

CR-Form-v7.1

CHANGE REQUEST

⌘ **33.246 CR 038** ⌘ rev **1** ⌘ Current version: **6.1.0** ⌘

For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.

Proposed change affects: UICC apps ME Radio Access Network Core Network

Title:	Clarify MUK key synchronisation for MSK push procedure		
Source:	Siemens		
Work item code:	MBMS	Date:	4418/02/2005
Category:	C	Release:	Rel-6
	<p><i>Use <u>one</u> of the following categories:</i></p> <p>F (correction) A (corresponds to a correction in an earlier release) B (addition of feature), C (functional modification of feature) D (editorial modification)</p> <p>Detailed explanations of the above categories can be found in 3GPP TR 21.900.</p>		<p><i>Use <u>one</u> of the following releases:</i></p> <p>Ph2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) Rel-4 (Release 4) Rel-5 (Release 5) Rel-6 (Release 6) Rel-7 (Release 7)</p>

Reason for change:	<p>It may happen that the UE has already generated a new MUK/MRK pair (after a GBA run and the subsequent application NAF derivation step) but the BM-SC was never informed. From the BM-SC point of view his known MUK/MRK pair may still be valid (lifetime has not expired), hence this MUK-ID can still be used within the MSK push procedure. While the UE has already installed a new MUK-ID, the BM-SC is using an old MUK for protecting the MSK push MIKEY messages. The UE behavior for this mismatch case is not specified.</p> <p>A similar handling as for the push solicited pull procedure is proposed. For the push solicit pull, the BM-SC is allowed to use a MUK-ID beyond the SA-lifetime (differently than the last generated one). This MUK-ID is known to the UE as the last-successfully used.</p>
Summary of change:	<p>Clarify the UE behavior when receiving a normal MIKEY push message with an old (still valid) MUK-ID. The UE shall handle the MIKEY push message in a similar way as the push solicited pull message. This guarantees that the UE contacts the BM-SC with the B-TID. Subsequently the MSK is pushed again to the UE (yet with the newer MUK).</p> <p>Clarify the handling of two MUKs within the UE.</p>
Consequences if not approved:	<p>UE's may behave differently which may result in non-optimized MSK handling.</p>

Clauses affected:	6.1, 6.3.2.1										
Other specs affected:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">Y</td> <td style="padding: 2px 5px;">N</td> </tr> <tr> <td style="padding: 2px 5px;">X</td> <td style="padding: 2px 5px;"></td> </tr> <tr> <td style="padding: 2px 5px;"></td> <td style="padding: 2px 5px;">X</td> </tr> <tr> <td style="padding: 2px 5px;"></td> <td style="padding: 2px 5px;">X</td> </tr> </table>	Y	N	X			X		X	Other core specifications	⌘ TS 31.102
	Y	N									
	X										
	X										
	X										
	Test specifications										
	O&M Specifications										
Other comments:	⌘										

===== BEGIN CHANGE =====

6.1 Using GBA for MBMS

TS 33.220 [6] (Generic Bootstrapping Architecture) is used to agree keys that are needed to run an MBMS Multicast User service.

Before a user can access an MBMS User service, the UE needs to share GBA-keys with the BM-SC. If no valid GBA-keys are available at the UE, the UE shall perform a GBA run with the BSF of the home network as described within clause 5 of TS 33.220 [6]. The BM-SC will act as a NAF (Network Application Function) according to TS 33.220 [6].

The MSKs for an MBMS User service shall be stored on either the UICC if the UICC is capable of MBMS key management or the ME if the UICC is not capable of MBMS key management.

Storing the MSKs on the UICC requires a UICC that contains the MBMS management functions.

As a result of a GBA_U run, the BM-SC will share a key Ks_ext_NAF with the ME and share a key Ks_int_NAF with the UICC. This key Ks_int_NAF is used by the BM-SC and the UICC as the key MUK (MBMS User Key) to protect MSK (MBMS Service Key) deliveries to the UICC as described within clause 6.3. The key Ks_ext_NAF is used as the key MRK (MBMS Request Key) within the protocols as described within clause 6.2.

A run of GBA_ME results in the BM-SC sharing a key Ks_(ext)_NAF with the ME. This key Ks_(ext)_NAF is used by the BM-SC and the ME to derive the key MUK and the key MRK. The key MUK is used to protect MSK deliveries to the ME as described within clause 6.3. The key MRK is used to authenticate the UE towards the BM-SC within the protocols as described within clause 6.2.

The MUK is identified by the combination of B-TID and NAF-ID and the MRK is defined by B-TID, where B-TID and NAF-ID are defined as specified in TS 33.220 [6].

In the UE two different MUKs, i.e. the last generated and the last successfully used, are used to guarantee that the UE and the BM-SC share always one MUK. The last generated MUK is replaced immediately after when a new MUK is generated and the last successfully used MUK is updated after the successful reception of the MIKEY message, which is protected using the last generated MUK. The usage of MUKs is described within clause 6.3.

For ME based key management:

- All MBMS keys (MUK, MRK, MSK and MTK) shall be deleted from the ME when a different UICC is inserted. Therefore the ME needs to store in non-volatile memory the last inserted UICC-identity to be able to compare that with the used UICC-identity at UICC insertion and power on.
- All MBMS keys (MRK, MSK and MTK) may be deleted from the ME when the ME is powered down. If the ME does not delete the MBMS keys at power down then the MBMS keys need to be stored in non-volatile memory. The ME should store the MUKs in non-volatile memory in order to be able to authenticate the first MIKEY message of a push solicited pull procedure (see clause 6.3.2.2.4).

NOTE: If the ME deletes the MSK at power down, then the MBMS client would need to request MSK to the BM-SC and may need to run GBA to reconvene an MBMS session.

===== END CHANGE =====

===== BEGIN CHANGE =====

6.3.2.3 MSK push procedures

6.3.2.3.1 Pushing the MSKs to the UE

The BM-SC controls when the MSKs used in a multicast service are to be changed. The below flow describes how MSK changes are performed.

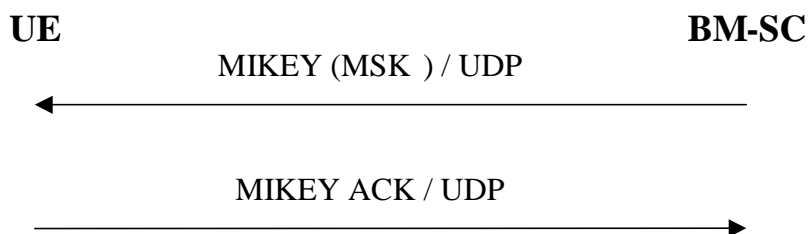


Figure 6.3: Pushing the MSKs to the UE

When the BM-SC decides that it is time to update the MSK, the BM-SC sends MIKEY message over UDP transporting the requested MSKs to the UE.

If requested by the BM-SC, the UE sends a MIKEY acknowledgement message to the BM-SC.

When an MSK push MIKEY message is not directly preceded by an MSK key request, then it may happen that the BM-SC uses a still valid MUK that is not the last generated MUK at the UE. The UE shall handle such a MIKEY push message in a similar way as the push solicited pull MIKEY message (i.e upon a successful integrity check the UE shall initiate an MSK request with the specified Key Group).

NOTE: This procedure guarantees that the UE contacts the BM-SC with the last B-TID, such that the UE now receives a MIKEY push message with the last generated MUK. The integrity of the initial pushed MIKEY message can be verified at the UE with the MUK-ID that is known as the last successfully used BM-SC MUK-ID.

6.3.2.3.2 Void

===== END CHANGE =====