

CHANGE REQUEST

33.220 **CR 040** rev 1 Current version: 6.2.0

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:	☞ New key management for ME based GBA keys		
Source:	☞ SA WG3		
Work item code:	☞ SEC1-SC	Date:	☞ 16/11/2004
Category:	☞ C	Release:	☞ Rel-6
	<p>Use <u>one</u> of the following categories:</p> <p>F (correction)</p> <p>A (corresponds to a correction in an earlier release)</p> <p>B (addition of feature),</p> <p>C (functional modification of feature)</p> <p>D (editorial modification)</p> <p>Detailed explanations of the above categories can be found in 3GPP TR 21.900.</p>		<p>Use <u>one</u> of the following releases:</p> <p><i>Ph2</i> (GSM Phase 2)</p> <p><i>R96</i> (Release 1996)</p> <p><i>R97</i> (Release 1997)</p> <p><i>R98</i> (Release 1998)</p> <p><i>R99</i> (Release 1999)</p> <p><i>Rel-4</i> (Release 4)</p> <p><i>Rel-5</i> (Release 5)</p> <p><i>Rel-6</i> (Release 6)</p> <p><i>Rel-7</i> (Release 7)</p>

Reason for change:	☞ To improve performance, new key management requirements are added. Similar wording is used for the SA3 approved CR 001 to TS 33.246 (S3-040863).
Summary of change:	☞ GBA keys are not always deleted when the ME is powered down. The new key management procedures for the ME based keys are: - all GBA related keys 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 GBA related keys may be deleted from the ME when the ME is powered down. If the ME does not delete the GBA keys at power down then the GBA keys need to be stored in non-volatile memory.
Consequences if not approved:	☞ GBA keys are always deleted from the ME when the ME is powered down.

Clauses affected:	☞ 4.5.3, 5.3.3						
Other specs affected:	<table border="1" style="border-collapse: collapse;"> <tr> <td style="padding: 2px;">Y</td> <td style="padding: 2px;">N</td> </tr> <tr> <td style="padding: 2px;"><input type="checkbox"/></td> <td style="padding: 2px;"><input checked="" type="checkbox"/></td> </tr> </table>	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>	☞ Other core specifications	☞
Y	N						
<input type="checkbox"/>	<input checked="" type="checkbox"/>						
	<input checked="" type="checkbox"/>	☞ Test specifications	☞				
	<input checked="" type="checkbox"/>	☞ O&M Specifications	☞				
Other comments:	☞						

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

4.5.3 Procedures using bootstrapped Security Association

Before communication between the UE and the NAF can start, the UE and the NAF first have to agree whether to use shared keys obtained by means of the GBA. If the UE does not know whether to use GBA with this NAF, it uses the Initiation of Bootstrapping procedure described in clause 4.5.1.

Once the UE and the NAF have established that they want to use GBA then every time the UE wants to interact with an NAF the following steps are executed as depicted in figure 4.4.

UE starts communication over reference point Ua with the NAF:

- in general, UE and NAF will not yet share the key(s) required to protect the reference point Ua. If they already do (i.e. if a key Ks_NAF for the corresponding key derivation parameter NAF_Id is already available), the UE and the NAF can start to securely communicate right away. If the UE and the NAF do not yet share a key, the UE proceeds as follows:
 - if a key Ks for the selected UICC application is available in the UE, the UE derives the key Ks_NAF from Ks, as specified in clause 4.5.2;
 - if no key Ks for the selected UICC application is available in the UE, the UE first agrees on a new key Ks with the BSF over the reference point Ub, and then proceeds to derive Ks_NAF;

NOTE 1: If it is not desired by the UE to use the same Ks for the selected UICC application to derive more than one Ks_NAF then the UE should agree on a new key Ks with the BSF over the reference point Ub, and then proceed to derive Ks_NAF;

- if the NAF shares a key with the UE, but the NAF requires an update of that key, e.g. because the key's lifetime has expired, it shall send a suitable bootstrapping renegotiation request to the UE and terminates the protocol used over reference point Ua, see figure 4.5. The form of this indication depends on the particular protocol used over reference point Ua. If the UE receives a bootstrapping renegotiation request, it starts a run of the protocol over reference point Ub, as specified in clause 4.5.2, in order to obtain a new key Ks.

NOTE 2: To allow for consistent key derivation in BSF and UE, both have to use the same FQDN for derivation (see NOTE 2 of clause 4.5.2). For each protocol used over Ua it shall be specified if only cases (1) and (2) of NOTE 2 of clause 4.5.2 are allowed for the NAF or if the protocol used over Ua shall transfer also the FQDN used for key derivation by UE to NAF.

NOTE 3: If the shared key between UE and NAF is invalid, the NAF can set deletion conditions to the corresponding security association for subsequent removal.

- the UE supplies the B-TID to the NAF, in the form as specified in clause 4.3.2, to allow the NAF to retrieve the corresponding keys from the BSF;

NOTE 4: The UE may adapt the key material Ks_NAF to the specific needs of the reference point Ua. This adaptation is outside the scope of this specification.

~~—when the UE is powered down, or when the UICC is removed, any keys Ks and Ks_NAF shall be deleted from storage;~~

- key management for GBA related keys in the ME (i.e., Ks and Ks_NAF keys):

- all GBA related keys 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;
- the Key Ks shall be deleted from the ME when the ME is powered down;
- all other GBA related keys may be deleted from the ME when the ME is powered down. If the ME does not delete the GBA keys at power down then the GBA keys need to be stored in non-volatile memory.
- when a new Ks is agreed over the reference point Ub and a key Ks_NAF, derived from one NAF_Id, is updated, the other keys Ks_NAF, derived from different values NAF_Id, stored on the UE shall not be affected;

NAF starts communication over reference point Zn with BSF

- The NAF requests key material corresponding to the B-TID supplied by the UE to the NAF over reference point Ua. If the NAF has several FQDNs, which may be used in conjunction with this specification, then the NAF shall transfer in the request over Zn the same FQDN, which was used over Ua (see NOTE 2 on key derivation in this clause);
- The NAF may also request application-specific user security settings for the applications, which the request received over Ua from UE may access;
- With the key material request, the NAF shall supply NAF’s public hostname that UE has used to access NAF to BSF, and BSF shall be able verify that NAF is authorized to use that hostname;
- The BSF derives the keys required to protect the protocol used over reference point Ua from the key Ks and the key derivation parameters, as specified in clause 4.5.2, and supplies to NAF the requested key Ks_NAF, as well as the lifetime of that key. If the key identified by the B-TID supplied by the NAF is not available at the BSF, the BSF shall indicate this in the reply to the NAF. The NAF then indicates a bootstrapping renegotiation request to the UE.

NOTE 5: The NAF shall adapt the key material Ks_NAF to the specific needs of the reference point Ua in the same way as the UE did. This adaptation is outside the scope of this specification.

- The BSF may also send the private user identity (IMPI) and requested user security settings to NAF according to the BSF's policy;

NAF continues with the protocol used over the reference point Ua with the UE.

Once the run of the protocol used over reference point Ua is completed the purpose of bootstrapping is fulfilled as it enabled UE and NAF to use reference point Ua in a secure way.

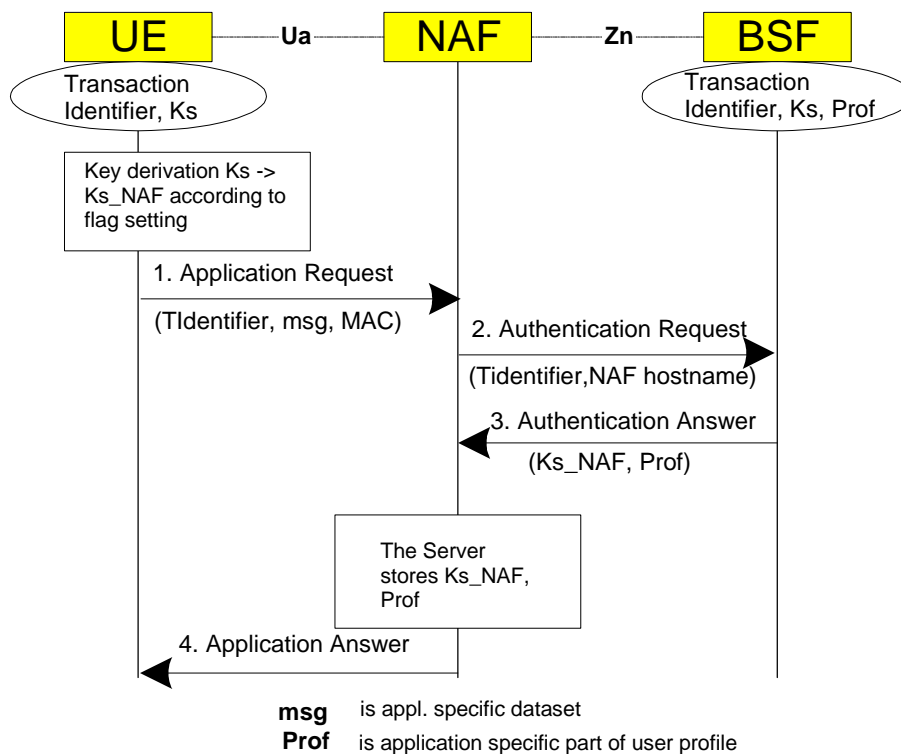


Figure 4.4: The bootstrapping usage procedure

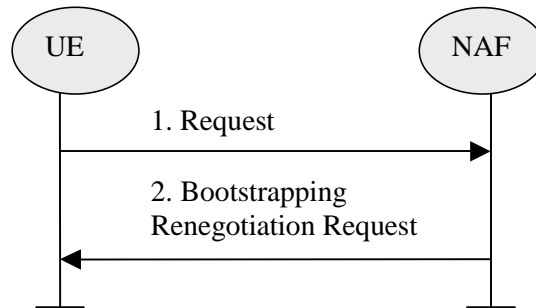


Figure 4.5: Bootstrapping renegotiation request

===== BEGIN NEXT CHANGE =====

5.3.3 Procedures using bootstrapped Security Association

Before communication between the UE and the NAF can start, the UE and the NAF first have to agree whether to use shared keys obtained by means of the GBA. If the UE does not know whether to use GBA with this NAF, it uses the Initiation of Bootstrapping procedure described in clause 5.3.1.

Once the UE and the NAF have established that they want to use GBA then every time the UE wants to interact with a NAF the following steps are executed as depicted in figure 5.3.

Next, the UE and the NAF have to agree, which type of keys to use, Ks_{ext_NAF} or Ks_{int_NAF} , or both. The default is the use of Ks_{ext_NAF} only. This use is also supported by MEs and NAFs, which are GBA_U unaware. If Ks_{int_NAF} , or both Ks_{ext} and Ks_{int} are to be used, this use has to be agreed between UE and NAF prior to the execution of the procedure described in the remainder of this clause 5.3.3. Any such agreement overrides the default use of the keys. How this agreement is reached is application-specific and is not within the scope of this document.

NOTE 1: This agreement may be mandated by the specification, which defines the U_a reference point between UE and NAF, e.g. TS 33.246 for the use of GBA in MBMS, or negotiated by the NAF and the UE over the U_a reference point, or reached by configuration.

Editors' Note: The support of unaware GBA_U MEs, which are GBA_{ME} aware only is FFS.

In general, UE and NAF will not yet share the key(s) required to protect the U_a reference point. If they do not, the UE proceeds as follows:

- if Ks_{ext_NAF} is required and a key Ks_{ext} for the selected UICC application is available in the UE, the UE derives the key Ks_{ext_NAF} from Ks_{ext} , as specified in clause 5.3.2;
- if Ks_{int_NAF} is required and a key Ks_{int} for the selected UICC application is available in the UICC, the ME requests the UICC to derive the key Ks_{int_NAF} from Ks_{int} , as specified in clause 5.3.2;

NOTE 2: If it is not desired by the UE to use the same $Ks_{ext/int}$ for the selected UICC application to derive more than one Ks_{ext/int_NAF} then the UE should first agree on new keys Ks_{ext} and Ks_{int} with the BSF over the U_b reference point, as specified in clause 5.3.2, and then proceeds to derive Ks_{ext_NAF} or Ks_{int_NAF} , or both, as required.

- if Ks_{ext} and Ks_{int} for the selected UICC application are not available in the UE, the UE first agrees on new keys Ks_{ext} and Ks_{int} with the BSF over the U_b reference point, as specified in clause 5.3.2, and then proceeds to derive Ks_{ext_NAF} or Ks_{int_NAF} , or both, as required;
- if the NAF shares a key with the UE, but the NAF requires an update of that key, it shall send a suitable bootstrapping renegotiation request to the UE and terminate the protocol used over U_a reference point. The form of this indication depends on the particular protocol used over U_a reference point. If the UE receives a bootstrapping renegotiation request, it starts a run of the protocol over U_b , as specified in clause 5.3.2, in order to obtain new keys.

NOTE 3: If the shared keys between UE and NAF become invalid, the NAF can set deletion conditions to the corresponding security association for subsequent removal.

NOTE 4: If it is not desired by the NAF to use the same Ks to derive more than one Ks_int/ext_NAF then the NAF should always reply to the first request sent by a UE by sending a key update request to the UE.

UE and NAF can now start the communication over Ua reference point using the keys Ks_ext_NAF or Ks_int_NAF, or both, as required. They proceed as follows:

- The UE supplies the B-TID to the NAF, as specified in clause 5.3.2, to allow the NAF to retrieve the corresponding keys from the BSF

NOTE 5: To allow for consistent key derivation in BSF and UE, both have to use the same FQDN for derivation (cf. NOTE 2 of clause 4.5.2). For each protocol used over Ua it shall be specified if only cases (1) and (2) of NOTE 2 of clause 4.5.2 are allowed for the NAF or if the protocol used over Ua shall transfer also the FQDN used for key derivation by UE to NAF.

NOTE 6: The UE may adapt the keys Ks_ext_NAF or Ks_int_NAF to the specific needs of the Ua reference point. This adaptation is outside the scope of this specification.

~~— when the UE is powered down, or when the UICC is removed, any GBA_U keys shall be deleted from storage in the ME. There is no need to delete keys Ks_int and Ks_int_NAF from storage in the UICC;~~

- key management for GBA related keys in the ME (i.e., Ks_ext_NAF keys):
 - all GBA related keys 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 GBA related keys may be deleted from the ME when the ME is powered down. If the ME does not delete the GBA keys at power down then the GBA keys need to be stored in non-volatile memory.
 - all GBA related keys in the UICC do not need to be deleted when the ME is powered down.

NOTE 7: After each run of the protocol over the Ub reference point, new keys Ks_ext and Ks_int, associated with a new B-TID, are derived in the UE according to clause 5.3.2, so that it can never happen, that keys Ks_ext and Ks_int with different B-TIDs simultaneously exist in the UE.

- When new keys Ks_ext and Ks_int are agreed over the Ub reference point and new NAF-specific keys need to be derived for one NAF_Id, then both, Ks_ext_NAF and Ks_int_NAF (if present), shall be updated for this NAF_Id, but further keys Ks_ext_NAF or Ks_int_NAF relating to other NAF_Ids, which may be stored on the UE, shall not be affected;

NOTE 8: This rule ensures that the keys Ks_ext_NAF and Ks_int_NAF are always in synch at the UE and the NAF.

NAF now starts communication over the Zn reference point with the BSF.

- The NAF requests from the BSF the keys corresponding to the B-TID, which was supplied by the UE to the NAF over the Ua reference point. If the NAF is GBA_U aware it indicates this by including a corresponding flag in the request. If the NAF has several FQDNs, which may be used in conjunction with this specification, then the NAF shall transfer in the request over Zn the same FQDN, which was used over Ua (see note above on key derivation in this clause).
- The NAF may also request application-specific user security settings for the applications, which the request received over Ua from UE may access;
- With the keys request over the Zn reference point, the NAF shall supply NAF's public hostname that UE has used to access NAF to BSF, and BSF shall be able to verify that NAF is authorized to use that hostname.
- The BSF derives the keys Ks_ext_NAF, and Ks_int_NAF (if additionally required), as specified in clause 5.3.2. If the NAF indicated in its request that it is GBA_U aware, the BSF supplies to NAF both keys, Ks_ext_NAF, and Ks_int_NAF, otherwise the BSF supplies only Ks_ext_NAF. In addition, the BSF supplies the lifetime time of these keys. If the key identified by the B-TID supplied by the NAF is not available at the BSF, the BSF shall indicate this in the reply to the NAF. The NAF then indicates a bootstrapping renegotiation request (See figure 4.5) to the UE;

NOTE: The NAF may adapt the keys Ks_ext_NAF and Ks_int_NAF to the specific needs of the Ua reference point in the same way as the UE did. This adaptation is outside the scope of this specification.

- The BSF may also send the private user identity (IMPI) and requested user security settings to NAF according to the BSF's policy.

The NAF now continues with the protocol used over the Ua reference point with the UE.

Once the run of the protocol used over Ua reference point is completed the purpose of bootstrapping is fulfilled as it enabled the UE and NAF to use Ua reference point in a secure way.

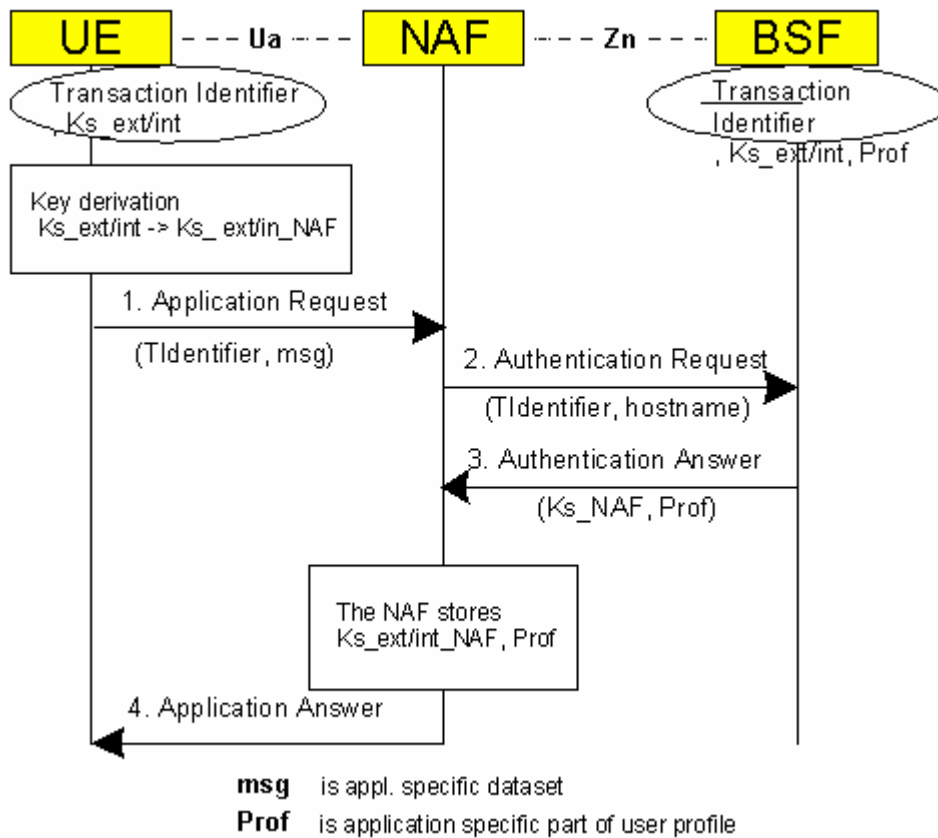


Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

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