3GPP TSG-T#7
Madrid, SPAIN, 13-15 March 2000

TSGT#7(00)0023

Agenda Item: 5.2.3

Source: T2

Title: R98 Change Requests

Document for: Approval

T2 Tdoc	Spec	CR	Rev	Ph	Subject	Cat	Version-Current	Version-New	Workitem
T2-000133	03.41	A060		R98	Addition of LCS message identifier to support GPS Navigation message	B	7.2.0	7.3.0	LCS
T2-000030	03.57	A002	1	R98	Chapter 8 restructuring (Corrections MExE release 98, chapter 8)	F	7.1.0	7.2.0	MExE
T2-000048	03.57	A003		R98	Corrections to WAP chapters	F	7.1.0	7.2.0	MExE

Document T2-000133
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

03.41	CR	A060	Current Version:	7.2.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to:	TSG-T#7	for approval	X	strategic		(for SMG use only)
list expected approval meeting # here ↑		For information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG *The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc*

Proposed change affects: *(at least one should be marked with an X)* (U)SIM [ ] ME [X] UTRAN / Radio [X] Core Network [ ]

Source: T2 **Date:** 28 Jan 2000

Subject: Addition of LCS message identifier to support GPS Navigation message

Work item: Location Services (LCS)

Category: *(only one category shall be marked with an X)*

		Release:	
Correction		Phase 2	
Corresponds to a correction in an earlier release		Release 96	
Addition of feature	X	Release 97	
Functional modification of feature		Release 98	X
Editorial modification		Release 99	
		Release 00	

Reason for change: Assigns a value to LCS Message Identifier to support GPS Navigation Message Bits broadcast messaging.

Clauses affected: 4

Other specs affected:

her 3G core specifications		→ List of CRs:
her GSM core specifications		→ List of CRs:
test specifications		→ List of CRs:
S test specifications		→ List of CRs:
M specifications		→ List of CRs:

Other comments:

9.3 Message Format on BTS-MS Interface

The messages which are transmitted by the BTS for the MS include the CBS Message (information for the user) and Schedule Message (schedule of CBS messages).
The use and the formatting of the CBS messages, which contain information for the MS user, is described in this section.
The Schedule Message is broadcast to support CBS DRX mode for Mobile Stations. The Schedule Message is helpful in minimizing battery usage for Cell Broadcast in the Mobile Station, because it allows the MS to ignore transmissions of messages the customer is not interested in. The use and formatting of the Schedule Message is described in GSM 04.12.

9.3.1 General Description

Each page of a CBS Message sent to the MS by the BTS is a fixed block of 88 octets as coded in GSM 04.12. This is sent on the channel allocated as CBCH by GSM 05.02. The 88 octets of the CBS Message are formatted as described in 9.3.2.

9.3.2 Message Parameter

Octet Number(s)	**Field**
1-2	Serial Number
3-4	Message Identifier
5	Data Coding Scheme
6	Page Parameter
7-88	Content of Message

The octets in the above table are transmitted in order, starting with octet 1. The bits within these octets are numbered 0 to 7; bit 0 is the low order bit and is transmitted first.

9.3.2.1 Serial Number

This parameter is a 16-bit integer which identifies a particular message (which may be one to fifteen pages in length) from the source and type indicated by the Message Identifier and is altered every time the message with a given Message Identifier is changed.

The two octets of the Serial Number field are divided into a 2-bit Geographical Scope (GS) indicator, a 10-bit Message Code and a 4-bit Update Number as shown below:

Octet 1								Octet 2							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
GS		Message Code										Update Number			

The most significant bit of the update number is octet 2 bit 3. The most significant bit of the Message Code is octet 1 bit 5 and the least significant bit of the Message Code is octet 2 bit 4. The most significant bit of the Geographical Scope is octet 1 bit 7.

- Message Code
  The Message Code differentiates between messages from the same source and type (i.e. with the same Message Identifier). Message Codes are for allocation by PLMN operators.

The Message Code identifies different message themes. For example, let the value for the Message Identifier be "Automotive Association" (= source), "Traffic Reports" (= type). Then "Crash on A1 J5" could be one value for the message code, "Cow on A32 J4" could be another, and "Slow vehicle on M3 J3" yet another.

- Geographical Scope
  The Geographical Scope (GS) indicates the geographical area over which the Message Code is unique, and the display mode. The message is not necessarily broadcast by all cells within the geographical area. When two messages are received with identical Serial Numbers/Message Identifiers in two different cells, the Geographical Scope may be used to determine if the messages are indeed identical.

  In particular, the Geographical Scope tells the mobile if the message is only cell wide (which means that any message if received in the next cell is regarded as "new"), or PLMN wide (which means that the Message Code and/or Update Number must change in the next cell for the message to be "new"), or Location Area wide (which means that a message with the same Message Code and Update Number may or may not be "new" in the next cell according to whether the next cell is in the same Location Area as the current cell).

  The display mode indicates whether the message is supposed to be on the display all the time ("immediate") or only when the user wants to see it ("normal"). In either case, the message will be displayed only if its Message Identifier is contained within the "search list" of the mobile (see 9.3.2). These display modes are indicative of intended use, without indicating a mandatory requirement or constraining the detailed implementation by mobile manufacturers. The user may be able to select activation of these different modes.

  The coding of the Geographical Scope field is shown below:

GS Code	**Display Mode**	**Geographical Scope**
00	Immediate	Cell wide
01	Normal	PLMN wide
10	Normal	Location Area wide
11	Normal	Cell wide

Immediate = default direct display
Normal = default display under user interaction
NOTE: Code 00 is intended for use by the network operators for base station IDs.

- Update Number
  The Update Number indicates a change of the message content of the same message, i.e. the message with the same Message Identifier, Geographical Scope, and Message Code.

  In other words, the Update Number will differentiate between older and newer versions of the same message, within the indicated geographical area. A new message may have Update Number 0000; however this number will increment by 1 for each update. Any Update Number eight or less higher (modulo 16) than the last received Update Number will be considered more recent, and shall be treated as a new message, provided the mobile has not been switched off.

9.3.2.2 Message Identifier

This parameter identifies the source and type of the message. For example, "Automotive Association" (= source), " Traffic Reports" (= type) could correspond to one value. A number of messages may originate from the same source and/or be of the same type. These will be distinguished by the Serial Number. The Message Identifier is coded in binary.

The ME shall attempt to receive the CBS messages whose Message Identifiers are in the "search list". This "search list" shall contain the Message Identifiers stored in the EF_{CBMI}, EF_{CBMID} and EF_{CBMIR} files on the SIM (see GSM 11.11) and any Message Identifiers stored in the ME in a "list of CBS messages to be received". If the ME has restricted capabilities with respect to the number of Message Identifiers it can search for, the Message Identifiers stored in the SIM shall take priority over any stored in the ME.

The use/application of the Message Identifier is shown in the following list, with octet 3 of the Message Identifier shown first, followed by octet 4. Thus "1234" (hex) represents octet 3 = 0001 0010 and octet 4 = 0011 0100.

0000 - 03E7 (hex): To be allocated by PLMN operator associations. If a Message Identifier from this range is in the "search list", the ME shall attempt to receive such message.

This version of GSM 03.41 does not prohibit networks from using Message Identifiers in the range 0000 - 03E7 (hex) for Cell Broadcast Data Download to the SIM.

03E8 (hex) LCS CBS Message Identifier for E-OTD Assistance Data message

03E9 (hex) LCS CBS Message Identifier for GPS Assistance Data message

03EA (hex) LCS CBS Message Identifier for GPS Navigation Message Bits Data message

03E~~B~~A - 0FFF (hex): Intended for standardization in future versions of GSM 03.41. These values shall not be transmitted by networks that are compliant to this version of GSM 03.41. If a Message Identifier from this range is in the "search list", the ME shall attempt to receive this message.

1000 - 107F (hex): Networks shall only use Message Identifiers from this range for Cell Broadcast Data Download in "clear" (i.e. unsecured) to the SIM (see GSM 11.14). If a message Identifier from this range is in the "search list", the ME shall attempt to receive this message.

1080 – 10FF (hex): Networks shall only use Message Identifiers from this range for Cell Broadcast Data Download secured according to GSM 03.48 [15] to the SIM (see GSM 11.14). If a message Identifier from this range is in the "search list", the ME shall attempt to receive this message.

1100 - 9FFF (hex): intended for standardization in future versions of GSM 03.41. These values shall not be transmitted by networks that are compliant to this version of GSM 03.41. If a Message Identifier from this range is in the "search list", the ME shall attempt to receive this message.

A000 - AFFF (hex): PLMN operator specific range. The type of information provided by PLMN operators using these Message Identifiers is not guaranteed to be the same across different PLMNs. If a Message Identifier from this range is in the "search list", the ME shall attempt to receive this message.

B000 - FFFE (hex): intended as PLMN operator specific range in future versions of GSM 03.41. These values shall not be transmitted by networks that are compliant to this version of GSM 03.41. If a Message Identifier from this range is in the "search list", then the ME shall attempt to receive this message.

FFFF (hex): Reserved, and should not be used for new services, as this value is used on the SIM to indicate that no Message Identifier is stored in those two octets of the SIM. If this Message Identifier is in the "search list", the ME shall attempt to receive this message.

Generally, the MMI for entering these codes in the ME is left to the manufacturers' discretion. However, the 1000 lowest codes shall be capable of being specified via their decimal representation i.e.:

Octet 3	Octet 4	
0000 0000	0000 0000	(decimal '000')
0000 0000	0000 0001	(decimal '001')
0000 0000	0000 0010	(decimal '002')
0000 0000	0000 0011	(decimal '003')
	:	: :
	:	: :
0000 0011	1110 0111	(decimal '999')

9.3.2.3 Data Coding Scheme

This parameter indicates the intended handling of the message at the MS, the alphabet/coding, and the language (when applicable). This is defined in GSM 03.38 [3].

When the SIM indicates one or more language preferences, the ME shall, by default, use the language(s) stored in the SIM (in the EF_{LP} file) to set any language filter mechanisms provided by the ME.

Optionally, the user can select the language(s) required by using an MMI, to determine whether a particular message should be read and displayed.

9.3.2.4 Page Parameter

This parameter is coded as two 4-bit fields. The first field (bits 0-3) indicates the binary value of the total number of pages in the message and the second field (bits 4-7) indicates binary the page number within that sequence. The coding starts at 0001, with 0000 reserved. If a mobile receives the code 0000 in either the first field or the second field then it shall treat the message exactly the same as a message with page parameter 0001 0001 (i.e. a single page message).

9.3.2.5 Content of Message

This parameter is a copy of the 'CBS-Message-Information-Page' as sent from the CBC to the BSC.

3GPP TSG-T2 #8 / ETSI SMG4
Puerto Vallarta, MEXICO
1 - 4 Feb 2000

T2-000030

Document
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

03.57	CR	A002r1	Current Version:	7.1.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to: TSG #7	for approval	X	strategic		(for SMG use only)
list expected approval meeting # here ↑	for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG *The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc*

Proposed change affects: *(at least one should be marked with an X)* (U)SIM [ ] ME [X] UTRAN / Radio [ ] Core Network [ ]

Source: T2 **Date:** 25 Jan 2000

Subject: MExE release 98, chapter 8

Work item: MExE

Category: *(only one category shall be marked with an X)*

F	Correction	X	**Release:**	Phase 2	
A	Corresponds to a correction in an earlier release			Release 96	
B	Addition of feature			Release 97	
C	Functional modification of feature			Release 98	X
D	Editorial modification			Release 99	
				Release 00	

Reason for change: A lot of clarifications and modifications have been identified during the process of creating the release 99. Also a total restructuring of the security section (chapter 8) was felt necessary which this CR will help to reflect in the release 98 as well.

Clauses affected: Chapter 8.

Other specs affected:

Other 3G core specifications	[ ]	→ List of CRs:
Other GSM core specifications	[ ]	→ List of CRs:
MS test specifications	[ ]	→ List of CRs:
BSS test specifications	[ ]	→ List of CRs:
O&M specifications	[ ]	→ List of CRs:

Other comments: After the first restructuring of the security section, revision marked text lost its meaning in this chapter. So the revision marks found in this CR, reflects the changes made to the text after the big restructuring.


help.doc

<--------- double-click here for help and instructions on how to create a CR.

8 Security

8.1 Generic security

In order to manage the MExE and prevent attack from unfriendly sources or transferred applications unintentionally damaging the MExE device a security system is required. This section defines the MExE security architecture.

The basis of MExE security is

- a framework of permissions which defines the permissions transferred MExE executables have within the MExE MS
- the secure storage of these permissions (and permission type as defined in clause 8.3)
- conditions within the execution environment that ensure that MExE executables can only perform actions for which they have permission.

The MExE permissions framework is defined in GSM 02.57 and is as follows (there is no implied hierarchy):

- MExE Security Operator Domain (MExE executables authorised by the HPLMN operator);
- MExE Security Manufacturer Domain (MExE executables authorised by the terminal manufacturer);
- MExE Security~~Trusted~~ Third Party Domain (trusted MExE executables authorised by trusted third parties);
- Support for the three domains is mandatory.
- Untrusted MExE executables are not in a specific domain, and have very reduced privileges as described in section 8.2.

8.2 MExE executable permissions

The following table 3 specifies the permissions of ~~each~~operator, manufacturer and third party security domains in the order of restriction.

The actions listed in the security table 3 are generic actions. ~~Application~~These actions can only be performed by MExE executables via application programming interfaces (APIs) (which are intrinsically part of the MExE ~~implementation may directly or indirectly cause these actions to be performed. When these actions occur, whether directly or indirectly called by such an API, the~~implementation) .tThe security restrictions shall~~apply.~~

apply to MExE executables whether the API functionality is called directly or indirectly by the MExE executable.Explicit user permission is required for all actions by MExE executables in all domains. Types of user permission are defined in clause 8.3

~~File access is not permitted for untrusted MExE executables, except that untrusted Java applications can access files only in the application's own directory.~~ Untrusted MExE executables are not permitted access to any actions which access the phone functionality (phone functionality includes all the actions in the table 3) except for ~~this one qualified exception, and as further~~the exceptions identified in 8.2.1.

Actions available using interfaces giving access to the phone functionality (either in existence at the time of approval of this specification or not) that are not listed~~or not of the same category as a group~~ in the security table 3 ~~can be accessed by all trusted MExE executables.~~

shall be categorised into one of the groups in the security table 3 by comparing its action against the groups in order as they are listed in the table 3. If an action can be categorised into a more restrictive group
~~Table 3: Security domains and actions~~near the top of the table, then it shall not be again categorised into another, less restrictive, group further down in the table. E.g if a new action eventually results in forwarding a call, it shall be categorised into Network access. If the action is totally new, it shall be categorised into some of the groups by comparing its functionality to the group description below and by comparing with the list of actions listed in the table within the group.

1. Device core function access includes functions, which are an essential part of the phone functionality .

2. SIM smart card low level access includes functions, which allow communications at the transport service access point (send and receive application protocol data unit).

3. Network security access includes all functionalities which relate to CHV, CHV2, UNBLOCK CHV and UNBLOCK CHV2 (verification, management, reading or modifying), GSM authentication, GSM ciphering.

4. Network property access includes functions, which enable the management of operator-related data parameters and network settings.

5. Network services access includes all functionalities which result in or need interaction via the operator´s network.

6. User private data access includes all functionalities which relate to management, reading or modifying of data that the user has stored in the MS including user preferences.

7. MExE security functions access includes all functionalitiess which, through an API relate to certificate handling in the MS, end to end encryption, signed content, hashing, access to public, private, secret keys stored in the MS or in a smart card.

8. Application access includes the functionalities which relate to launch provisioned functionality, MExE executables, external executables (SIM tool kit application,…) usage.

9. Lifecycle management includes the functionalities which are needed for installing or removing MExE executables in the MS.

10. Terminal data access includes the functions which relate to accessing terminal data, i.e. not user data.

11. Peripheral access includes the functionalities related to peripherals other than user interface peripherals usage through a high level software application interface.

12. Input output user interface access includes the functionalities related to the user interface and user notification means usage.

	MExE Security Domains		
~~**Actions**~~	~~**Operator**~~	~~**Manufacturer**~~	~~**Trusted Third Party**~~
~~**User private data access**~~ [1] ~~Read~~ ~~Write~~ ~~Get properties~~ ~~Delete~~ ~~Get Location Information~~ ~~Read stored SMS~~ ~~Delete stored SMS~~	~~Yes~~ [2]		
Actions	**Operator**	**Manufacturer**	**Third Party**
Device core function access 1. Start/stop radio 2. Turn on/off device 3. Write time and/or date 4. Activate a user profile 5. Modify a user profile	No		
SIM smart card low level access [11] 1. Send APDU **2.** Slot management (power on/off, reset, port lock…)	No		
[11] *– Access to SIM is provided using more high level API as phonebook, application launching*			
Network Security access 1. Run algorithm **2.** Verify CHV/2 or UNBLOCK CHV/2 **3.** Activate/deactivate CHV **4.** Modify CHV/2	No		
Network property access 1. Get IMSI 2. Get home network 3. Select network	Yes	No	
~~**Network access**~~ ~~Initiate a voice/data connection~~ [3] ~~Accept a voice/data connection~~ [3] ~~Call forward~~ [4] ~~Multiparty call~~ [4] ~~Call deflection~~ [4] ~~Explicit call transfer~~ [4] ~~Terminate an existing connection~~ ~~Hold an existing connection~~ ~~Resume an existing connection~~ ~~Send point-point message (e.g. SMS, USSD)~~ [4] ~~Generate DTMF~~ ~~Query network status~~ ~~Get signal level~~ ~~Get call list~~	~~Yes~~		~~Yes~~ [6]

~~Actions~~	MExE Security Domains		
	~~Operator~~	~~Manufacturer~~	~~Trusted Third Party~~
Network services access 1. Initiate a voice/data connection [3] 2. Accept a voice/data connection [3] 3. Call forward [4] 4. Multiparty call [4] 5. Call deflection [4] 6. Explicit call transfer [4] 7. Terminate an existing connection 8. Hold an existing connection 9. Resume an existing connection 10. Send point-point message (e.g. SMS, USSD) [4] 11. Generate DTMF 12. Query network status 13. Get signal level 14. Get call list 15. QoS management	Yes		Yes [6]
~~**Lifecycle management**~~ ~~Install a MExE Executable~~ ~~Uninstall a MExE executable~~	~~Yes~~		
~~**Miscellaneous functions**~~ ~~Get manufacturer software version~~	~~Yes~~		
~~**Certificate management**~~ ~~Install a certificate for a given domain~~ ~~Uninstall a certificate for a given domain~~ ~~Replace a certificate for a given domain~~	~~Yes~~ [5]		
~~**Audio access**~~ ~~Sound generation to speaker (e.g. via stream)~~ ~~Select Melody~~ ~~Set speaker volume~~ ~~Get melody list~~	~~Yes~~		
~~**Power management**~~ ~~Monitor the power state~~ ~~Change the power state~~	~~Yes~~		
~~**Network selection/operator data access**~~ ~~Get IMSI~~ ~~Get home network~~ ~~Select network~~	~~Yes~~	~~No~~	

Table 3 (concluded): Security domains and actions

Actions	MExE Security Domains		
	Operator	Manufacturer	Trusted Third Party
Device core function access Start/stop radio Turn on/off device	No		
User profile management Activate a user profile Modify a user profile	No		
User preference management Modify user preferences	Yes[7]		

Legend for above footnotes:

[1] – User private data includes user files, phonebook, etc located on the MS.
[2] – The user shall be able to specify data access permissions within the capabilities of the device.
[3] – A network connection may be via any supported bearer service
[4] – Multiparty, deflection, and explicit call transfer shall be permitted only to numbers explicitly supplied by the user to the MExE Executable. Modification of call forward numbers stored in the network shall only be permitted to numbers explicitly supplied by the user to the operator.
[5] – Only the organisation whose public key is certified (or the organisation that certified the public key) can add, delete or replace a particular certificate.
[6] – The Trusted Third Party domain's permission to access the networking action depends on the provisioning mechanism as described in section 8.15
[7] – Trusted applications only have permission to modify user preferences, and not to activate or de-activate them. The user shall be able to specify for each domain, the preferences that applications in that domain can access. All other preferences shall not be accessible to that domain. The default shall be that there is no access.

[3] – A network connection may be via any supported bearer service
[4] – Multiparty, deflection, and explicit call transfer shall be permitted only to numbers explicitly supplied by the user to the MExE Executable. Modification of call forward numbers stored in the network shall only be permitted to numbers explicitly supplied by the user to the operator.
[6] – The Third Party domain's permission to access the networking action depends on the provisioning mechanism as described in section 8.15

User private data access[1]	
1. Read	Yes[2]
2. Write	Yes[2]
3. Get properties	Yes[2]
4. Delete	Yes[2]
5. Get Location Information	Yes[2]
6. Read stored SMS	Yes[2]
7. Delete stored SMS	Yes[2]
8. Modify user preferences	Yes[7]

[1] – User private data includes user files, phonebook, etc located on the MS.
[2] – The user shall be able to specify data access permissions within the capabilities of the device. It is not applied to user preferences
[7] – Trusted applications only have permission to modify user preferences, and not to activate or de-activate them. The user shall be able to specify for each domain, the preferences that applications in that domain can access. All other preferences shall not be accessible to that domain. The default shall be that there is no access. Single action user permission is the only type of user permission that shall be possible for changes to User Preferences.

~~Actions~~	~~MExE Security Domains~~		
	~~Operator~~	~~Manufacturer~~	~~Trusted Third Party~~
MExE security functions access			
1. Install a certificate for a given domain		Yes [5]	
2. Uninstall a certificate for a given domain		Yes [5]	
3. Replace a certificate for a given domain		Yes [5]	
4. Data encryption API		Yes	
5. Verify a signature API		Yes	
6. Compute a digital signature API		Yes	
7. Hash a content API		Yes	
8. Non repudiation API		Yes	
[5] – *Only the organisation whose public key is certified (or the organisation that certified the public key) can add, delete or replace a particular certificate.*			
Application access			
1. Get application list		Yes[8]	
2. Launch an application		Yes[8]	
3. Get application status		Yes[8]	
4. Stop, suspend, resume an application		Yes[9]	
[8] – *Device* provisioned functionality *access is limited to manufacturer domain. SIM tool kit application access is limited to operator domain. MExE executable access is limited to MExE executable issued by the same issuer (identify by the certificate) of launched MExE executable* [9] – *Access is limited to MExE executable which launch the application. But the end user, shall have a way to stop the launched application, MExE environment may stop the launched application or launched application may stop itself.*			
Lifecycle management		Yes	
1. Install a MExE Executable			
2. Uninstall a MExE executable			
Terminal data access			
1. Get manufacturer software version		Yes	
2. Read time and date		Yes	
Peripheral access		Yes	
1. Sound generation to speaker (e.g. via stream)			
2. Set speaker volume			
3. printer access			
4. Monitor the power state			
5. Change the power state			
6. Activate/ access Serial port (RS232, Irda, Bluetooth, USB …) access			
7. Activate/access Parallel port			
8. Activate/access Smart card other than SIM card (Send APDU, Slot management)			
Input output User interface access			
1. Input device (keyboard, mouse …)		Yes[10]	
2. Output device (display)		Yes[10]	
3. Output notification device(smart icon, sound, light, vibrator …)		Yes	
[10] – *Access request no user permission.*			

Table 3: Security domains and actions

The lists in the groups in the table 3 are not exhaustive, and other actions which are of the same category shall be included in the group for the purposes of requesting user permission.

8.2.1 ~~MExE executable p~~Permissions for untrusted ~~applications~~MExE executables

Clause 8.2 identifies the permissions for MExE executables in the 3 domains (operator, MS manufacturer and~~Trusted~~ Third Party). The permissions do not apply to untrusted ~~applications~~MExE executables which are not permitted to execute within the domains.

In order to facilitate untrusted ~~applications~~MExE executables having some limited access to MExE MS functionality beyond their very limited privileges, the following specific~~network~~ access permissions in Table 3 are extended to untrusted ~~applications:~~ MExE executables:-

- User interface

  An untrusted, uninstalled MExE executable (e.g. an applet) can access the user interface output (display) and input (keyboard, mouse, ..) without user permission, but the sending of user data to a server to which the MExE executables has a session connection (e.g. as part of a browser sesssion) requires user permission.

  An installed untrusted MExE executable shall only be able to access the user interface output (display) and input (keyboard, mouse, ..) with user permission. (Clearly, for the usability of untrusted MExE executables such as games, blanket user permission should be sought and given, and this is permissible.)

- File

  File access is not permitted for untrusted MExE executables, except that untrusted MExE executables can access files only in the MExE executable's own directory.

- Initiate a voice/data connection

  Untrusted ~~applications~~MExE executables shall be able to make calls under the following conditions.

  In addition to an untrusted ~~application~~MExE executable possibly displaying the number to be called to the user, the number to be called shall be presented to the user for permission by a provisioned functionality of the MExE MS and not by the ~~application~~MExE executable itself. (This facility would support, for example, "click to dial" button/links in an untrusted ~~application,~~MExE executable, and a MExE MS provisioned functionality then represents the number to the user for confirmation.)

- Generate DTMF

  Untrusted ~~applications~~MExE executables shall be able to generate DTMF tones under the following conditions.

  An untrusted ~~application~~MExE executable is only permitted to send DTMF tones in a currently active call. The request to generate DTMF tones in the currently active call, shall result in the characters which the tones represent being presented to the user for permission by a provisioned functionality of the MExE MS.

The untrusted ~~applications~~MExE executables permitted to use the above facilities shall be ~~applications~~MExE executables the user has downloaded himself, and not be ~~applications~~MExE executables that have been pushed to the user. ~~Applications/applets~~MExE executables/applets on the MExE MS due to the user having visited a particular web site are considered to be ~~applications~~MExE executables that the user had downloaded himself.

Untrusted ~~applications~~MExE executables shall not be permitted access to any other functions.

8.2.2 Separation of I/O streams

There shall be strict separation of the user interface input and output streams between different MExE executables, i.e. it shall not be possible for one MExE executable to access the user interface input or output of another MExE executable. In particular, it shall not be possible for an untrusted MExE executable to access the user interface input and output destined for or proceeding from a trusted MExE executable. (This requirement is to prevent a long lived malicious MExE executable from eavesdropping upon on interfering with the user to MExE executables communications, for instance PINs, of a trusted MExE executable).

8.3 User permission types

The term "user permission" is defined to mean that the user can give permission for a specific action in one of the ways defined in Table 4. ~~Blanket~~Support of blanket permission and single action permission ~~are~~is mandatory, ~~and~~but support of session permission is optional.

All prompts for user permission as described in table 4 must display ~~the alias name for~~a user friendly name identifying the signer of the corresponding MExE ~~executable.~~

executable, if available. The user shall be able to request to see the "subject" field of the certificate of the signer ("subject" here refers to the "subject" fields of WTLS and X.509 certificates and an equivalent field for any other format of certificate). If an application, for which user permission is being sought, is untrusted, the fact that the application is untrusted shall be visually indicated to the user whenever user permission is sought.

The user shall be prompted for user permission relating to ~~each~~all action groups listed in the table 3 that are required by the MExE executable. If a prompt for permission relates to more than one action, e.g. networking and user data, then it shall list the individual action group permissions which will be granted, though the action group permissions can all be granted with a single user action. This condition applies to any prompts relating to user permissions in table 4.

Note that blanket permission ~~and session permission can't~~cannot be used for uninstalled MExE executables e.g. applets, WMLS.

Table 4: User Permissions

	User Permissions		
Permission Type	**Description**	**Invocation**	**Revocation**
blanket permission	The user gives blanket permission to the MExE executable for the specified action, and the MExE executable subsequently uses the user's original permission for the identified subsequent actions whenever the MExE executable is running.	Typically such permission would be given at MExE executable configuration or run time.	The blanket permission maybe revoked by the user at any time. The user permission no longer applies once the MExE executable has been removed.
session permission	The user gives permission to the MExE executable for the specified action during a specific run time session of an MExE executable, and the MExE executable subsequently uses the user's permission for the identified subsequent actions whilst the MExE executable session is still running.	Typically such permission would be given at MExE executable run time.	The session permission maybe revoked by the user at any time. The user permission no longer applies once the MExE executable run time session has terminated.
single action permission	The user gives a single permission to the MExE executable for the specified action; if the MExE executable subsequently wishes to repeat the action it must again request the user's permission for the identified subsequent action.	Typically such permission would be given at MExE executable run time.	The user permission no longer applies once the action has terminated.

8.4 Certification and authorisation architecture

In order to enforce the MExE security framework a MExE capable MS is required to operate an authentication mechanism for verifying downloaded MExE executables. A successful authentication will result in the MExE executable being trusted; and able to be executed in a security domain (as determined by the root public key of its certification tree).

As the MExE MS may want to authenticate content from many sources, a public key based solution is mandatory. Before trusting MExE executables, the MExE MS will therefore check that the MExE executable was signed with a

private key, for which the MExE MS has the corresponding public key. The corresponding public key held in the MS must either be a root public key (securely installed in the MS, e.g. at manufacture) or a signed public key provided in a certificate. The MExE MS must be able to verify ~~this certificate,~~certificates, i.e. have the public key (as a root key or in a certificate) corresponding to the private key used to sign the certificate. ~~A~~Support of certificate chains is therefore mandatory.

The requirements on authorisation and certification are given in clause ~~8.8.1.~~8.4.1. An example authorisation and certification process is described in clause ~~8.8.2.~~8.4.2.

8.4.1 Certification requirements

A MExE MS cannot verify certified MExE executables of a particular domain unless it has a root public key for that particular domain.

Root public keys shall be securely installed in the MExE MS, say, at manufacture.

It is recommended that a "disaster recovery" root public key be securely installed on the terminal, to be used to install new root public keys when all other root public keys on the terminal are invalid.

~~Trusted~~ Third Party Domain root public keys will typically be installed along with and integrated into the MExE ME browser, as is done for PC-based browsers.

A MExE executable can only be verified if the MExE MS contains a valid root or certified public keys corresponding to the private key used to sign the MExE executable.

A MExE MS shall support at least one level of certificate under operator, manufacturer or ~~TTP~~Third Party root public keys. The MExE MS shall support at least one level of certificate chain analysis in a signed content package, as shown in figure 5.

A certificate (other than one containing a root public key) shall only be considered valid if ~~it~~the signature on the certificate is verified by a valid ~~root public key or a valid certificate~~public key (root or contained in a certificate) already present on the MS and if the certificate being verified has not expired.

~~No public key shall~~Public keys shall not be shared between~~several~~ domains.

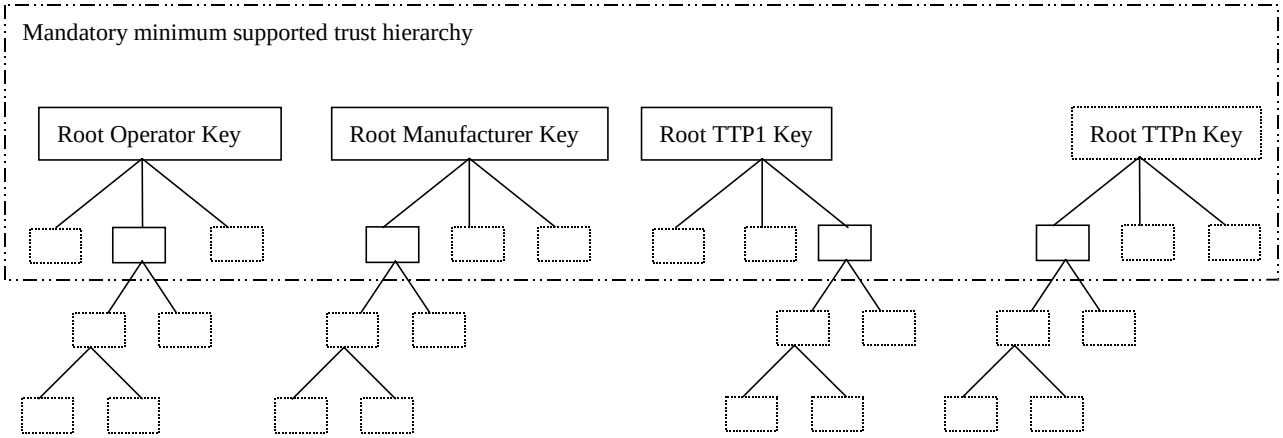

Figure 5: Trust hierarchy

The boxes below the root keys represent individual public key certificates. The solid boxes represent the minimum MExE ~~requirements (e.g. support of at least one public key certificate at the first level)~~, and the dotted boxes represent possible further support for public key certificates (either at the first or subsequent levels).

8.4.2 Example certification process

The following processes might be followed in order to securely download a~~Trusted~~ Third Party application to a MExE MS.

Root public keys for a number of Certification Authorities (CAs) are installed in the ME, along with the ME browser, at manufacture. These root public keys can be used to verify certificates for~~Trusted~~ Third Party MExE executables.

1. A third party software developer generates a private and public key pair (or obtains such a pair from a CA).
2. The third party software developer obtains a certificate for the public key from a CA. The certificate contains the developer public key, signed with the private key of the CA.
3. The 3rd party software developer adds all the certificates required in the key chain in the JAR.
4. The MExE MS downloads a MExE executable of the third party software developer.
5. The MExE MS verifies the certificate using the root public key, contained in the browser, of the relevant CA, and extracts the third party software developer public key and may store it in the certificate store for future use.
6. The MExE MS verifies that the MExE executable was signed using the private key corresponding to the third party software developer public key and installs or rejects the MExE executable accordingly.

8.5 Root Public keys

8.5.1 Operator root public key~~certificate~~

The ~~operator root certificate contains the operator root public key. The MS~~ME shall support secure storage ~~(in ME or SIM) for~~for at least one certificate containing an operator root public key~~.~~. The certificate contains a root public key generated either ~~from~~by the operator, or ~~a root public key from~~by a CA~~implicitly~~ trusted by the operator. The ME ~~will~~shall get the operator root public key from the secure area every time it needs to verify a signature, rather than cache the root public key for use in subsequent verifications.

If the MS does not contain a valid operator root public key, then the certificate chain to MExE executable previously executing in the Operator Domain will be invalid, and they will be excluded from the operator domain.

The user ~~is not authorised~~shall not be able to add or delete any type of operator public key ~~or certificate.~~(root or contained in a certificate).

Optionally, the operator ~~certificates may have~~may install a corresponding disaster-recovery root public key stored in the MS, enabling the operator to use a secure mechanism (involving the disaster-recovery key) to replace the ~~certificate.~~

certificate containing the standard operator root public key. It shall not be possible to use the disaster recovery operator root public key to replace the standard operator root public key unless both public keys are from the same operator.

There shall be no more than one valid operator root public key on the MS (excluding the disaster recovery root public key).

An application signed by ~~a operator other than the operator whose certificate is the operator certificate,~~an operator shall not be able to execute in the Operator Domain unless the root public key of that operator is installed in the MS (either ME or SIM) and is marked as trusted.

8.5.1.1 ME actions on SIM insertion and/or power up.

The requirements in this sub-clause ensure that the operator domain on the ME belongs to the same operator as the operator that issued the SIM inserted in the ME.

On power up of the terminal, the terminal shall behave as dictated by figure 6 below.

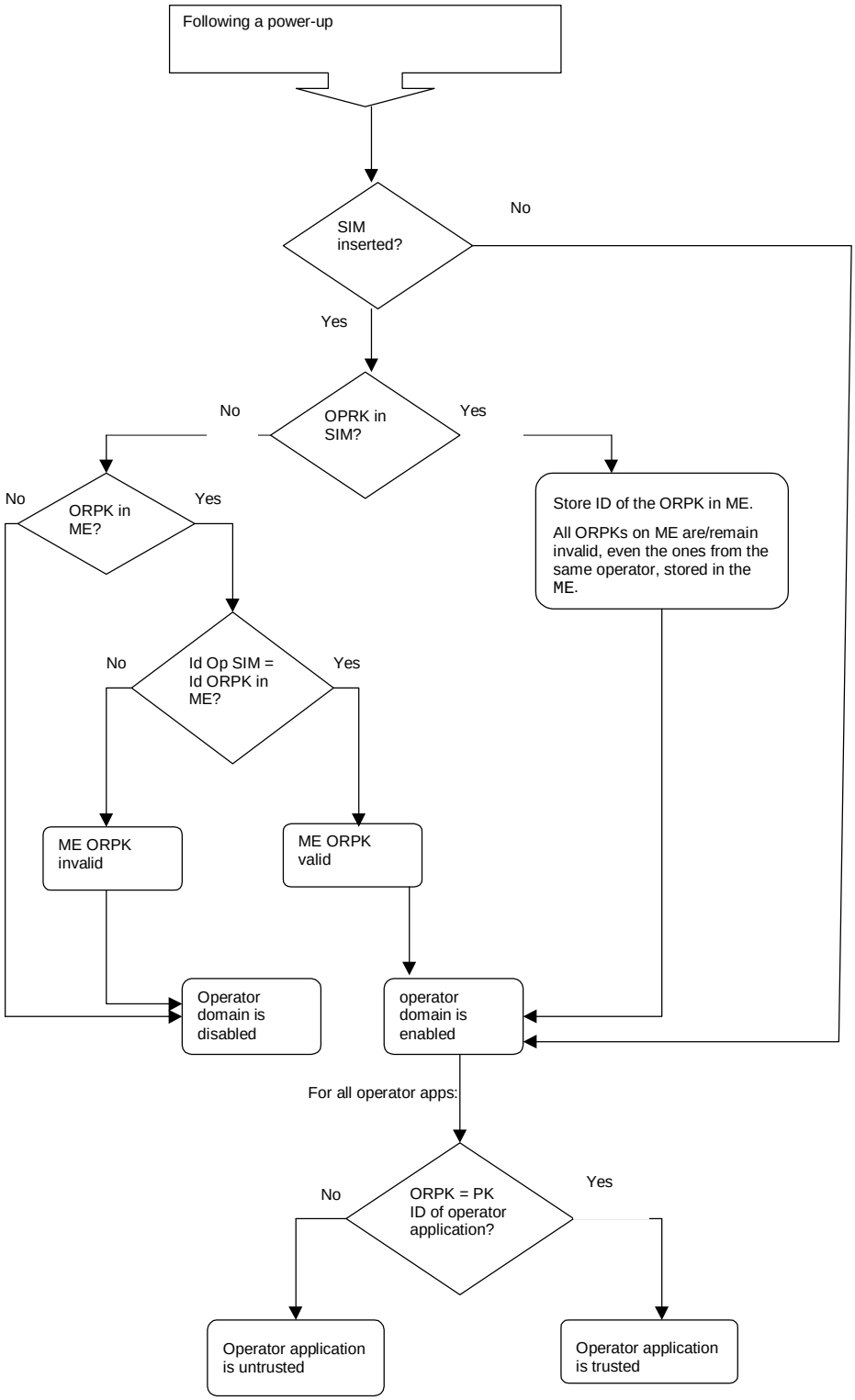

Figure 6: Terminal behaviour on power up

Editor's note: On DCS1900 the MCC+MNC is 6 digits, but elsewhere it is 5 digits. The ME needs to know how many digits to use. This problem may have been solved already. The identity of the root public key has to be defined.

When an operator root public key stored on the ME is marked as invalid, all operator applications verified using that root public key or by certificates verified by a chain that terminates with that root public key, shall cease operation as soon as possible and shall be marked as untrusted.

8.5.1.2 ME actions on removal of the SIM

Removal of the SIM shall not cause the status (i.e. valid or invalid) of any operator root public key on the terminal to change.

If a SIM is removed from the ME (without another SIM being inserted), operator applications shall continue to execute in the operator domain .

8.5.2 Manufacturer root public key~~certificate~~

~~The manufacturer root certificate contains the manufacturer root public key.~~ The ME shall support secure storage for a certificate containing a manufacturer root public key. The certificate contains a root public key ~~from~~generated by the manufacturer of the~~same~~ device, or ~~a root public key from~~by a CA~~implicitly~~ trusted by the manufacturer of the~~same~~ device.

If the ME does not contain a valid manufacturer root public key, then the certificate chain to MExE executable previously executing in the Manufacturer Domain will be invalid, and they will be excluded from the manufacturer domain.

The user ~~is not authorised~~shall not be able to add or delete any type of manufacturer public key ~~or certificate.~~(root or contained in a certificate).

The M~~a~~nufacturer shall put a root public key and optionally its corresponding~~optional~~ disaster-recovery key in the device at the time of manufacture, and use a proprietary secure mechanism (e.g. using the disaster-recovery key) to replace the certificate containing the manufacturer root public key. It shall not be possible to use the disaster recovery manufacturer root public key to replace the standard ~~certificate.~~manufacturer root public key unless both public keys are from the same manufacturer.

An application signed by a manufacturer~~other than the manufacturer whose certificate is the Manufacturer Certificate~~ shall not be able to run in the Manufacturer Domain unless the root public key of that manufacturer is installed in the MS and is marked as trusted.

There shall be no more than one valid manufacturer root public key on the MS (excluding the disaster recovery root public key).

8.5.3 T~~rusted t~~hird party root public key~~certificate~~

The ME shall support secure storage for at least one certificate containg a~~The trusted third party root certificate contains the trustedThe ME shall support secure storage for a TTP root public key.~~ third party root public key. -The ME may contain root public key (s) ~~from~~generated by CA(s) implicitly trusted by the user. The user will be able to securely install (using a secure transport) or remove~~these~~ root public key~~s~~ at any time using a system administrative tool.

The Manufacturer, Operator and Administrator may at their discretion, securely install ~~(using a secure transport) Trusted~~certificates containing Third Party root public key(s) on behalf of the user, e.g. at the time of manufacture by the Manufacturer. See section 8.~~14~~ 6 for details of Administrator control of ~~TTP~~Third Party certificate download.

If a~~Trusted~~ Third Party public key is deleted or becomes invalid, then the certificate chain to MExE executables previously executing in the~~Trusted~~ Third Party Domain certified by that public key will become "untrusted".

There may be any number of ~~TTP~~Third Party root public keys on the MS.

The third party domain administrator (user or other body) shall be able to enable and disable ~~TTP~~Third Party root public keys by using CCM. The process of adding/removing public keys and enabling/disabling public key are independent.

~~Using a provisioned functionality, the user is able to delete a TTP public key or certificate, but not the TTP root public key or certificate.~~All third party certificates shall be subject to restrictions imposed by valid certificate configuration messages.

8.5.4 Administrator root public key~~certificate~~

~~The administrator root certificate contains the administrator root public key.~~ The ME shall support secure storage for a certificate containing an administrator root public key. Only one administrator root public key shall be ~~held~~valid on the MExE MS.

The MExE MS shall support the administrator designation mechanism and the secure downloading of CCMs explained in section ~~8.12.~~8.8.

The user ~~is not authorised~~shall not be able to delete an administrator root public key or certificate.

The system shall support a mechanism (as part of a provisioned functionality and/or inherently part of the MExE implementation) allowing the owner of the MExE MS to manage the administrator root public key (including the download of a new administrator root public key) as defined in Section 8.~~9~~10.4. This mechanism shall be secure so that only the owner can use this functionality.

The administrator root public key can be downloaded to the MExE MS as described in section 8.10.4.

The same root public key may be used for both the Administrator role and the operator or manufacturer domain. This facility does not imply any increased right of the manufacturer or operator to take the Administrator role.

8.6 Certificate management

~~Four type of certificates are provided for in MExE:~~

~~• operator,~~

~~• manufacturer,~~

~~• trusted third party, and~~

~~• Administrator of trusted third party domain.~~

The manufacturer may load initial third party certificates on the device. Downloaded certificates shall be verified by an existing trusted certificate and placed in the domain defined by the root public key at the top of the verification chain for the downloaded certificate. ~~Third party root public keys shall be stored in protected memory. All third party certificates shall be subject to restrictions~~

~~imposed by valid certificate configuration messages. New third party root public keys may be downloaded as signed WAP or WWW content. The signature on the content shall be of the Administrator, Operator or Manufacturer.~~

~~The manufacturer root public key is pre-loaded in protected memory on the device at manufacture time. It is recommended that the manufacturer should include a mechanism to re-key the device due to key compromise. See clause 8.5.~~ The administrator root certificate shall be provided on the SIM if support for certificate storage on the SIM exists. For SIMs not having certificate storage the administrator root may be downloaded using the root download procedure described in section 8.10~~9~~.2.

The actions that may be performed for a given certificate are:

- addition,
- deletion,
- mark un-trusted (un-trusted certificates cannot be used to verify applications or other certificates. This process may be preferred to certificate deletion as there is a chance that the certificate may become trusted again in the near future),
- mark trusted (marking as trusted is the process of allowing an untrusted certificate to come into use again),

- modify fine grain access permissions (proposed as a future enhancement).

The ability to perform these actions depend on the certificate type being modified as well as the access level of the entity performing the operation. Users may add a third party certificate as long as it is certified by an existing trusted certificate.

Using a provisioned functionality, users may delete ~~TTP certificates, but not TTP root public keys, nor any operator or manufacturer certificate.~~Third Party certificates.

Table 4. Allowed certificate types in signed packages

Signature on Package	**Allowed Certificate types in package**
~~Administrator~~	~~TTP~~
Administrator	Third Party
~~Manufacturer~~	~~Administrator, Manufacturer, Operator, TTP~~
Manufacturer	Administrator, Manufacturer, Operator, Third Party
~~Operator~~	~~Administrator, Operator, TTP~~
Operator	Administrator, Operator, Third Party

8.7 Certificate configuration message (CCM)

The MExE device shall use the CCM to determine the third party certificates (and only the ~~TTP~~Third Party certificates) that are trusted for use on the MExE MS. The CCM shall only be used to enable or disable third party certificates and can not be used to delete certificates. The CCM may be periodically fetched or downloaded to a MExE device by the Administrator to dynamically configure the ~~trusted~~ third party list using the mechanisms defined in section 8.~~15.2~~7.1. The Certificate Configuration Message shall be as shown in Figure 8. This message is essentially a simplified version of a certificate revocation list to satisfy a particular use case. More complex usage requires a full certificate revocation list.

The MExE device may additionally support other means of enabling/disabling root certificates.

~~Version~~ ~~(1 byte)~~	~~CertificateAdvice~~ ~~(1 byte)~~
~~Listlength, the total number of bytes of fingerprints and signature~~ ~~(2 bytes)~~	
~~hashtype~~ ~~(1 byte)~~	~~hashvalue (fingerprint 1)~~ ~~(16 bytes if MD5 or 20 bytes if SHA-1)~~
~~hashtype~~ ~~(1 byte)~~	~~hashvalue (fingerprint 2)~~ ~~(16 bytes if MD5 or 20 bytes if SHA-1)~~
~~hashtype~~ ~~(1 byte)~~	~~hashvalue (fingerprint 3)~~ ~~(16 bytes if MD5 or 20 bytes if SHA-1)~~

~~hashtype~~ ~~(1 byte)~~	~~hashvalue (fingerprint 4)~~ ~~(16 bytes if MD5 or 20 bytes if SHA-1)~~

~~...~~

~~hashtype~~ ~~(1 byte)~~	~~hashvalue (fingerprint 'n')~~ ~~(16 bytes if MD5 or 20 bytes if SHA-1)~~
~~hashtype~~ ~~(1 byte)~~	~~Signature (the administrator signature, to confirm that the above TTP fingerprint list is valid)~~ ~~((listlength – (bytes used by fingerprints)) bytes)~~

~~Figure 8 Logical CCM format~~

~~CCM octet order is left to right (see figure 9), commencing with "Version" followed by "certificateAdvice", and the bit order is from most significant bit to least significant bit. The most significant byte is the leftmost byte, and the least significant byte is the rightmost byte.~~

1	2	3	4
Version	Certificate advice	MSB list length	LSB List length
5	6	7	8
Hashtype 1	Hash1 value octet 1	Hash1 value octet 2	Hash1 value octet 3
9	10	11	12
Hash1 value octet 4	Hash1 value octet 5	Hash1 value octet 6	Hash1 value octet 7
• • •			
Hash1 value last octet	Hashtype 2	Hash2 value octet 1	Hash2 value octet 2
• • •			
Hash2 value second last octet	Hash2 value last octet	Hashtype (signature)	Signature value 1
• • •			
n-3	n-2	n-1	n
Signature value m-3	Signature value m-2	Signature value m-1	Signature value m

~~Figure 9: CCM octet order~~

8.12.1 CCM Numbering convention

Bits are grouped into octets. The bits of an octet are shown horizontally and are numbered from 0 to 7. Multiple octets are shown vertically and are numbered from 0 to n.

8.12.2 CCM Order of transmission

Frames are transferred in units of octets, in ascending numerical octet order (i.e., octet 0, 1, …, n-1, n). The order of bit transmission is specific to the underlying protocols used to transport the CCM.

8.12.3 CCM Field mapping convention

When a field is contained within a single octet, the lowest bit number of the field represents the lowest-order value. When a field spans more than one octet, the order of bit values within each octet progressively decreases as the octet number increases. In that part of the field contained in a given octet the lowest bit number represents the lowest-order value.

For example, a 16 bit number can be represented as shown in Figure 7.

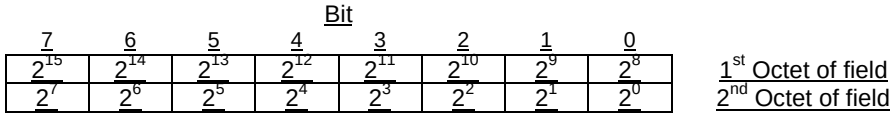

Figure 7: Field mapping convention

Bit 7	6	5	4	3	2	1	0	Octet
Version								0
Certificate Advice								1
Issued Timestamp . .								2 – 8
Expiry Timestamp . .								9 –15
SignerInfo								16
Listlength								17 –18
Fingerprint 1 Hashtype								19
Hashvalue .(Variable length k according to hash type) . .								20
Fingerprint 2 Hashtype								k+20
Hashvalue (Variable length m according to hash type) . .								
Fingerprint 3 . .								(k+20)+m+1
Fingerprint n . .								etc.
CCM Signature Hashtype								Listlength+19
Administrator signature (Variable length according to administrator signature type)								

Figure 8 Format of a CCM

version = For MExE-98 the CCM format version is 0. All other values are reserved for future use.

certificateAdvice = enumerated { enable all present and future ~~TTP~~Third Party certificates (0), disable all present and future ~~TTP~~Third Party certificates (1), enable present list only (2),enable CCM list (3), disable CCM list (4) }. All other values are reserved for future use.

Issue and Expiry Timestamps = Fields used to identify the issue and expiry date of the CCM. The issue timestamp indicates a time before the current time of day (GMT) when a CCM message must be considered invalid. The expiry timestamp (GMT) identifies the time when a CCM is to be deemed no longer valid. The receiver shall use these parameters to detect a replay attack. A MExE MS maintains information on the last valid CCM message received. A replay attack is an attacker replaying a previous valid CCM message to a MS in order to change the security settings. This is particularly dangerous for CCM messages used to enable certificates. Administrators should try and set the expiration time to be no longer than the next expected system update time of CCM information. CCM messages used to enable-all (rather than disable-all) certificates should be very short lived as the danger of these being used in a replay attack should be considered serious.

The encoding of time (GMT) shall be coded as an OCTET SEQUENCE of seven octets in length as follows:

Octet 0	1	2	3	4	5	Octet 6
Year		Month	Day	Hour	Minute	Second

Element	**Size (bits)**	**Range**

Year	16	$(0 – 65535)_{10}$
Month	8	$(1 – 12)_{10}$
Day	8	$(1 - 31)_{10}$
Hour	8	$(0- 23)_{10}$
Minute	8	$(0 – 59)_{10}$
Second *	8	$(0 – 60)_{10}$

*Note: The second field range includes the value 60 in order to accommodate leap seconds.

For example, 1st January, 2001 00:00:30 would be encoded as: 07 d1 01 01 00 00 1E.

SignerInfo = one octet indicating the type of signer information for this CCM. The only currently defined value is device-admin = 0. In this case, no further signer information follows as it is implicit. All other values are reserved for future use.

listLength = The total length of the ~~following list.~~fingerprint list not including the final CCM signature. Shall be zero when certificateAdvice = enable-all or disable-all.

hashType = enumerated { signature (0), MD5 (1), SHA-1 (2) } All other values are reserved for future use.

hashLength = The number of octets output by the selected hash type (16 for MD5 [23] and 20 for SHA-1 [24]).

The list entries shall contain certificate *fingerprints* in the form of hashes of the encoded signed certificates. The full hash output for the specified algorithm shall be used to generate the fingerprint. A list generator shall check to insure that no two list entries match when creating a list. For an X509v3 [26] or X9.68 (currently being drafted) certificate the fingerprint hash shall be computed over the ASN.1 encoded signed certificate object, first octet to last octet. For WTLS certificates the hash shall be computed over the signed WTLS certificate in network transmission format, first octet to last octet.

The signature type and length shall be indicated by the administrator certificate, which shall be present on the device. If no administrator certificate is on the device or the signature does not verify the message shall be rejected.

Upon receipt of a valid certificate configuration message the MExE device shall go through the~~trusted~~ third party certificate list, computing fingerprints if they are not stored with the certificate, enabling or disabling each certificate according to the following conditions

- certificateAdvice is enable-all all ~~TTP~~Third Party certificates shall be enabled
- certificateAdvice is disable-all all ~~TTP~~Third Party certificates shall be disabled
- certificateAdvice is enable present list only enable all ~~TTP~~Third Party certificates currently on device, do not enable any future certificates (this option allow the list to be frozen at time of manufacture) until Administrator changes
- certificateAdvice is enable-list if its fingerprint occurs in the CCM, it shall be enabled, otherwise it shall be disabled
- certificateAdvice is disable-list if its fingerprint occurs in the CCM, it shall be disabled, otherwise it shall be enabled

For future releases, the setting of fine grained permissions for each certificate is expected to be supported.

An implementation shall keep track of the domain that authorised a given application. If a CCM message is received while MExE applications are currently running the implementation shall check to ensure any applications no longer in a trusted domain have their permissions re-configured appropriately and actions that are no longer permissible are terminated.

8.7.1 Authorised CCM download mechanisms

The download of~~trusted~~ third party certificate lists by a remote administrator shall be performed by using a secure mechanism as defined below. The download mechanisms shall use HTTP over IP and/or the WAP Protocol. The URL from which the CCM is downloaded shall be in the administrator certificate if the CCM was not downloaded with the Administrator certificate. The format for storing the URL information with the certificate shall be as shown in figure ~~12:~~9:

Urltype	CharacterSet	UrlLength	URL

Figure ~~12:~~9: CCM Message URL storage format

Urltype= one byte, enumerated {WAP (0), HTTP (1)}. All other values are reserved for future use

CharacterSet = one byte, Internet Assigned Numbers Authority assigned character set.

UrlLength = one byte unsigned integer, length of the URL in octets.

The format for storing the URL information in the certificate shall be defined as part of the enhanced administrator mechanism.

When the administrator is changed, then the CCM shall also be changed. If there is URL information with the certificate as described in figure ~~12,~~9, then the new CCM shall be obtained using the URL. If the Administrator certificate was downloaded in a JAR file, the CCM shall be obtained from the same JAR file.

8.8 Provisioned mechanism for designating administrative responsibilities and adding~~trusted~~ third parties in a MExE MS

All applications in the~~TTP~~ Domain are to be signed by a key which shall be verified back to a ~~TTP~~Third Party root public key on the MExE MS. The ~~TTP~~Third Party root public keys shall be managed (e.g. addition/deletion/mark trusted/mark untrusted/change fine grained access privileges) by an administrator that is designated by the owner of the MExE MS using the MExE administrator provisioning mechanism. A mechanism is required to be provided to enable the owner of the device to dynamically assign an administrator. The mechanism shall support the following cases:

- the user is the owner
- the owner is at a remote location. In this case the owner could be the operator, a service provider or a third party.
- the owner of the MExE-SIM wants to be a temporary administrator.

8.8.1 Determining the administrator of the MExE MS.

The administrator of the MExE MS shall be determined by the logical process shown in the flowchart in figure 10. During power-up the provisioned mechanism shall look for an administrator root public key that is stored on the ME.

- Administrator root public key is absent

  if the administrator root public key is absent, then the user shall automatically become the administrator of the MExE MS

- administrator root public key is present

  if an administrator root public key is present, this root public key shall be used for all remote administration authentication, implying that the owner of the administrator root public key is the administrator

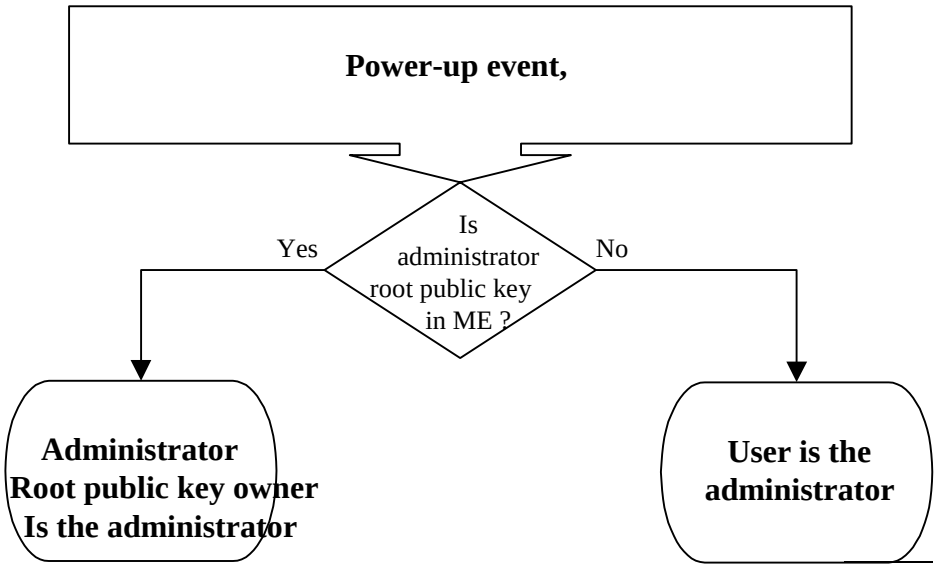

Figure 10: MExE Release 98 administrator mechanism

The rest of the mechanism is subsequently defined, however it is a future release implementation, see figure 11. This future enhanced administrator Mechanism shall be initiated after a power-up event is processed or when a MExE-SIM is detected.

(The following subclauses assume that ~~TTP~~Third Party certificates can be added using the MExE-SIM, however ~~TTP~~Third Party certificates may be added using a non-SIM approach.)

8.8.1.1 Administrator of the MExE MS is the user

~~•~~ If the administrator is the user, then a check shall be made to determine whether there is a MExE-SIM. If a MExE-SIM is present, then a check shall then be made to determine whether there is a certificate in the MExE-SIM. The enhanced administrator Mechanism shall allow the MExE MS to determine (via a format) what type of certificate is present:

- certificate present ~~-trusted~~ third party (CP-T~~T~~P)

  a certificate present in the MExE-SIM shall be considered by the ME as a~~Trusted~~ Third Party certificate, whilst that MExE-SIM is inserted in the ME. The user shall be queried to allow or disallow the certificate as a~~Trusted~~ Third Party.

- certificate present - administrator (CP-Admin)

  If a temporary certificate is present in the MExE-SIM, the user shall be queried whether to allow the certificate on the MExE-SIM to take temporary control of the~~trusted~~ third party domain. By temporary control, it is meant that once the card is removed the administrator reverts back to the user administrator settings. The above mechanism implies that the previous configuration settings for the administrator shall be saved, so that they may be restored. If the user disallows the MExE-SIM certificate, the ~~TTP~~Third Party Domain shall not be able to use any of the network capabilities in the~~trusted~~ third party domain as identified in the network access section of the security table 3

If a certificate is not present on the MExE-SIM and the administrator is the user, the user shall continue to be the administrator and may make use of all functionality.

8.8.1.2 Administrator of the MExE MS is not the user

If the administrator is not the user, then a check is made to determine if there is a MExE-SIM. If a MExE-SIM is present, then a check is made to see if there is a certificate in the MExE-SIM. If a certificate is present in the MExE-SIM, then a comparison is made of the certificate's root public key on the MExE-SIM with the root public key on the ME for the following cases:-

- Case (a): they are the same

- Case (b): they are not the same, but the ME certificate is cross-certified with the MExE-SIM certificate (a cross-certificate exists on the ME)
- Case (c): they are not the same, but the ME certificate has a line of trust back to the MExE-SIM certificate domain
- Case (d): they are not the same.

If the owner of the public key in the certificate on the MExE-SIM is to be a temporary administrator (CP-Admin), then in cases (a), (b) and (c), the temporary administrator shall be the owner of the CP-Admin root public key. In case (d), the ~~TTP~~Third Party domain shall not use any of the network capabilities in the~~trusted~~ third party domain as identified in the network access section of the security table 3. If the certificate is to be a ~~TTP,~~Third Party, then the certificate (CP-T~~T~~P) shall be verified with the CCM and based on the content and permissions of the CCM, the certificate shall be added to the ~~TTP~~Third Party list or rejected.

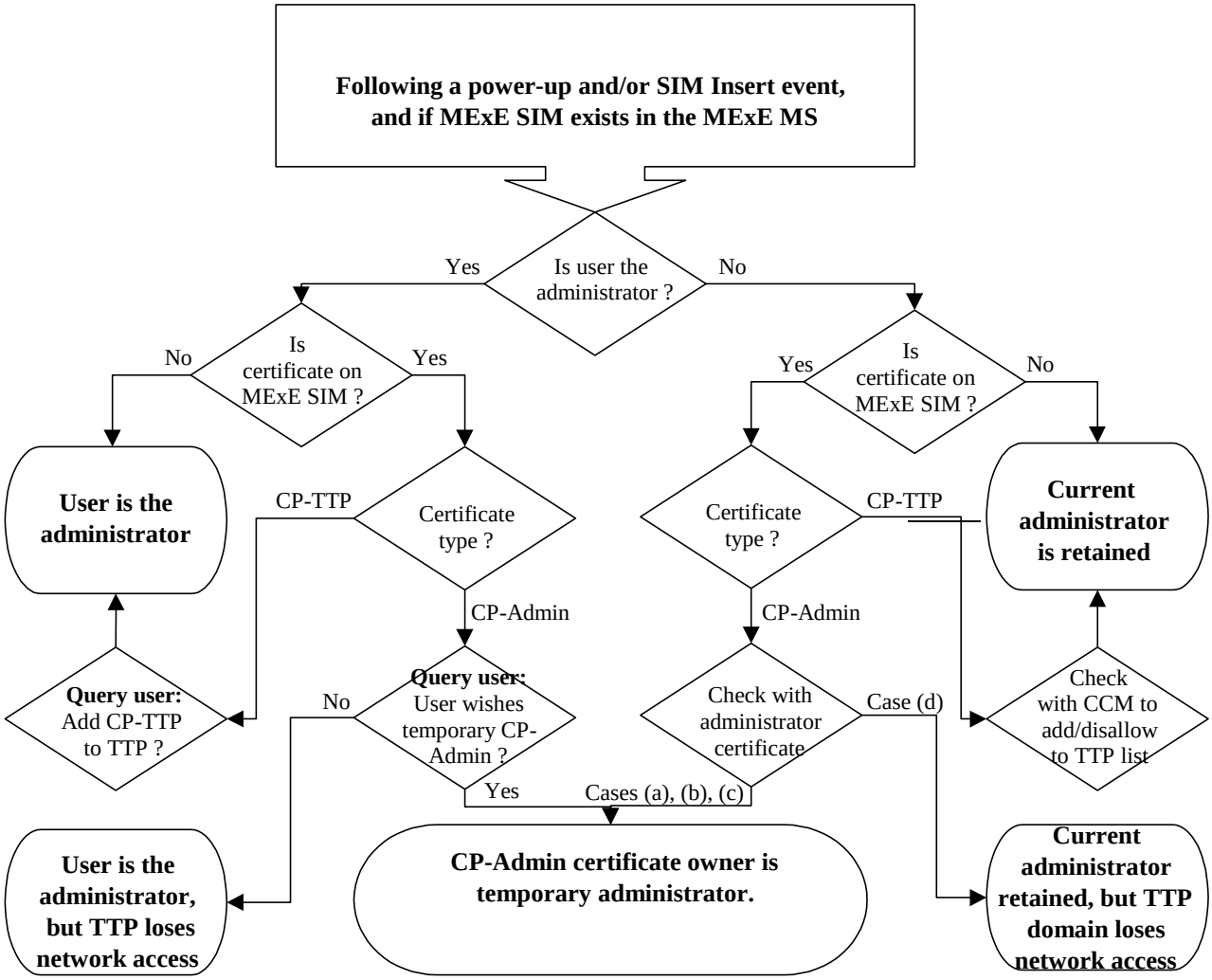

Figure 11: Enhanced administrator mechanism

8.9 Java security

There are two types of Java security [20]: sandbox, and fine grain.

The sandbox model [18] has just one domain; there is no concept of a *partly trusted* domain. The sandbox meaning of "trusted" means it is totally unrestricted to access all system resources.

Using the sandbox system, each MExE domain shall be implemented as running in a sandbox, configured with different privileges corresponding to those of the domain.

Using the fine grain Java security system [19], each MExE domain will be a set of constraints within which a Java fine grain security domain can be configured.

8.9.1 Java applet certification

Support for trusted applets is optional. Although a Java application shall be executed in a trusted domain if its certification can be validated, a Java Applet will not necessarily be executed in a trusted domain even if it does have a valid signature. It will be up to the implementers to decide if "trusted" Applets will be supported. (In certain implementations, all Applets may be always executed as "untrusted".)

8.9.2 Java application signature verification

The verification of the certification of the application or applet shall be performed as described in section 8.4~~8~~.

8.9.3 Java loading native libraries

The MExE Java VM may be able to load native libraries that are intrinsically part of the ME implementation, and MExE native libraries. The MExE Java VM shall not load other native libraries.

8.10 Signed packages used for installation

The Java Archive (JAR) file format shall be supported on classmark 2 MExE devices for securely packaging objects that are to be downloaded and installed on the ME. The method for securely packaging objects for MExE classmark 1 devices may be referenced from the WAP specifications in a future release of this specification. A MExE device may support other proprietary means of downloading and installing objects.

The JAR file shall contain a manifest file that has at least the following attribute:

`MExE-Implementation-Type`
Whose value shall be either

- "**`MExENativeLibrary`**" in the case of a MExE Native Library (as described in 8.10~~9~~.1);
- "**`TTPCertificate`**" in the case of a certificate containing a 3rd party root public key (as described in 8.~~9~~10.2);
- "**`ManufacturerCertificate`**" in the case of a certificate containing a manufacturer root public key (as described in 8.10~~9~~.2);
- "**`OperatorCertificate`**" in the case of a certificate containing an operator root public key (as described in 8.10~~9~~.2);
- "**`AdminCertificate`**" in the case of an administrator certificate (as described in 8.10~~9~~.2); or
- "**`CCM`**" in the case of a CCM (as described in 8.~~12~~7); or
- *-free-format-value-* in the case of proprietary binaries or Java classes such as native DSP code, provisioned functionality upgrades and patches (as described in 8.11~~9~~.2).

E.g.

```
MExE-Implementation-Type: MExENativeLibrary
```

See figure ~~7.~~12. When a download of a JAR file is completed, the system installer shall read the manifest to determine what types of files are contained in the JAR, and install them appropriately.

Note that a signed package containing a library which does not have a manifest attribute "MExE-`Implementation-Type: MExENativeLibrary`" shall be considered to be some type of upgrade to libraries that are intrinsically part of the ME implementation rather than a "MExE native library". E.g.

```
MExE-Implementation-Type: ManufacturerUpgrade (something.dll)
```

(Recommended behaviour for the server is that it uses the capability information supplied from the ME to determine how to offer appropriate upgrades.)

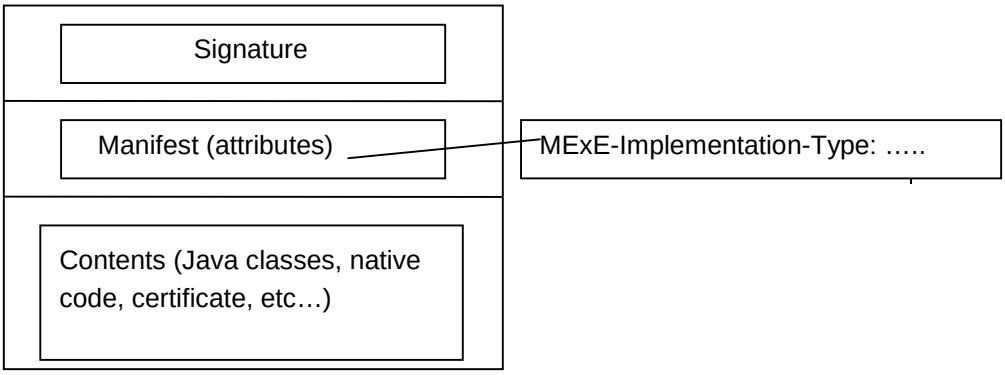

Figure 7:12: signed packages

8.10.1 Installing MExE native libraries

A signed native library whose signature verifies as describe in section 8.48 as belonging to the Manufacturer Domain may be installed as a "MExE native library".

A MExE native library may be called by a MExE executable, and shall not compromise the MExE security system.

Support of MExE native library signed package installation is optional.

8.10.2 Installation of root certificates in a signed data package

Root certificates in a signed package (whose signature verifies as described in section 8.109 to the Manufacturer root, Operator root, or the Administrator root), may be installed to the root public key store on the ME. Note that the certificate thus packaged does not necessarily belong to the manufacturer domain. The types of certificate that can be present and installed by packages are given in Table 4. The ME shall store the root public key as indicated by the certificate type.

When a certificate containing an Administrator root public key is thus contained in a signed package, the signed package (JAR) shall contain two files: the Administrator root public key and the CCM.

8.10.3 Installation of other signed data

A signed package of proprietary binaries or Java classes such as native DSP code, provisioned functionality upgrades and patches, whose signature verifies as described in section 8.84 as belonging to the Manufacturer Domain may be installed. The use of such binaries is outside the scope of MExE, but the manufacturer shall be responsible for ensuring that the integrity of MExE is not compromised.

Support of this feature is optional.

8.10.4 Administrator root certificate download mechanism

Devices supporting SIMs without certificates shall at least support the following procedure to download the administrator root certificate.

1. Upon sign-up with an administrator the user and administrator will make contact.
2. The administrator service centre will obtain any required information from the user and inform the user by SMS or other means of the location of the administrator root certificate.
3. The user will initiate the download of the Administrator root certificate using a signed package.

4. Once the procedure is complete the device shall compute the hash of the received Administrator certificate containing root public key.

5. ~~display~~The user will contact the administrator and enters on the device at least the first 8 bytes using decimal value of the hash of the Administrator root public~~key. The user will read this information back to the administrator. If this information is correct the~~key information provided by the administrator . The device compares the beginning of computed hash value and the abbreviated hash value entered by the user. If these two values are the same~~-~~, the provisioning process will be ~~complete.~~complete. If the two values are different this shall be indicated to the user who should inform the administrator of this.

Alternative methods to download an administrator root certificate may be used where appropriate but must insure that the certificate is received by the device unaltered.

8.11 Pre-verification of applications

This is an optional feature added to eliminate the potentially excessive overhead of checking a signature each time an application is launched.

To use this process the MExE device shall create a hash of the executable object (executable object fingerprint) as if checking the signature. This shall be stored in a protected verified application list, along with indication of the domain permissions for the application. The hash used shall be the same type as that used for signing the object. When launching an application or downloading an applet, the hash shall be performed as for when computing the signature. The verified application list shall then be checked; if the hash value is present and the entry has not expired then the application or applet may execute. If no list entry exists for this object, or the entry has expired, the process shall then proceed with the full signature verification. Note that the lists for applications and applets should be separate and that an implementation determines management policy for the lists (e.g., ageing policy, which entries to delete when trying to add a new entry to a full list etc.). One restriction imposed that shall be enforced is that the maximum number of uses for an entry before it is marked invalid is limited to some maximum value.

In the event that a new CCM is received by the MExE MS, all verified application list entries shall be marked invalid unless some mechanism to determine the validity of an authorising certificate entry for each application is provided by the ME implementation.

~~8.10 WAP security~~

~~The WTLS security model provides a flexible framework based on TLS, which is independent of the GSM security model.~~

3GPP TSG-T2 #8 / ETSI SMG4
Mexico, Puerto Vallarta, 1-4 February 2000

TSGT2#8(00)48

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

03.57	CR	003	Current Version:	7.1.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to:	TSG-T#7	for approval	X	strategic		(for SMG use only)
list expected approval meeting # here ↑		for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG *The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc*

Proposed change affects: *(at least one should be marked with an X)* (U)SIM [ ] ME [X] UTRAN / Radio [ ] Core Network [ ]

Source: T2 **Date:** 1.02.2000

Subject: Corrections to WAP chapters

Work item: MExE

Category: *(only one category shall be marked with an X)*

F	Correction	X
A	Corresponds to a correction in an earlier release	
B	Addition of feature	
C	Functional modification of feature	
D	Editorial modification	

Release:

Phase 2	
Release 96	
Release 97	
Release 98	X
Release 99	
Release 00	

Reason for change: Make corrections to call control description

Clauses affected: 5.3

Other specs affected:

Other 3G core specifications	[ ]	→ List of CRs:
Other GSM core specifications	[ ]	→ List of CRs:
MS test specifications	[ ]	→ List of CRs:
BSS test specifications	[ ]	→ List of CRs:
O&M specifications	[ ]	→ List of CRs:

Other comments:

help.doc

<--------- double-click here for help and instructions on how to create a CR.

5.3 Call control

WAP telephony services are written in WML and WMLScript. The WAP Telephony API (WTAI) exposes telephony functions to service authors as a set of libraries. The WTAI function libraries can be accessed from WML as URIs, and from WMLScript as script functions. The following libraries have been specified:

- Public library
  This includes functions that are available in all networks, and can be provided by any third party service provider; and not only the network operator. The user must acknowledge the function before it is carried out. ~~One Three f~~Functions ~~has~~ have been specified, which can~~.~~ ~~It They can~~ be used e.g. to initiate a mobile originated call, send DTMF tones and add phonebook entry.

- Network Common library
  This includes functions that are available in all networks, and can be provided only by the network operator. E.g. ~~F~~functions for advanced call control, accessing the phonebook, and sending and reading network text (SMS) have been specified.

- Network Specific library
  Functions that are only available in certain types of networks, and can be provided only by the network operator. For GSM, e.g. functions for call reject, call hold, call transfer, ~~and~~ multiparty, getting location information and sending USSD have been specified.

The WML and WMLScript author uses the WTAI libraries to create web services for mobile phones with telephony capabilities.

Call control shall be performed using WTA authenticated scripts.