------------------------->|<---BASE_LAT_REG bits in ascending order
                 |<------------------------------------------MSB of BASE_LAT_REG
 |<----->|<---------------------------------------------------RFU
```

Byte 4:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                          |<---------LSB of BASE_LONG_REG
 |<--------------------------------------->|<--------BASE_LONG_REG bits in ascending order
```

Byte 5:

b8	b7	b6	b5	b4	b3	b2	b1

```
|<------------------------------------------------------->|<--BASE_LONG_REG bits in ascending order
```

Byte 6:

b8	b7	b6	b5	b4	b3	b2	b1

```
                        |<----------------------------->|<----BASE_LONG_REG bits in ascending order
                  |<-------------------------------------------MSB of BASE_LONG_REG
|<----->|<----------------------------------------------------RFU
```

Byte 7:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                  |<-------LSB of REG_DIST_REG
|<------------------------------------------------>|<------------REG_DIST_REG bits in ascending order
```

Byte 8:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                          |<----->|<----REG_DIST_REG bits in ascending order
                                   |<-------------------MSB of REG_DIST_REG
|<------------------------------>|<-----------------------------RFU
```

3.4.12 Access Overload Class ($ACCOLC_p$)

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 ACCOLC is the set of the last 4 digits of the IMSI_M. [5]

Identifier: ‘6F2C’	Structure: transparent		Mandatory
File size: 1 byte	Update Activity: low		
Access Conditions: READ — CHV UPDATE — ADM INVALIDATE — ADM REHABILITATE — ADM			
Bytes	Description	M/O	Length
1	Access Overload Class ($ACCOLC_p$)	M	1 byte

Byte 1:

```
| b8 | b7 | b6 | b5 | b4 | b3 | b2 | B1 |
                                    |<------LSB of ACCOLCp
                         |<----->|<--------------middle bits of ACCOLCp
                    |<------------------------------MSB of ACCOLCp
  |<-------------------->|<----------------------------------------RFU
```

3.4.13 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 CHV UPDATE CHV INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Analog/Digital/Call Termination preferences	M	1 byte

Byte 1:

b8	b7	b6	b5	b4	b3	b2	b1

b1--------MOB_TERM_FOR_NID$_p$

'0': Disallow mobile-terminated call while a NID roamer

'1': Allow mobile-terminated call while a NID roamer

b2----------------MOB_TERM_FOR_SID$_p$

'0': Disallow mobile-terminated call while a SID roamer

'1': Allow mobile-terminated call while a SID roamer

b3-----------------------MOB_TERM_HOME$_p$

'0': Disallow mobile-terminated call while using home (SID, NID) pair

'1': Allow mobile-terminated call while using home (SID, NID) pair

|<----------------------------->|-----------RFU

3.4.14 Suggested Slot Cycle Index

This EF suggests a value for the mobile station’s preferred slot cycle index for CDMA operation (see 6.3.11 of [14]).

Identifier: ‘6F2E’		Structure: transparent		Optional
File size: 1 byte		Update Activity: low		
Access Conditions: READ — CHV UPDATE — CHV INVALIDATE — ADM REHABILITATE — ADM				
Bytes	Description		M/O	Length
1	Suggested slot cycle index		M	1 byte

Byte 1:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                    |<----------LSB of suggested slot cycle index
                                              |<-----------------middle bit of suggested slot cycle index
                                        |<------------------------MSB of suggested slot cycle index
  |<------------------------------>|---------------------------------RFU
```

3.4.15 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		Mandatory
File size: 7 bytes		Update Activity: low	
Access Conditions: READ CHV UPDATE CHV 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, Sys. A	M	2 bytes
5-6	Analog First Dedicated Control Channel, Sys. B	M	2 bytes
7	Number of Dedicated Control Channels to Scan	M	1 byte

Each Channel is represented by an 11-bit binary number.

Bytes 1, 3, 5:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                   |<---LSB of channel number
 |<------------------------------------------->|<-------------channel number bits, in ascending order
```

Bytes 2, 4, 6:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                              |<--->|<-----channel number bits, in ascending order
                                        |<------------------MSB of channel number
 |<---------------------------->|<----------------------------RFU
```

3.4.16 Preferred Roaming List

This EF stores the Preferred Roaming List, as described in Section 3.5.3 of [7], "Over-the-Air Service Provisioning of Mobile Stations in Spread Spectrum Systems." The Preferred Roaming List includes selection parameters from [14], Annex F.

Identifier: '6F30'	Structure: transparent	Mandatory
File size: 'PR_LIST_SIZE'+ 4	Update Activity: low	

Access Conditions:

READ	CHV
UPDATE	CHV
INVALIDATE	CHV
REHABILITATE	CHV

Bytes	Description	M/O	Length
1-2	PR_LIST_MAX_SIZE	M	2 bytes
3-4	PR_LIST_SIZE	M	2 bytes
5-6	PR_LIST_ID	M	2 bytes
7	PREF_ONLY	M	1 byte
8	DEF_ROAM_IND	M	1 byte
9-10	NUM_SYS_RECS, N	M	2 bytes
11-12	NUM_ACQ_RECS, M	M	2 bytes
13-14	PR_LIST_CRC	M	2 bytes
15-16	SYS_TABLE entry1: SID	M	2 bytes
17	SYS_TABLE entry1: attributes (NID_INCL, GEO, PRI, PREF_NEG)	M	1 byte
18-19	SYS_TABLE entry1: ACQ_INDEX	M	2 bytes
20-21	SYS_TABLE entry1: NID (if included)	M	2 bytes
22	SYS_TABLE entry1: ROAM_IND (if included)	M	1 byte
:	....		
:	SYS_TABLE entry(n): SID	M	2 bytes
:	SYS_TABLE entry(n): attributes (NID_INCL, GEO, PRI, PREF_NEG)	M	1 byte
:	SYS_TABLE entry(n): ACQ_INDEX	M	2 bytes
:	SYS_TABLE entry(n): NID (if included)	M	2 bytes
8N + 14	SYS_TABLE entry(n): ROAM_IND (if included)	M	1 byte
8N + 15	ACQ_TABLE entry1:	M	variable
	Refer to text below. Storage requirement depends on TYPE.		
	There are 6 TYPES. TYPES 1, 2, and 4 require 1 byte for		
	Storage. TYPE 5 requires between 2 and 5 bytes.		
	TYPES 3 and 6 require between 4 and 66 bytes of storage.		
:			
:			
:	....		
:	ACQ_TABLE entry(m):	M	

Bytes 1 and 2: PR_LIST_MAX_SIZE

The PR_LIST_MAX_SIZE is the memory that may be allocated for the Preferred Roaming List on the R-UIM. This parameter may be passed to the ME in order that only PR lists of an appropriate size may be updated on the R-UIM. PR_LIST_MAX_SIZE is not an input to the calculation of PR_LIST_CRC.

Bytes 3 and 4: PR_LIST_SIZE

These two bytes define the length of the Preferred Roaming List as it is stored on the R-UIM. This is determined by the service provider, and must be no greater than PR_LIST_MAX_SIZE.

Bytes 5 and 6: PR_LIST_ID

Refer to [7], section 3.5.5.

Byte 7: PREF_ONLY

Refer to [7], section 3.5.5.

b8	b7	b6	b5	b4	b3	b2	b1

|<-------’0’ : non-preferred operation allowed
‘1’ : operate if PREF_NEG = ‘1’
|<--->|<------------------RFU

Byte 8: DEF_ROAM_IND

Refer to [7], section 3.5.5.

Byte 9: LSB’s of NUM_SYS_RECS

Refer to [7], section 3.5.5.

b8	b7	b6	b5	b4	b3	b2	b1

|<---------LSB of NUM_SYS_RECS
|<--->|<----------NUM_SYS_RECS bits in ascending order

Byte 10: MSB’s of NUM_SYS_RECS

Refer to [7], section 3.5.5.

b8	b7	b6	b5	b4	b3	b2	b1

|<---------------------------->|<---NUM_SYS_RECS bits in ascending order
|<---MSB of NUM_SYS_RECS
|<------>|<---RFU

Byte 11: LSB’s of NUM_ACQ_RECS

Refer to [7], section 3.5.5.

b8	b7	b6	b5	b4	b3	b2	b1

|<-------LSB NUM_ACQ_RECS
|<--->|<----------NUM_ACQ_RECS bits in ascending order

Byte 12: MSB of NUM_ACQ_RECS
Refer to [7], section 3.5.5.

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                    |<---------MSB of NUM_ACQ_RECS
 |<--------------------------------------------->|<----------------RFU
```

Bytes 13 and 14: PR_LIST_CRC
Refer to [7], sections 3.5.5 and 3.5.5.1.

Byte 15: SYS_TABLE, SID (lower of 2 bytes)

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                    |<-----LSB of SID
 |<--------------------------------------------->|<--------------SID bits in ascending order
```

Byte 16: SYS_TABLE, SID (upper of 2 bytes)

b8	b7	b6	b5	b4	b3	b2	b1

```
                   |<------------------------------------->|<---------SID bits in ascending order
          |<--------------------------------------------------MSB of SID
 |<-----------------------------------------------------------RFU
```

Byte 17: SYS_TABLE, attributes
Refer to [7], section 3.5.5.3

b8	b7	b6	b5	b4	b3	b2	b1

```
                                             |<---->|<----------NID_INCL
                                     |<-----------------------PREF_NEG
                             |<-------------------------------GEO
                     |<-------------------------------------PRI
 |<------------->|----------------------------------------------RFU
```

Byte 18: SYS_TABLE, ACQ_INDEX (lower of 2 bytes)
Refer to [7], section 3.5.5.3.

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                    |<-----LSB ACQ_INDEX
 |<--------------------------------------------->|<--------------ACQ_INDEX bits in ascending order
```

Byte 19: SYS_TABLE, ACQ_INDEX (upper of 2 bytes)
Refer to [7], section 3.5.5.3.

b8	b7	b6	b5	b4	b3	b2	b1

|<---------MSB of ACQ_INDEX

|<-->|<-----------------RFU

Byte 20: SYS_TABLE, NID (lower of 2 bytes), if included.
If NID is not included, this field shall be set to ‘00’.

b8	b7	b6	b5	b4	b3	b2	b1

|<-----LSB of NID

|<-->|<--------------SID bits in ascending order

Byte 21: SYS_TABLE, NID (upper of 2 bytes), if included.
If NID is not included, this field shall be set to ‘00’.

b8	b7	b6	b5	b4	b3	b2	b1

|<-->|<---------NID bits in ascending order

|<---MSB of NID

Byte 22: ROAM_IND
Refer to [7], section 3.5.5.3.
If ROAM_IND is not included, this field shall be set to ‘00’.

Byte 23 through byte (8N + 12) represent SYS_TABLE entries 2 through the end of the table. SYS_TABLE consists of N entries, each containing 8 bytes. The structure is as shown for bytes 15 to 22 above.

Byte 8N + 15: ACQ_TABLE entries
The ACQ_TABLE consists of M entries (M is defined in bytes 11 and 12 above); each entry may have variable length depending on the entry type. There are six types; each type is shown below. In order to show byte addressing, each ACQ_TABLE type is shown as if it were the entry having a starting address of ‘8N + 13’ i.e., the “top” of ACQ_TABLE. Type names are from [7], section 3.5.5.2. Each type is identified by a bold title.

Type 1: Cellular Analog

Byte 8N + 13: ACQ_TABLE TYPE and Preferences, for Cellular Analog
Refer to [7], section 3.5.5.2

b8	b7	b6	b5	b4	b3	b2	b1

|<--------------------->|<----TYPE field = ‘0001’,

|<---->|--A/B selection

‘0 0’: System A
‘0 1’: System B
‘1 0’: Reserved
‘1 1’: System A or B
|<---->|--Unused, set to ‘00’

Type 2: Cellular CDMA (Standard Channels)

Byte 8N + 13: ACQ_TABLE TYPE and Preferences, for Cellular CDMA (Standard Channels)
Refer to [7], section 3.5.5.2.

b8	b7	b6	b5	b4	b3	b2	b1

|<-------------------->|<----TYPE field = ‘0010’,
|<---->|--A/B selection
‘0 0’: System A
‘0 1’: System B
‘1 0’: Reserved
‘1 1’: System A or B
|<----->|---PRI_SEC selection
‘0 0’: Reserved
‘0 1’: Primary CDMA Channel
‘1 0’: Secondary CDMA Channel
‘1 1’: Primary or Secondary CDMA Channel

Type 3: Cellular CDMA (Custom Channels)

Byte 8N + 13: ACQ_TABLE TYPE, for Cellular CDMA (Custom Channels)
Refer to [7], section 3.5.5.2.

b8	b7	b6	b5	b4	b3	b2	b1

|<--------------------->|<----TYPE field = ‘0011’,
|<-------------------->|<---Unused, set to ‘0000’

Byte 8N + 14: ACQ_TABLE, number of channels, for Cellular CDMA (Custom Channels)

b8	b7	b6	b5	b4	b3	b2	b1

|<----------------------------->|<----------number of channels
|<------------->|<-----------Unused, set to ‘000’

Byte 8N + 15: ACQ_TABLE, channel (lower of 2 bytes) for Cellular CDMA (Custom Channels)
There may be up to 32 channels in this section.

b8	b7	b6	b5	b4	b3	b2	b1

|<-----LSB of channel 1
|<--->|<--------------channel 1 bits in ascending order

Byte 8N + 16: ACQ_TABLE, channel (upper of 2 bytes) for Cellular CDMA (Custom Channels)

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                    |<----->|<---------channel 1 bits in ascending order
                                           |<----------------------------MSB of channel 1
 |<----------------------------->|<-----------------------------------Unused, set to '00000'
```

Bytes 8N + 17, 8N + 18 are used to store channel 2, bytes 8N + 19, 8N + 20 are used to store channel 3, up to 8N + 77, 8N + 78 if storage for 32 channels is needed.

Type 4: Cellular CDMA Preferred

Byte 8N + 13: ACQ_TABLE TYPE and Preferences, for Cellular CDMA Preferred
Refer to [7], section 3.5.5.2

b8	b7	b6	b5	b4	b3	b2	b1

```
                                   |<-------------------->|<----TYPE field = '0100',
                  |<---->|--A/B selection
                               '0  0': System A
                               '0  1': System B
                               '1  0': Reserved
                               '1  1': System A or B
 |<---->|--Unused, set to '00'
```

Type 5: PCS CDMA (Using Blocks)

Byte 8N + 13: ACQ_TABLE TYPE and number of blocks, for PCS CDMA (Using Blocks)
Refer to [7], section 3.5.5.2.

b8	b7	b6	b5	b4	b3	b2	b1

```
                                   |<---------------------->|<----TYPE field = '0101',
           |<------------>|<---number of blocks
 |<--------Unused, set to '0'
```

Byte 8N + 14: ACQ_TABLE, block identifier for PCS CDMA (Using Blocks)
There may be up to 8 blocks, coded onto (up to) 4 identifier bytes

b8	b7	b6	b5	b4	b3	b2	b1

```
                                          |<------------->|<------block number, for block 1
                                  |<---------------------Unused, set to '0'
           |<------------>|<------block number, for block 2
 |<---------------------Unused, set to '0'
```

Bytes 8N + 15, 8N + 16, and 8N +17, if needed, are used to store blocks 3 through 8.

<u>Type 6: PCS CDMA (Using Channels)</u>

Byte 8N + 13: ACQ_TABLE TYPE, for PCS CDMA (Using Channels)
Refer to [7], section 3.5.5.2.

b8	b7	b6	b5	b4	b3	b2	b1

```
                              |<-------------------->|<----TYPE field = ‘0110’,
 |<--------------------->|<---Unused, set to ‘0000’
```

Byte 8N + 14: ACQ_TABLE, number of channels, for PCS CDMA (Using Channels)

b8	b7	b6	b5	b4	b3	b2	b1

```
                       |<----------------------------->|<----------number of channels
 |<------------->|<-----------Unused, set to ‘000’
```

Byte 8N + 15: ACQ_TABLE, channel (lower of 2 bytes) for PCS CDMA (Using Channels)
There may be up to 32 channels in this section.

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                     |<-----LSB of channel 1
|<---------------------------------------------->|<--------------channel 1 bits in ascending order
```

Byte 8N + 16: ACQ_TABLE, channel (upper of 2 bytes) for PCS CDMA (Using Channels)

b8	b7	b6	b5	b4	b3	b2	b1

```
                                           |<----->|<---------channel 1 bits in ascending order
                                    |<----------------------------MSB of channel 1
|<----------------------------->|<------------------------------------Unused, set to ‘00000’
```

Bytes 8N + 17, 8N + 18 are used to store channel 2, bytes 8N + 19, 8N + 20 are used to store channel 3, up to 8N + 77, 8N + 78 if storage for 32 channels is needed.

3.4.17 Removable UIMID

This EF stores an (up to) 56-bit electronic identified number (ID) that is unique to the R-UIM. The UIMID is meant to emulate many of the functions of the ESN. Therefore, if future standards require an increase in size of the ESN, then the size of the UIMID will increase correspondingly. The R-UIMID is unrelated to both the ICCID and to the ESN of any host equipment to which the R-UIM may be attached.

Identifier: '6F31'	Structure: transparent	Mandatory
File size: 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

3.4.18 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 mobile equipment (ME) shall not select this service.

Identifier: '6F32'	Structure: transparent	Mandatory
File size: n bytes	Update activity: low	

Access Conditions:

READ	CHV
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
etc.			
N	Services (4n-3) to (4n)	O	1 byte

Services:

Service n1 :	CHV disable function
Service n2 :	Abbreviated Dialling Numbers (ADN)
Service n3 :	Fixed Dialling Numbers (FDN)
Service n4 :	Short Message Storage (SMS)
Service n5 :	RFU
Service n6 :	RFU
Service n7 :	RFU
Service n8 :	RFU
Service n9 :	RFU
Service n10 :	Extension1
Service n11 :	Extension2
Service n12 :	SMS Parameters
Service n13 :	Last Number Dialled (LND)
Service n14 :	RFU
Service n15 :	RFU
Service n16 :	RFU
Service n17 :	Service Provider Name
Service n18 :	Service Dialling Numbers (SDN)
Service n19 :	Extension3
Service n20 :	RFU

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’.

First byte:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                 |<---->|<--------------Service n1
                                      |<---->|<-------------------------Service n2
                          |<---->|<-------------------------------------Service n3
               |<---->|<------------------------------------------------Service n4
```

Second byte:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                                 |<---->|<--------------Service n5
                                      |<---->|<-------------------------Service n6
                          |<---->|<-------------------------------------Service n7
               |<---->|<------------------------------------------------Service n8
```

etc.

3.4.19 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.

<table>
<tr><td colspan="2">Identifier: ‘6F33’</td><td>Structure: transparent</td><td>Mandatory</td></tr>
<tr><td colspan="2">File size: 3 bytes</td><td colspan="2">Update Activity: low</td></tr>
<tr><td colspan="4">Access Conditions:

READ ADM
UPDATE ADM
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td>Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1-3</td><td>Service Programming Code</td><td>M</td><td>3 bytes</td></tr>
</table>

SPC is a 6-digit number d1d2d3d4d5d6, where d1 is the most significant digit and d6 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. The BCD digits are mapped to the three bytes as follows:

byte 3 bits 1 through 4 contain the BCD coding of d6;
byte 3 bits 5 through 8 contain the BCD coding of d5;
byte 2 bits 1 through 4 contain the BCD coding of d4;
byte 2 bits 5 through 8 contain the BCD coding of d3;
byte 1 bits 1 through 4 contain the BCD coding of d2;and
byte 1 bits 5 through 8 contain the BCD coding of d1.

3.4.20 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.

<table>
<tr><td colspan="2">Identifier: ‘6F34’</td><td colspan="2">Structure: transparent</td><td>Mandatory</td></tr>
<tr><td colspan="2">File size: 1 byte</td><td colspan="3">Update Activity: low</td></tr>
<tr><td colspan="5">Access Conditions:

READ CHV
UPDATE CHV
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td colspan="2">OTAPA/SPC_Enable</td><td>M</td><td>1 byte</td></tr>
</table>

Byte 1:

b8	b7	b6	b5	b4	b3	B2	b1

```
                                                            |<------OTAPA_Enable
                                      |<------------->|<--------------RFU
                              |<----------------------------------------SPC_Change_Enable
|<-------------->|<-----------------------------------------------RFU
```

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 be to performed on the NAM. Refer to [7], Section 3.2.2.

For SPC_Change Enable, a value of ‘0’ for the R-UIM indicates that the user consents to allow the service provider to change the value of the Service Programming Code. A value of ‘1’ indicates that the user denies permission for the service provider to change the value of SPC.

3.4.21 NAM_LOCK

This EF stores the locked/unlocked state of the NAM. This EF is based upon information in [7], section 4.5.4.3.

Identifier: '6F35'	Structure: transparent		Mandatory
File size: 1 byte		Update Activity: low	
Access Conditions: READ CHV UPDATE CHV INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	SPASM protection indicator (NAM_LOCK) status	M	1 byte

Byte 1:

b8	b7	b6	b5	b4	b3	b2	b1

|<-------NAM_LOCK

|<--->|<--------------RFU

For bits 1 through 4, a value of '0' indicates that the SPASM protection mechanism has locked the NAM. A value of '1' indicates that the NAM is unlocked.

3.4.22 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 a subset of the information in [7], section 3.5.1.7.

Identifier: '6F36'	Structure: transparent		Mandatory
File size: 2N + 1 bytes		Update Activity: low	
Access Conditions: READ UPDATE INVALIDATE REHABILITATE		CHV ADM ADM ADM	
Bytes	Description	M/O	Length
1	N, number of OTASP/OTAPA features	M	1 byte
2	NAM Download (DATA_P_REV) ID	M	1 byte
3	DATA_P_REV	M	1 byte
4	Key Exchange (A_KEY_P_REV) ID	M	1 byte
5	A_KEY_P_REV	M	1 byte
6	System Selection for Preferred Roaming (SSPR_P_REV) ID	M	1 byte
7	SSPR_P_REV	M	1 byte
8	Service Programming Lock (SPL_P_REV) ID	M	1 byte
9	SPL_P_REV	M	1 byte
10	Over-The-Air Parameter Admin (OTAPA_P_REV) ID	M	1 byte
11	OTAPA_P_REV	M	1 byte
:	:	:	:
2N	Feature N	M	1 byte
2N + 1	Protocol Revision for Feature N	M	1 byte

Coding of features and protocol revisions is described in [7], section 3.5.1.7.

3.4.23 Service Preferences

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

<table>
<tr><td colspan="2">Identifier: '6F37'</td><td colspan="2">Structure: transparent</td><td>Mandatory</td></tr>
<tr><td colspan="3">File size: 1 byte</td><td colspan="2">Update Activity: low</td></tr>
<tr><td colspan="5">Access Conditions:

READ CHV
UPDATE CHV
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td colspan="2">Service Preferences (e.g. band class, analog vs. cdma)</td><td>M</td><td>1 byte</td></tr>
</table>

Byte 1:

b8	b7	b6	b5	b4	b3	b2	b1

```
                                       |<------------>|<------System A/B Preference:     '000'No Preference
                                  |<----------------------------RFU                      '001' A preferred
                                                                                         '010' B preferred
                                                                                         '011' RFU
                                                                                         '100' RFU
                                                                                         '101' A only
                                                                                         '110' B only
                                                                                         '111' RFU
        |<------------>|<----------------------Analog/cdma Preference:            '000'No Preference
  |<---------------------------------------------RFU                               '001' Analog Preferred
                                                                                   '010' cdma preferred
                                                                                   '011' RFU
                                                                                   '100' RFU
                                                                                   '101' Analog only
                                                                                   '110' cdma only
                                                                                   '111' RFU
```

3.4.24 ESN_ME

This EF stores an (up to) 56-bit Electronic Serial Number of the Mobile Equipment (ME) to which the R-UIM is attached. This number is transferred to the R-UIM when the Mobile Equipment determines that the R-UIM has been inserted.

Identifier: '6F38'	Structure: transparent	Mandatory
File size: 8 bytes	Update Activity: High	

Access Conditions:

READ	ALW
UPDATE	CHV
INVALIDATE	ADM
REHABILITATE	ADM

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

3.4.25 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	UIM Revision	M	1 byte

An R-UIM complying with this specification shall set the R-UIM Phase to ‘00000000’.

3.4.26 Preferred Languages

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.

<table>
<tr><td colspan="2">Identifier: ‘6F3A’</td><td colspan="2">Structure: transparent</td><td>Mandatory</td></tr>
<tr><td colspan="2">File size: 1-n byte</td><td colspan="3">Update Activity: low</td></tr>
<tr><td colspan="5">Access Conditions:

READ ALW
UPDATE CHV
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td colspan="2">1st language code (highest priority)</td><td>M</td><td>1 byte</td></tr>
<tr><td>2</td><td colspan="2">2nd language code</td><td>O</td><td>1 byte</td></tr>
<tr><td>:</td><td colspan="2">:</td><td>:</td><td>:</td></tr>
<tr><td>N</td><td colspan="2">Nth language code (lowest priority)</td><td>O</td><td>1 byte</td></tr>
</table>

The language code shall be set according to Table 9-2 of [10].

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
File size: variable [1]	Update Activity: high	
Access Conditions: READ CHV UPDATE CHV INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Status	M	1 byte
2	MSG_LEN	M	1 byte
3	SMS_MSG_TYPE	M	1 byte
4	PARAMETER_ID	M	1 byte
5	PARAMETER_LEN	M	1 byte
6 to PARAMETER_LEN	Parameter Data	M	PARAMETER_LEN bytes

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

- Status

  Contents:

  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:

b8	b7	b6	b5	b4	b3	b2	b1	
					X	X	0	free space
					X	X	1	used space
					0	0	1	message received by MS from network; message read
					0	1	1	message received by MS from network; message to be read
					1	0	1	MS originating message; message sent to the network
					1	1	1	MS originating message; message to be sent

- MSG_LEN
  The length of the message. 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_MSG_TYPE
  Contents: See Table 3.4-1 of [8].

- PARAMETER_ID
  Contents: See Table 3.4.3-1 of [8].

- PARAMETER_LEN
  Contents: This field shall be set to the number of octets in the SMS message parameter, not including the PARAMETER_ID and PARAMETER_LEN fields.

- Parameter Data
  Contents: See 3.4.3 of [8].

3.4.28 EF$_{SMSP}$ (Short message service parameters)

This EF contains values for Short Message Service header Parameters (SMSP), which can be used by the Mobile Equipment (ME) for user assistance in preparation of mobile originated short messages. For example, a Message Center (MC) address will often be common to many short messages sent by the subscriber.

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 a short message is to be sent from the Mobile Station (MS), the parameter in the R-UIM record, if present, shall be used when a value is not supplied by the user.

Identifier: '6F3D'	Structure: linear fixed	Optional
File size: Variable	Update Activity: high	
Access Conditions: READ CHV UPDATE CHV INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
[1] [2]	Teleservice Identifier	M	4 bytes
	Parameter Indicators	M	2 bytes
	Origination Address [3]	M	Variable[1]
	Destination Address [4]	M	Variable[1]
	Data Coding Scheme	M	1 byte
	Validity Period	M	1 byte
	Service Category	O	4 bytes
	Origination Subaddress [3]	O	Variable [1]
	Destination Subaddress [4]	O	Variable [1]
	Bearer Reply Option	O	3 bytes
	Bearer Data	O	Variable [1]

Notes:

[1] See [8].

[2] Starting and ending bytes depend on [1]

[3] For mobile-terminated messages (not present in mobile-originated messages)

[4] For mobile-originated messages (not present in mobile-terminated messages)

<u>Encoding:</u>

Storage is allocated for all of the possible SMS parameters, regardless of whether they are present or absent. Any bytes unused, due to parameters not requiring all of the bytes, or due to absent parameters, shall be set to 'FF'.

- The supported teleservices include [16] Extended Protocol Enhanced Services, Wireless Paging Teleservice, Wireless Messaging Teleservice, Voice Mail Notification, and Wireless Application Protocol. See [8] for details.

• 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 1

Allocation of bits

Bit number	Parameter indicated
1	Origination Address
2	Destination Address
3	Reserved, set to 1
4	Data Coding Scheme
5	Validity Period
6	Service Category
7	Origination Subaddress
8	Destination Subaddress

Byte 2

Allocation of bits

Bit number	Parameter indicated
1	Bearer Reply Option
2	Bearer Data
3	Reserved, set to 1
4	Reserved, set to 1
5	Reserved, set to 1
6	Reserved, set to 1
7	Reserved, set to 1
9	Reserved, set to 1

Bit value	Meaning
0	Parameter present
1	Parameter absent

• Origination Address

Contents and Coding: As defined in [8].

• Destination Address

Contents and Coding: As defined in [8].

• Data Coding Scheme

Contents and Coding: As defined in [10].

• Validity Period

Contents and Coding: As defined in [8].

• Service Category

Contents and Coding: As defined in [8].

• Origination Subaddress

Contents and Coding: As defined in [8].

• Destination Subaddress

Contents and Coding: As defined in [8].

- Bearer Reply Option

  Contents and Coding: As defined in [8].

- Bearer Data

  Contents and Coding: As defined in [8].

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 CHV UPDATE CHV 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	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.
  Contents: This flag indicates whether or not there is memory capacity available to store SMS messages.
  Coding:
  b1=1 means flag unset; memory capacity available
  b1=0 means flag set

b2 to b8 are reserved and set to 1.

3.4.30 Supplementary Services Feature Code Table

This EF stores the numeric feature code to be used by the M 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 prepending an asterisk

Identifier: '6F3F'	Structure: transparent		Optional
File size: variable		Update Activity: low	
Access Conditions:			
READ		CHV	
UPDATE		CHV	
INVALIDATE		ADM	
REHABILITATE		ADM	

Bytes	Description	M/O	Length
1	N, Number of Feature Codes	M	1 byte
2-3	User Selectable Call Forwarding with a pre-registered number (USCF)	M	2 bytes
4-5	User Selectable Call Forwarding to a number stored in the R-UIM of the MS (USCF)	M	2 bytes
6-7	User Selectable Call Forwarding to voice mail	M	2 bytes
8-9	Answer Holding (AH)	M	2 bytes
10-11	Activate Rejection of Undesired Annoying Calls (RUAC)	M	2 bytes
12-13	Deactivate Rejection of Undesired Annoying Calls (RUAC)	M	2 bytes
14-15	Advice of Charge (AOC)	M	2 bytes
16-17	Activate Call Forwarding – Busy (CFB)	M	2 bytes
18-19	De-activate Call Forwarding – Busy (CFB)	M	2 bytes
20-21	Activate Call Forwarding – Default (CFD)	M	2 bytes
22-23	De- activate Call Forwarding – Default (CFD)	M	2 bytes
24-25	Activate Call Forwarding – No Answer (CFNA)	M	2 bytes
26-27	De-activate Call Forwarding – No Answer (CFNA)	M	2 bytes
28-29	Activate Call Forwarding – Unconditional (CFU)	M	2 bytes
30-31	De-activate Call Forwarding – Unconditional (CFU)	M	2 bytes
32-33	Cancel Call Waiting, per call (CCW)	M	2 bytes
34-35	Call Trace (COT)	M	2 bytes
36-37	Calling Name Restriction (CNAR)	M	2 bytes
38-39	Calling Number Identification Restriction (CNIR)	M	2 bytes
40-41	Automatic Callback (AC)	M	2 bytes
42-43	Activate Automatic Recall (AR)	M	2 bytes
44-45	De-activate Automatic Recall (AR)	M	2 bytes
46-47	Do Not Disturb (DND)	M	2 bytes
48-49	Priority Calling (PACA)	M	2 bytes
50-51	Activate Selective Call Acceptance (SCA)	M	2 bytes

52-53	De-activate Selective Call Acceptance (SCA)	M	2 bytes
54-55	Voice Message Retrieval (VMR)	M	2 bytes
:	:	:	:
2N+1	FCN	M	2 bytes

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:

unused digits of the feature code are set to hexadecimal ‘F’;
the most significant digit is encoded in the most significant four bits of the first byte;
the next most significant digit is encoded in the least significant four bits of the first byte;
the next most significant digit is encoded in the most significant four bits of the second byte; and
the least significant digit is encoded in the least significant four bits of the second byte.

For example, if the feature code for USCF with a pre-registered number were “*789”, bytes 2-3 of the EF would be set to hexadecimal ‘F789’.

Unsupported feature entries will be encoded as hexadecimal 'FF'

3.4.31 CDMA Home Service Provider Name

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 UPDATE INVALIDATE REHABILITATE	 ALW ADM ADM 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 should be displayed when the MS is registered in the home service area.

Coding: see below

Byte One:

b8	b7	b6	b5	b4	b3	b2	b1

|< ---b1=0 display of registered system not required
b1=1 display of registered system required
|< --- > |< ------ RFU

Byte Two:

b8	b7	b6	b5	b4	b3	b2	b1

|< --------------------------- >|< --b1-b5 = Character Encoding [10].
|< ------------- >|< ---RFU

Byte Three:

b8	b7	b6	b5	b4	b3	b2	b1

|< -- >|< ----b1-b8 = Language Indicator [10].

Bytes 4 – 35:

Service Provider Name

Contents: service provider string to be displayed

Coding: the string shall use SMS conventions as defined in [10], Tables 9-1 & 9-2. The string shall be left justified. Unused bytes shall be set to 'FF'.

4 ANSI-41-Based Authentication

This section describes the interface between the ME and the R-UIM. Details of the [15] protocols are provided in order to clarify the interface. Section 4.1 describes parameter storage and flow. Section 4.2 describes the components of [15]-based security procedures 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.

4.1 Parameter Storage and Parameter Exchange Procedures

The following parameters are stored on the R-UIM:

- Algorithm(s) for Authentication and for 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 ("old" and "new") of SSD 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.
- RUIMID, a parameter that is stored in EF RUIMID having an identifier of '6F31'.
- Service Programming Code (SPC), having an identifier of '6F33.' SPC is used in the OTASP/OTAPA procedures.
- OTAPA/SPC_Enable, having an identifier of '6F34.' This stores the user's input to the OTASP/OTAPA procedures.
- NAM_LOCK, having an identifier of '6F35.' This stores the lock/unlock status of the NAM.

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.
- ME Electronic Serial Number (ESN).
- 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 UIM uses these to compose the "Auth Data" field for some ME messages. Refer to [14], Table 6.3.12.1-1, 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.
- ME Electronic Serial Number (ESN_ME), passed from the ME to the R-UIM upon insertion of the R-UIM into the 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 generation parameters P, P Length, G, G Length, A-key Protocol Revision, BS Result, 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

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 encryption algorithm(s) this may include 64 bit key and 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 generation parameters MS Result, 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 [15]-based 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 three primary 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” that is stored in the 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 [14] and other EIA/TIA air interface documents.

A subscriber’s home network is the only entity that may update the subscriber’s Shared Secret Data (SSD). This is illustrated in Figure 4.2.1-1. 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 UIM. This in turn causes the UIM to generate RANDBS. The relationship of RANDBS to RANDSeed shall be specified by the issuer of the UIM. For example, the UIM may set RANDBS equal to RANDSeed, it may derive RANDBS by applying a pseudo-random process to RANDSeed, or it may ignore RANDSeed and generate RANDBS independently. The command Get Response directs the UIM to pass RANDBS to the ME, which in turn forwards RANDBS to the network**.**

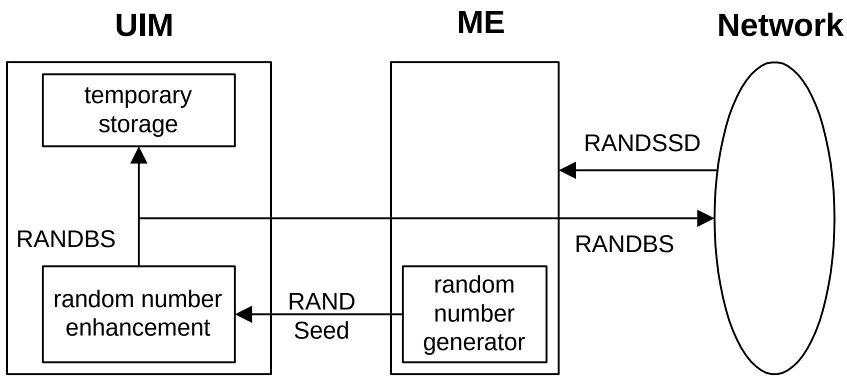

Figure 4.2.1-1 Base Station Challenge Function

Next the ME performs the Update SSD function by sending a command to the UIM, containing the parameter RANDSSD and a control data field. Refer to Figure 4.2.1-2. The UIM then calculates a new (trial) value of SSD, and also calculates an expected value of the network's response to RANDBS, called AUTHBS. The parameters ESN and IMSI that are used for these calculations are determined at the time of R-UIM insertion into the ME. For details, refer to section 4.6, "ESN Management Control", and to section 3.4.3, "EF IMSI_M".

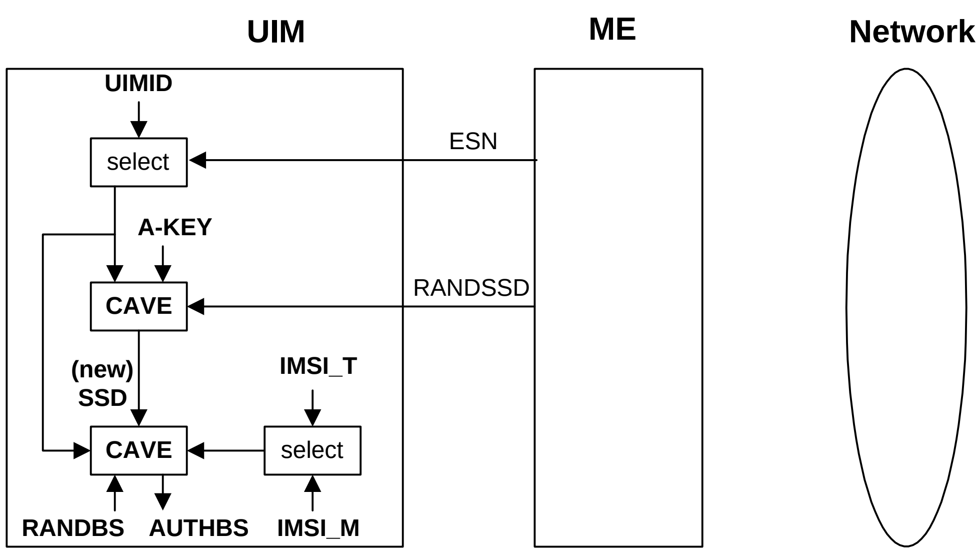

Figure 4.2.1-2 Update SSD Function, AUTHBS Calculation

At the network, the parameter RANDSSD is also used to generate a new value of SSD for the selected UIM. When RANDBS is received from the subscriber's ME, the network combines it with the new SSD to calculate AUTHBS. AUTHBS is then sent from the network to the subscriber's phone. Refer to

Figure 4.2.1-3. The ME in turn forwards the received value of AUTHBS to the UIM as a parameter of the Confirm SSD function. The UIM then compares its calculated value of AUTHBS to that sent by the network.

If the UIM finds the two values to be equivalent, the SSD Update procedure has been a success. The new value of SSD is then stored in semi-permanent memory on the UIM and used for all subsequent authentication calculations, with one exception, noted below. If the two values of AUTHBS are different, the UIM discards the new SSD and continues to retain its current value. Refer to Figure 4.2.1-3.

If the SSD Update procedure is being performed as part of an OTASP/OTAPA procedure, the ME shall set “process control” bit 2 to the value of ‘1’ as an input parameter of the “Update SSD” command. This will cause the UIM to retain the current value of SSD in semi-permanent memory but use the new value for re-authentication calculations. The UIM will set the value of SSD to the new value only upon UIM acceptance of the “Commit Request Message” from the network.

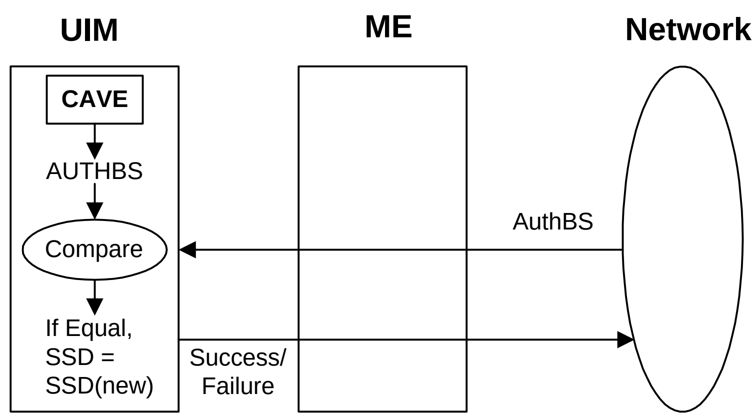

Figure4.2.1-3 Confirm SSD Function

4.2.2 Performing Authentication Calculations and Generating Encryption Keys

The second UIM security-related function is to perform authentication calculations and generate encryption keys for use with ME ciphering techniques. See Figure 4.2.2-1. This is performed by the **Run CAVE** function, having either the input parameter RAND (for a “global” challenge) or RANDU (for a “unique” challenge). Other ME-delivered parameters may include a subset of (coded) dialed digits. 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. For details, refer to section 4.6, “ESN Management Control”, and to section 3.4.3, “EF IMSI_M”.

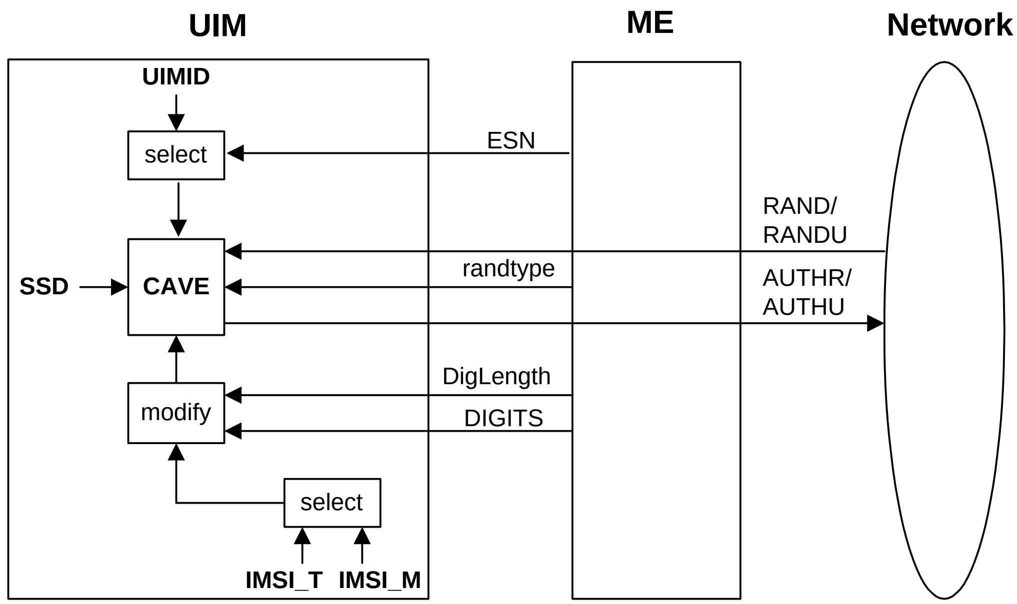

Figure 4.2.2-1 Run CAVE Function

The 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 coding of the lower 7 digits, and IMSI_M_S2 and IMSI_T_S2 to identify coding of the next 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 next 3 digits may also be used. For details, refer to 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 3.4.3, "EF IMSI_M".

In order that conformance to [11] 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 command **Get Response** causes the UIM to pass the output AUTHR or AUTHU ("global" challenge response or "unique" challenge response) to the ME. Temporary parameters may be stored on the 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 Figure 4.2.2-2. This function will produce keys for some of the ciphering mechanisms as specified in [14]. **Generate Key/VPM** will process temporary stored parameters that were produced during the calculation of an authentication response by the **Run CAVE** function. **Generate Key/VPM** 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.

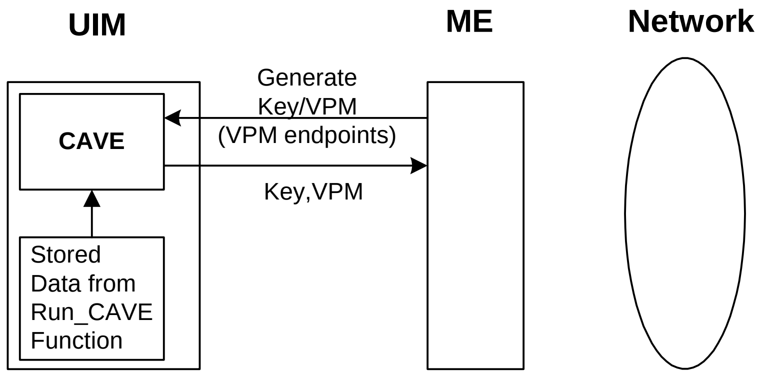

Figure 4.2.2-2 Generate Key/VPM Function

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 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 UIM has been “cloned” and take remedial action.

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

4.3 Description of [7]-based OTASP/OTAPA Functions

A complete description of Over-the-Air Service Provisioning (OTASP) and Over-the-Air Parameter Administration (OTAPA) may be found in TIA/EIA/IS-683-A. 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 “Service Programming Code” (see Section 3.4.19)

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 “OTAPA/SPC_Enable” (see Section 3.4.20)

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 allows the user to enable (or deny) over-the-air changes to be made to his SPC.

4.3.1.3 EF "NAM_LOCK" (see Section 3.4.21)

TIA/EIA/IS-683-A 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 "OTASP/OTAPA Features (see Section 3.4.22)

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, such as model number, that are stored in the ME.

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

Eleven (11) 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 the EF "OTASP/OTAPA Features" in order to format the "features" component of the response, then adds information stored in the ME in order to complete the response.

4.3.2.2 MS Key Request/Response Messages

This is the command that causes the R-UIM to generate its private and public key pair. This key pair is intended for use in a subsequent Diffie/Hellman key exchange that enables calculation of the "A-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 x power, modulo P, and temporarily stores the result as MS_RESULT. The R-UIM computes a "Result Code" and sends this in response to the MS Key Request message. The ME forwards the Result Code to the network to complete this transaction.

4.3.2.3 Key Generation Request/Response Messages

This request/response pair completes the Diffie/Hellman key exchange. The network sends BS_RESULT to the R-UIM, and the R-UIM in turn sends MS_RESULT to the network. The R-UIM calculates the Diffie/Hellman result by raising BS_RESULT to the x power, modulo P. A subset of this result is temporarily stored as the A-key. Details of this process are in [7], section 5.1.

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 UIM accepts the "Commit Request Message." This slight deviation from the [14] procedure is accommodated by the setting of "bit 2" of the "process control" parameter of the "Update SSD" command to the R-UIM.

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 command **Run CAVE** with input parameters:

- RANDTYPE=’0000 0000’ (i.e., 32 bits)
- RAND=RAND received by ME
- DigLength, DIGITS as specified by AUTH_DATA
- Process Control

  Bit0: ‘0’ (inactive)

  Bit1: ‘0’ (inactive)

  Bit2: ‘1’ (wait for Commit before storing A-key, SSD)

  Bit3: ‘0’ (inactive)

  Bit4: ‘1’ (save registers)

  Bit5: ‘0’ (inactive)

  Bit6: ‘0’ (inactive)

  Bit7: ‘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 Validate 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. The ME then accumulates the R-UIM responses and sends a composite response to the network.

[7] section 4.5.4 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/Response Messages

The ME receives the Configuration Request Message, 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. The ME accumulates the set of block responses and sends a composite response to the network.

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. 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 **Download Request** command, the R-UIM returns the Block ID and Result Code. 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. The ME acts as a message translator, and is otherwise transparent to this operation**.**

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. The R-UIM responds with the Block ID, Result Code, Segment Offset, and Segment Size, as described in [7], sections 4.5.1.9 and 3.5.1.9. The ME acts as a message translator, and is otherwise transparent t this operation.

4.3.2.11 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. 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. The input parameter "ESN" described in section 3.3.7 shall be set to the "ESN" parameter field that is to be used for air interface access messages (e.g., origination, registration, termination). The R-UIM passes RAND_OTAPA, a Result Code, and NAM_LOCK indication to the ME, which re-formats this data and sends it to the network.

4.3.2.12 Commit Request/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 Result Code, which the ME forwards to the network via the "Commit Response Message."

4.4 Description of ANSI-41-based Security-Related Commands

The commands **BASE STATION CHALLENGE, Update SSD, and Confirm SSD** are performed in sequence. If either **Update SSD** or **Confirm SSD** are run out of sequence, the card shall return '9834', SW1=98 and SW2=34.

4.4.1 Update SSD

Command	Class	INS	P1	P2	Lc	Le
UPDATE SSD	'A0'	'84'	'00'	'00'	'08'	'00'

Command parameters/data:

Octet(s)	Description	Length
1 - 7	RANDSSD	7 bytes
8	Process_Control*	1 byte

The input parameter Process_Control is coded as follows:

- The least significant bit (bit 0) is reserved for future use.
- The next-least significant bit (bit 1) is reserved for future use.
- Bit 2 of Process_Control specifies the trigger that causes newly-calculated values of SSD to become stored in semi-permanent memory.

  '000x 00xx' successful validation of AUTHBS via **Confirm SSD** command
  '000x 01xx' acceptance of a **Commit Request Message** command
  during OTASP/OTAPA

- Bit 3 of Process_Control is reserved for future use.
- Bit 4 specifies the need to save registers:

  '0001 0xxx' save registers ON
  '0000 0xxx' save registers OFF

If save registers is set (to ON) this causes the authentication process to maintain or “freeze” the state of internal registers following the generation of an authentication response.

The use of bit 4 is only relevant to the Run CAVE command, in which the generation of keys may follow the generation of an authentication response.

- Bits 5-7 of Process_Control are reserved for future use.

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’	empty

Command parameters/data:

Octet(s)	Description	Length
1 - 3	AuthBS	3 bytes

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’.

4.4.4 Run CAVE

Command	Class	INS	P1	P2	Lc	Le
RUN CAVE	‘A0’	‘88’	‘00’	‘00’	‘11’	‘03’

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	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.

Response parameters/data:

Octet(s)	Description	Length
1 - 3	AUTHR/AUTHU	3 bytes

The input parameter Process_Control is coded as follows:

- The least significant bit (bit 0) is reserved for future use.

- The next-least significant bit (bit 1) is reserved for future use.

- Bit 2 of Process_Control specifies the trigger that causes newly-calculated values of SSD to become stored in semi-permanent memory.

  ‘000x 00xx’ successful validation of AUTHBS via **Confirm SSD** command
  ‘000x 01xx’ acceptance of a **Commit Request Message** command during OTASP/OTAPA

- Bit 3 is reserved for future use and shall be set to ‘0’.

- Bit 4 specifies the need to save registers:

  '0001 0xxx' save registers ON
  '0000 0xxx' save registers OFF

If save registers is set (to ON) this causes the authentication process to maintain or "freeze" the state of internal registers following the generation of an authentication response.

The use of bit 4 is only relevant to the Run CAVE command, in which the generation of keys may follow the generation of an authentication response.

- Bit 5 is reserved for future use and shall be set to '0'.

- Bits 6 and 7 of Process_Control are reserved for future use and shall be set to '0'.

4.4.4.1 Advisory Note on the use of Run CAVE

In early versions of R-UIM specifications, the **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 **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 **Run CAVE** command should carry the input parameter Process_Control with bit 4 set to ON ('1'). Once the authentication response has been delivered via the **Get Response** command, 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 **Run CAVE** command with bit 4 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 **Run CAVE** command with the "save" function ON. One or more instances of **Run CAVE** 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 **Run CAVE** command with the "save registers" function turned ON. **Generate Key/VPM** will provide a fixed-length 64-bit key along with a key of host-specified length to the host function upon the execution of the **Get Response** command.

4.4.5 Generate Key/VPM

This command relies on the prior successful execution of the Run CAVE command with the "save" function activated. If this has not occurred, the status word SW='98' and SW='34' shall be returned upon the invocation of this command.

Command	Class	INS	P1	P2	Lc	Le
GENERATE KEY/VPM	'A0'	'8E'	'00'	'00'	'02'	'xx'

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

Response parameters/data:

Octet(s)	Description	Length
1 - 8	Key	8 bytes
9 -	VPM octets	*

* The number of VPM octets varies as specified by command parameter

4.5 Description of [7]-based OTASP/OTAPA Commands

4.5.1 MS Key Request

Command	Class	INS	P1	P2	Lc	Le
Generate Public Key	'A0'	'E0'	'00'	'00'	'6B'	'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 – 87	Parameter P	64 bytes
88 - 107	Parameter G	20 bytes

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], section 3.5.1.3, “MS Key Response Message.”

4.5.2 Key Generation Request

Command	Class	INS	P1	P2	Lc	Le
Key Generation Request	‘A0’	‘E2’	‘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], section 3.5.1.4.

4.5.3 Commit

Command	Class	INS	P1	P2	Lc	Le
Commit	‘A0’	‘CC’	‘00’	‘00’	‘00’	‘01’

Response parameters/data:

Octet(s)	Description	Length
1	Result Code	1 byte

Details of the Commit Request and Response are in [7], sections 4.5.1.6 and 3.5.1.6, respectively.

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], section 3.5.1.10.

4.5.5 Configuration Request

Command	Class	INS	P1	P2	Lc	Le
Configuration Request	‘A0’	‘E6’	‘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.

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], section 3.5.1.1.

4.5.6 Download Request

Command	Class	INS	P1	P2	Lc	Le
Download Request	‘A0’	‘E8’	‘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

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], section 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 Parameter Block), 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], section 3.5.1.8, “SSPR Configuration Response Message.”

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	Lc –2 bytes

* Note: Lc=Length of Param Data + 2.

Details of the command parameters are in [7], section 4.5.1.9, “SSPR 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

Details of the response are in [7], section 3.5.1.9, “SSPR Download Response Message.”

4.5.9 OTAPA Request

Command	Class	INS	P1	P2	Lc	Le
OTAPA Request	‘A0’	‘EE’	‘00’	‘00’	‘05’	‘06’

Command parameters/data:

Octet(s)	Description	Length
1	Start/Stop	1 byte
2 - 5	RANDSeed	4 bytes

Details of the command parameter “Start/Stop” are in [7], section 4.5.1.11, “OTAPA Request Message.”

Response parameters/data:

Octet(s)	Description	Length
1	Result Code	1 byte
2	NAM Lock Indication	1 byte
3 - 6	RAND OTAPA	4 bytes

Details of the response are in [7], section 3.5.1.11, “OTAPA Response Message.”

4.6 ESN Management Command

4.6.1 Store ESN_ME

Command	Class	INS	P1	P2	Lc	Le
Store ESN_ME	‘A0’	‘DE’	‘00’	‘00’	‘08’	‘01’

Command parameters/data:

Octet(s)	Description	Length
1	ESN_ME Length and Usage	1 byte
2-8	ESN_ME	7 bytes

The ESN_ME is stored in EF ‘6F38’. The ESN_ME length, expressed in octets, is specified by bits 0 through 3, inclusive, of Octet 1, where bit 3 is MSB and bit 0 is LSB.

Bits 4 and 5 of Octet 1 form a “Usage Indicator” and are RFU. “Usage” refers to the assignment of parameters that identify the Mobile Station and the assignment of parameters to be input to the authentication process.

Bits 6 and 7 of Octet 1 are RFU.

Response parameters/data:

Octet(s)	Description	Length
1	Change Flag, Usage Indicator Confirmation	1 byte

Bit 0 (LSB) of Octet 1 indicates whether the ESN_ME is different from the previous ESN that was stored in EF ‘6F38’. Bit 0 is set to ‘0’ if it is the same, and is set to ‘1’ if the ESN_ME has changed. This allows the ME to re-register if necessary.

Bits 1 through 3 inclusive are RFU and are set to ‘000’.

Bit 5 of Octet 1 is RFU.

Bit 4 of Octet 1 forms a “Usage Indicator.” Bit 4 determines whether the UIM_ID or the ESN from the handset is sent over the air interface to the serving network to identify the mobile-based recipient of wireless services. Bit 4 also determines whether the 32 LSBs of the UIM_ID or the 32 LSBs of the handset ESN are used as the “ESN” input to calculations performed using CAVE. If bit 4 is set to ‘0’, UIM_ID is used for both identification and for authentication calculations; i.e. UIM_ID is used instead of ESN in every place where ESN is used in [11], [14] and in [1]. If bit 4 is set to ‘1’, the handset ESN is used for both identification and for authentication calculations.

Bits 6 and 7 of Octet 1 are RFU and are set to ‘00’.

5 Additional Air Interface Procedures

5.1 Registration Procedure

5.1.1 R-UIM Insertion

Upon the insertion of a new R-UIM (i.e. bit 0 of octet 1 of the response parameters/data to the Store ESN_ME command is set to '1') into a powered-on ME when $REG\_ENABLED_S$ is equal to YES, the mobile station shall perform a power up registration regardless of the state of $POWER\_UP\_REG_S$ and $REGISTERED_S$. These parameters are described in [5], [14].

5.1.2 Procedure when ESN changes with TMSI Assigned

When the ME detects that a new R-UIM is inserted, it will use the Store ESN_ME command to inform the R-UIM of the ESN of the ME. If bit 0 of octet 1 of the response parameters/data to the Store ESN_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 $TMSI\_CODE_{S-p}$ field of the TMSI EF are not all set to '1'), the ME shall perform the following:

The ME shall store the value USE_TMSI_S in a temporary variable;

The ME shall set USE_TMSI_S to '0';

The ME shall initiate a power up registration regardless of the state of $POWER\_UP\_REG_S$ and $REGISTERED_S$; and

The ME shall 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 $TMSI\_CODE_{S-p}$ field of the TMSI EF 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 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 [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 the IMSI_M EF:

- $IMSI\_M\_CLASS_p$ 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 6.3.5 of [14].

When the IMSI_T_PROGRAMMED bit of the IMSI_T EF is set to ‘0’, the ME shall use the following values for IMSI_T in lieu of the values programmed in the IMSI_T EF:

- $IMSI\_T\_CLASS_p$ 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 Preferred Access Channel Mobile Station ID Type

When the ME receives the Preferred Access Channel Mobile Station ID Type, $PREF\_MSID\_TYPE_R$ in the overhead information (see section 6.6.2.2.5 of [14], section 2.6.2.2.5 of [5], and sections 2.6.2.2.5 and 2.6.2.2.13 of [5-A]), and $PREF\_MSID\_TYPE_R$ is set to ‘10’, the ME shall set $PREF\_MSID\_TYPE_S$ to ‘11’.