

CR-Form-v7.1

## CHANGE REQUEST

33.220 CR 039 rev - 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

<b>Title:</b>	Clean up of TS 33.220		
<b>Source:</b>	Ericsson		
<b>Work item code:</b>	SEC1-SC	<b>Date:</b>	22/11/2004
<b>Category:</b>	<b>F</b>	<b>Release:</b>	Rel-6
	Use <u>one</u> of the following categories: <b>F</b> (correction) <b>A</b> (corresponds to a correction in an earlier release) <b>B</b> (addition of feature), <b>C</b> (functional modification of feature) <b>D</b> (editorial modification) Detailed explanations of the above categories can be found in 3GPP <a href="#">TR 21.900</a> .		Use <u>one</u> of the following releases: <b>Ph2</b> (GSM Phase 2) <b>R96</b> (Release 1996) <b>R97</b> (Release 1997) <b>R98</b> (Release 1998) <b>R99</b> (Release 1999) <b>Rel-4</b> (Release 4) <b>Rel-5</b> (Release 5) <b>Rel-6</b> (Release 6) <b>Rel-7</b> (Release 7)

<b>Reason for change:</b>	Cleaning up TS 33.220 for rel-6		
<b>Summary of change:</b>	Figures 4.3, 4.4, 5.1 and 5.3 are been updated with the following changes: -the naming B-TID is used instead of Identifier, Tidentifier and Transaction Identifier, as B-TID is already defined in abbreviations, -Key lifetime parameter is added to Zn interface from BSF to NAF -Key lifetime parameter is added to Ub interface from BSF to UE  In Figure 5.3, in message 3 from BSF to NAF, parameter name Ks_NAF has been corrected to Ks_ext/int_NAF.  The text in chapter 4.5.3 has been mapped to the messages in figure 4.4.  The text in chapter 5.3.3 has been mapped to the messages in figure 5.3.		
<b>Consequences if not approved:</b>	Unclear specification.		

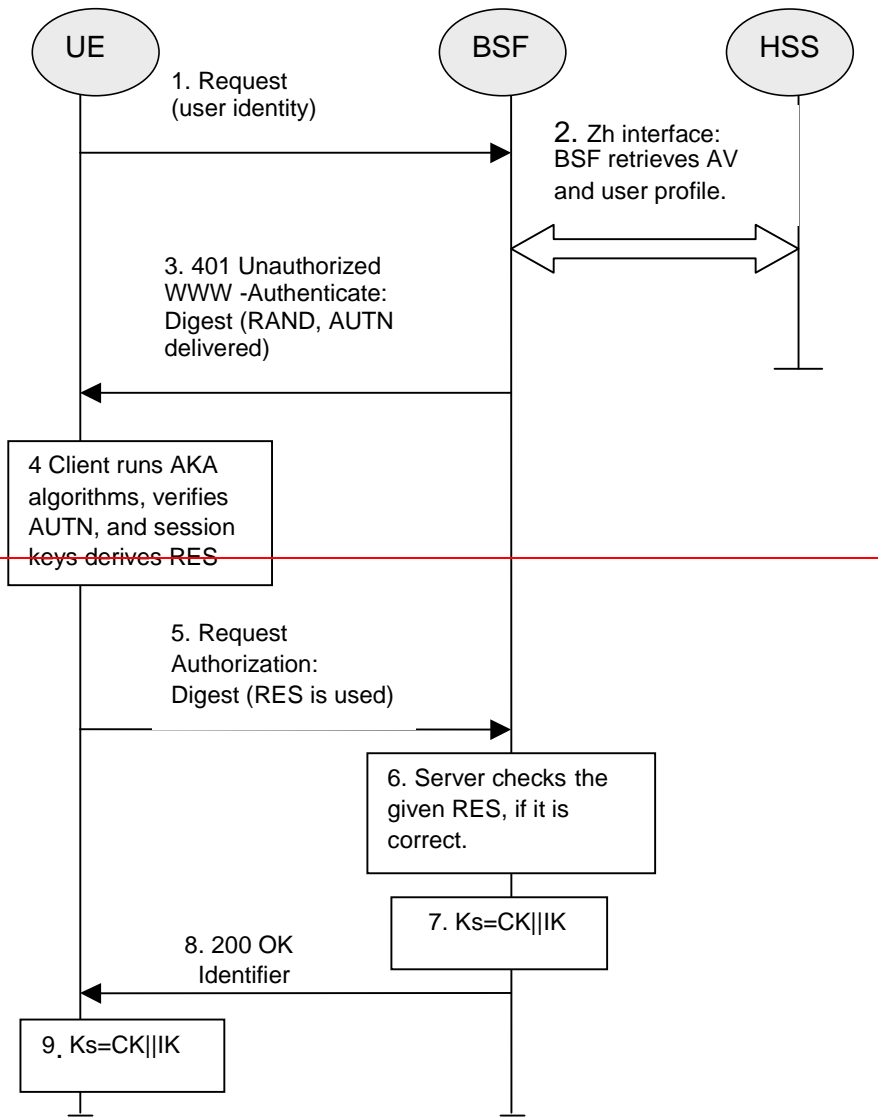
<b>Clauses affected:</b>	4.5.2, 4.5.3, 5.3.2, 5.3.3										
<b>Other specs affected:</b>	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">Y</td> <td style="width: 20px; text-align: center;">N</td> </tr> <tr> <td style="width: 20px; text-align: center;"> </td> <td style="width: 20px; text-align: center;"> </td> </tr> <tr> <td style="width: 20px; text-align: center;"> </td> <td style="width: 20px; text-align: center;"> </td> </tr> <tr> <td style="width: 20px; text-align: center;"> </td> <td style="width: 20px; text-align: center;"> </td> </tr> </table> Other core specifications <input type="checkbox"/> Test specifications <input type="checkbox"/> O&M Specifications <input type="checkbox"/>	Y	N								
Y	N										
<b>Other comments:</b>											

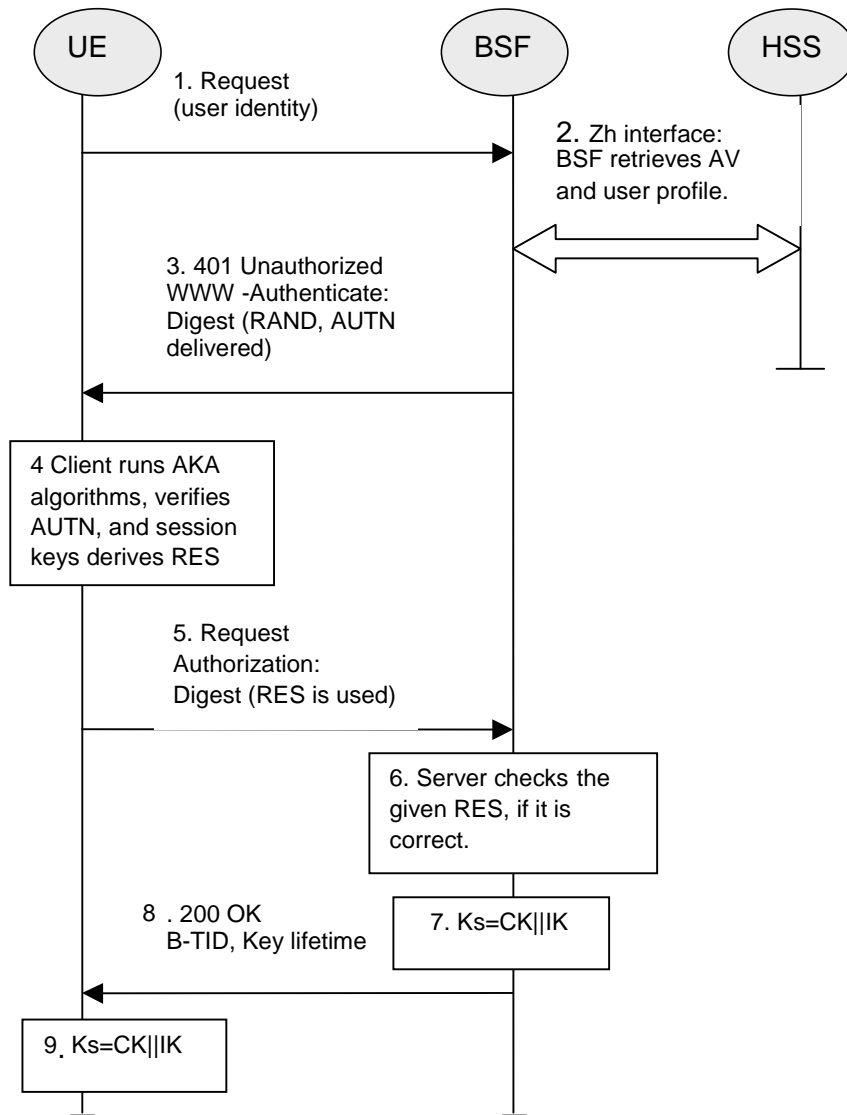
\*\*\*\*\* Begin of Change \*\*\*\*\*

## 4.5.2 Bootstrapping procedures

When a UE wants to interact with a NAF, and it knows that the bootstrapping procedure is needed, it shall first perform a bootstrapping authentication (see figure 4.3). Otherwise, the UE shall perform a bootstrapping authentication only when it has received bootstrapping initiation required message or a bootstrapping negotiation indication from the NAF, or when the lifetime of the key in UE has expired (cf. subclause 4.5.3).

NOTE 1: The main steps from the specifications of the AKA protocol in TS 33.102 [2] and the HTTP digest AKA protocol in RFC 3310 [4] are repeated in figure 3 for the convenience of the reader. In case of any potential conflict, the specifications in TS 33.102 [2] and RFC 3310 [4] take precedence.





**Figure 4.3: The bootstrapping procedure**

1. The UE sends an HTTP request towards the BSF.
2. BSF retrieves the complete set of GBA user security settings and one or a whole batch of Authentication Vectors (AV, AV = RAND||AUTN||XRES||CK||IK) over the reference point Zh from the HSS.
3. Then BSF forwards the RAND and AUTN to the UE in the 401 message (without the CK, IK and XRES). This is to demand the UE to authenticate itself.
4. The UE checks AUTN to verify that the challenge is from an authorised network; the UE also calculates CK, IK and RES. This will result in session keys IK and CK in both BSF and UE.
5. The UE sends another HTTP request, containing the Digest AKA response (calculated using RES), to the BSF.
6. The BSF authenticates the UE by verifying the Digest AKA response.
7. The BSF generates key material  $K_s$  by concatenating CK and IK. The B-TID value shall be also generated in format of NAI by taking the base64 encoded [12] RAND value from step 3, and the BSF server name, i.e. base64encode(RAND)@BSF\_servers\_domain\_name.
8. The BSF shall send a 200 OK message, including a B-TID, to the UE to indicate the success of the authentication. In addition, in the 200 OK message, the BSF shall supply the lifetime of the key  $K_s$ . The key material  $K_s$  is generated in UE by concatenating CK and IK.

9. Both the UE and the BSF shall use the Ks to derive the key material Ks\_NAF during the procedures as specified in clause 4.5.3. Ks\_NAF shall be used for securing the reference point Ua.

Ks\_NAF is computed as  $Ks\_NAF = KDF(Ks, \text{key derivation parameters})$ , where KDF is a suitable key derivation function, and the key derivation parameters consist of the user's IMPI, the NAF\_Id and RAND. The NAF\_Id consists of the full DNS name of the NAF. KDF shall be implemented in the ME.

NOTE 2: To allow consistent key derivation based on NAF name in UE and BSF, at least one of the three following prerequisites shall be fulfilled:

- (1) The NAF is known in DNS under one domain name (FQDN) only, i.e. no two different domain names point to the IP address of the NAF. This has to be achieved by administrative means. This prerequisite is not specific to 3GPP, as it is necessary also under other circumstances, e.g. for TLS V1.0 without use of wildcard or multiple-name certificates.
- (2) Each DNS entry of the NAF points to a different IP address. The NAF responds to all these IP addresses. Each IP address is tied to the corresponding FQDN by NAF configuration. The NAF can see from the IP address, which FQDN to use for key derivation.
- (3) Ua uses a protocol which transfers the host name (FQDN of NAF as used by UE) to NAF (e.g. HTTP/1.1 with mandatory Host request header field). This requires the NAF to check the validity of the host name, to use this name in all communication with UE where appropriate, and to transfer this name to BSF to allow for correct derivation of Ks\_NAF. In case of a TLS tunnel this requires either multiple-identities certificates or the deployment of RFC 3546 [9] or other protocol means with similar purpose.

**Editor's note:** The definition of the KDF is left to ETSI SAGE and is to be included in the Annex B of the present specification.

The UE and the BSF shall store the key Ks with the associated B-TID for further use, until the lifetime of Ks has expired, or until the key Ks is updated.

\*\*\*\*\* Next 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.

1. 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;
- 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;

#### 2. 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 to verify that NAF is authorized to use that hostname;

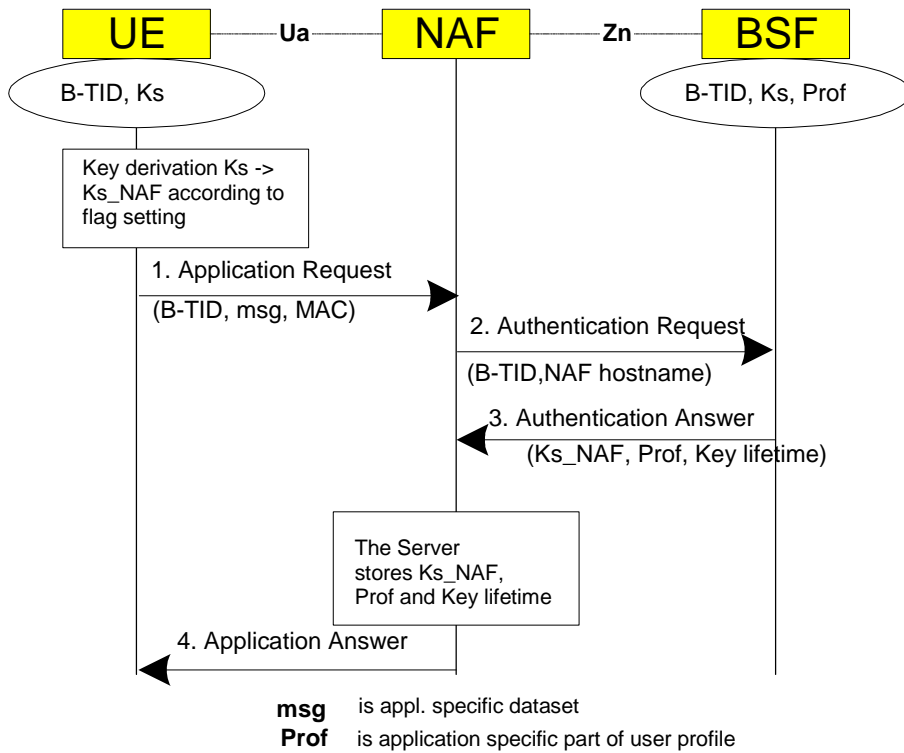
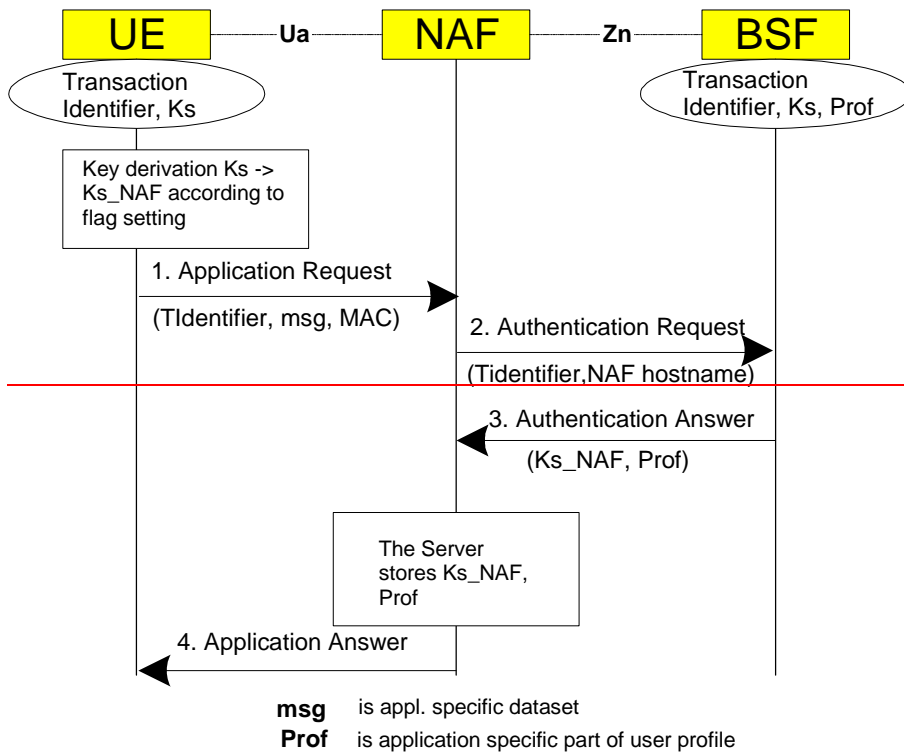
3. -The BSF derives the keys required to protect the protocol used over reference point Ua from the key Ks and the key derivation parameters, as specified in clause 4.5.2, and supplies to NAF the requested key Ks\_NAF, as well as the lifetime of that key. If the key identified by the B-TID supplied by 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;

#### 4. 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**

\*\*\*\*\* Next Change \*\*\*\*\*

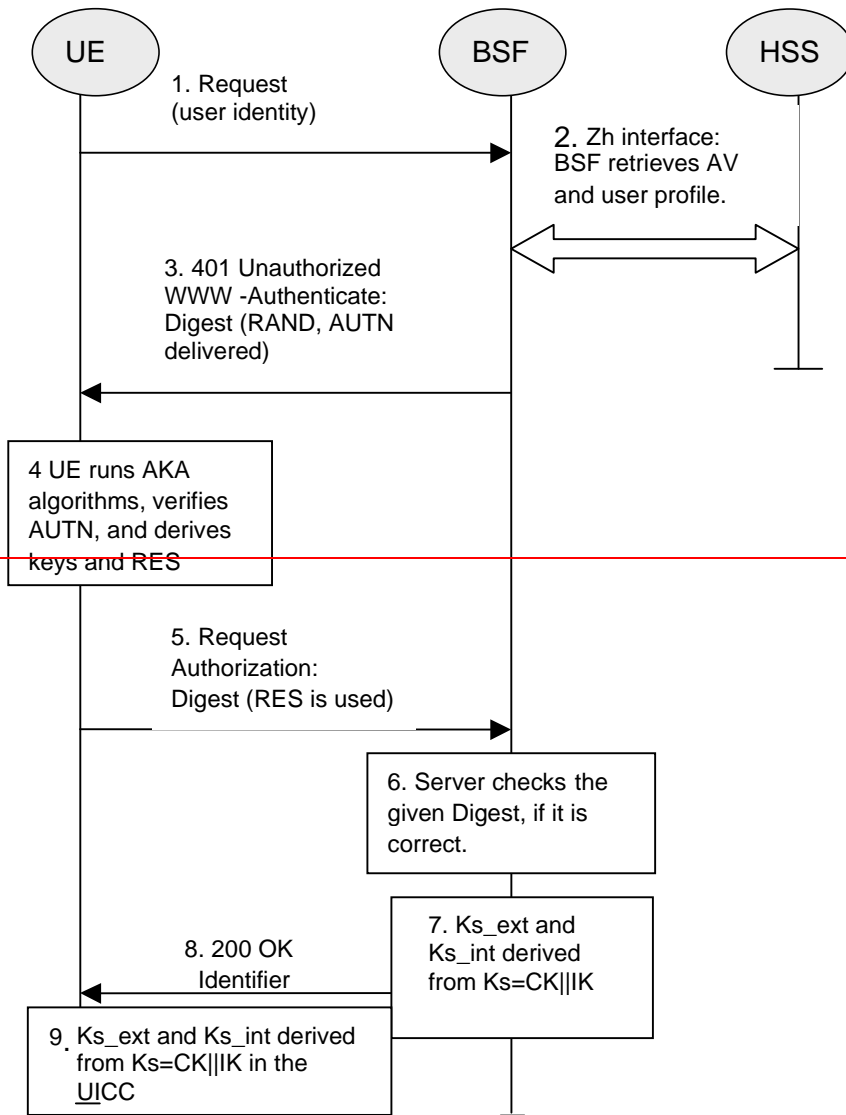
### 5.3.2 Bootstrapping procedure

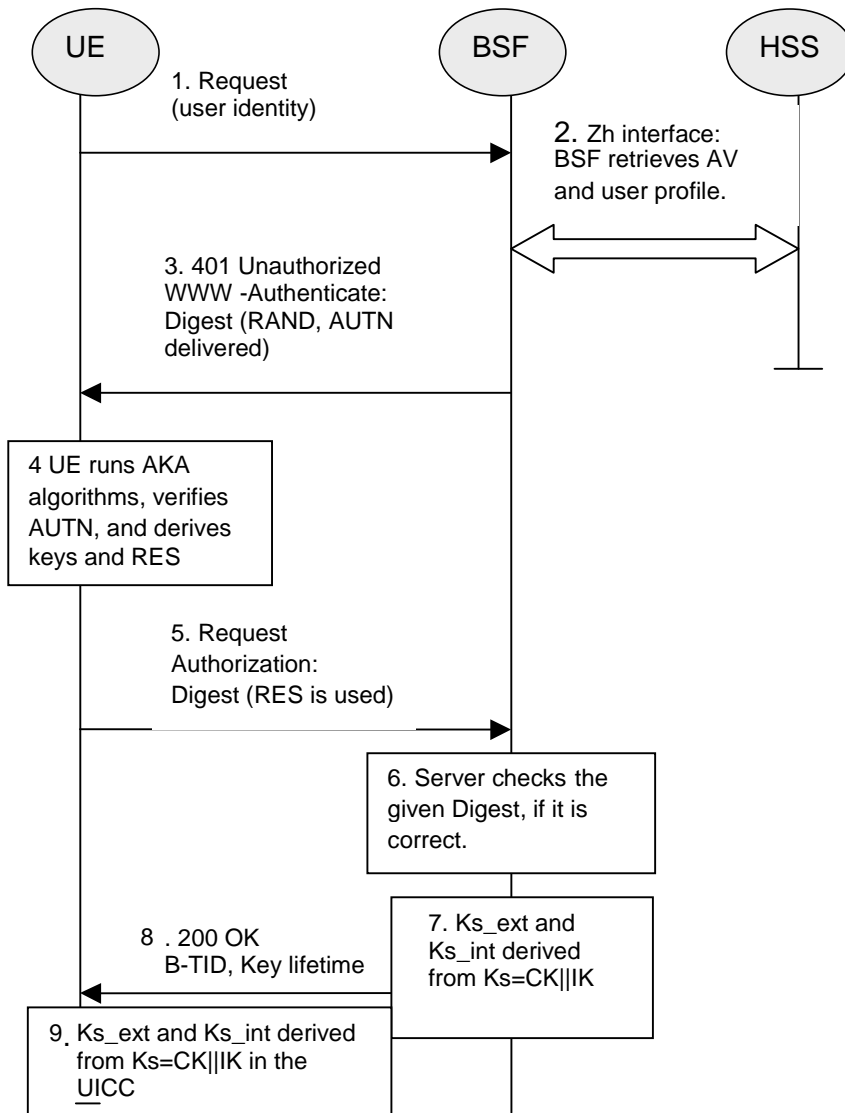
The procedure specified in this clause differs from the procedure specified clause 4.5.2 in the local handling of keys and Authentication Vectors in the UE and the BSF. The messages exchanged over the Ub reference point are identical for both procedures.

When a UE wants to interact with a NAF, and it knows that the bootstrapping procedure is needed, it shall first perform a bootstrapping authentication (see figure 5.1). Otherwise, the UE shall perform a bootstrapping authentication only when it has received bootstrapping initiation required message or a bootstrapping renegotiation indication from the NAF, or when the lifetime of the key in UE has expired (see clause 5.3.3).

NOTE: The main steps from the specifications of the AKA protocol in TS 33.102 [2] and the HTTP digest AKA protocol in RFC 3310 [4] are repeated in figure 5.1 for the convenience of the reader. In case of any potential conflict, the specifications in TS 33.102 [2] and RFC 3310 [4] take precedence.







**Figure 5.1: The bootstrapping procedure with UICC-based enhancements**

1. The ME sends an HTTP request towards the BSF.
2. The BSF retrieves the complete set of GBA user security settings and one or a whole batch of Authentication Vectors (AV,  $AV = RAND \parallel AUTN \parallel XRES \parallel CK \parallel IK$ ) over the Zh reference point from the HSS. The BSF can then decide to perform GBA<sub>U</sub>, based on the user security settings (USSs). In this case, the BSF proceeds in the following way:

- BSF computes  $MAC^* = MAC \oplus SHA-1(IK1)$  (where  $IK = IK1 \parallel IK2$  and \* is a exclusive or as described in TS 33.102 [2])

**Editor's note:** The exact format of the MAC modification function is to be reviewed. The output of SHA-1 needs to be truncated to exact amount of bits needed (64 bits).

The BSF stores the XRES after flipping the least significant bit.

3. Then BSF forwards the RAND and AUTN\* (where  $AUTN^* = SQN \oplus AK \parallel AMF \parallel MAC^*$ ) to the UE in the 401 message (without the CK, IK and XRES). This is to demand the UE to authenticate itself.
4. The ME sends RAND and AUTN\* to the UICC. The UICC calculates IK and MAC (by performing  $MAC = MAC^* \oplus SHA-1(IK1)$ ). Then the UICC checks AUTN (i.e.  $SQN \oplus AK \parallel AMF \parallel MAC$ ) to verify that the

challenge is from an authorised network; the UICC also calculates CK and RES. This will result in session keys CK and IK in both BSF and UICC.

5. The UICC then applies a suitable key derivation function h1 to Ks, which is the concatenation of CK and IK, and possibly further h1-key derivation parameters to obtain two keys, Ks\_ext and Ks\_int, each of length 128 bit, i.e.  $h1(Ks, h1 \text{ key derivation parameters}) = Ks\_ext \parallel Ks\_int$  (see also figure 5.2). The UICC then transfers RES (after flipping the least significant bit) and Ks\_ext to the ME and stores Ks\_int/ks\_ext on the UICC.

**Editors' Note:** The definition of the h1 is left to ETSI SAGE and is to be included in the Annex B of the present specification.

**Editors' Note:** The location (whether in the UICC or in the ME) of the storage of Ks\_ext is ffs.

6. The ME sends another HTTP request, containing the Digest AKA response (calculated using RES), to the BSF.
7. The BSF authenticates the UE by verifying the Digest AKA response.
8. The BSF generates the key Ks by concatenating CK and IK. Then the BSF applies the key derivation function h1 to Ks and possibly further h1-key derivation parameters to obtain two keys, Ks\_ext and Ks\_int, in the same way as the UICC did in step 5. The B-TID value shall be also generated in format of NAI by taking the base64 encoded [12] RAND value from step 3, and the BSF server name, i.e. `base64encode(RAND)@BSF_servers_domain_name`.
9. The BSF shall send a 200 OK message, including the B-TID, to the UE to indicate the success of the authentication. In addition, in the 200 OK message, the BSF shall supply the lifetime of the keys Ks\_ext and Ks\_int, The lifetimes of the keys Ks\_ext and Ks\_int shall be the same.
10. The BSF shall use the keys Ks\_ext and Ks\_int to derive the NAF-specific keys Ks\_ext\_NAF and Ks\_int\_NAF, if requested by a NAF over the Zn reference point. Ks\_ext\_NAF and Ks\_int\_NAF are used for securing the Ua reference point. The UE shall use the key Ks\_ext to derive the NAF-specific key Ks\_ext\_NAF, if applicable. The UICC shall use the key Ks\_int to derive the NAF-specific key Ks\_int\_NAF, if applicable.

Ks\_ext\_NAF is computed as  $Ks\_ext\_NAF = h2(Ks\_ext, h2\text{-key derivation parameters})$ , and Ks\_int\_NAF is computed in the UICC as  $Ks\_int\_NAF = h2(Ks\_int, h2\text{-key derivation parameters})$ , where h2 is a suitable key derivation function, and the h2-key derivation parameters include the user's IMPI, the NAF\_Id and RAND. The NAF\_Id consists of the full DNS name of the NAF.

**Editors' Note:** The definition of the h2 is left to ETSI SAGE and is to be included in the Annex B of the present specification.

**NOTE:** The NOTE 2 of clause 4.5.2 also applies here.

The ME, the UICC and the BSF store the keys Ks\_ext and Ks\_int together with the associated B-TID for further use, until the lifetime of Ks\_ext and Ks\_int has expired, or until the keys Ks\_ext and Ks\_int are updated.

\*\*\*\*\* 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 overrules 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 Ua 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 Ua 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 Ua 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 Ub 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 Ub 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 Ua reference point. The form of this indication depends on the particular protocol used over Ua reference point. If the UE receives a bootstrapping renegotiation request, it starts a run of the protocol over Ub, 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.

**1.** 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;

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.

**2.** 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.

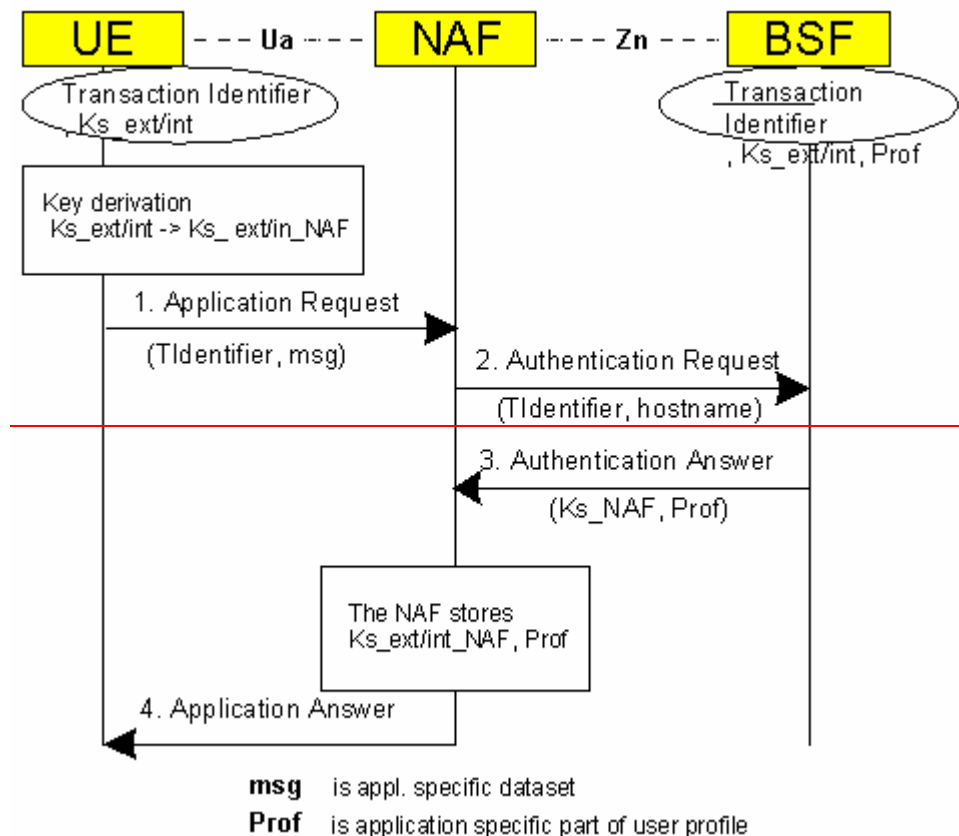
3.-The BSF derives the keys  $Ks\_ext\_NAF$ , and  $Ks\_int\_NAF$  (if additionally required), as specified in clause 5.3.2. If the NAF indicated in its request that it is GBA\_U aware, the BSF supplies to NAF both keys,  $Ks\_ext\_NAF$ , and  $Ks\_int\_NAF$ , otherwise the BSF supplies only  $Ks\_ext\_NAF$ . 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.

4. 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.



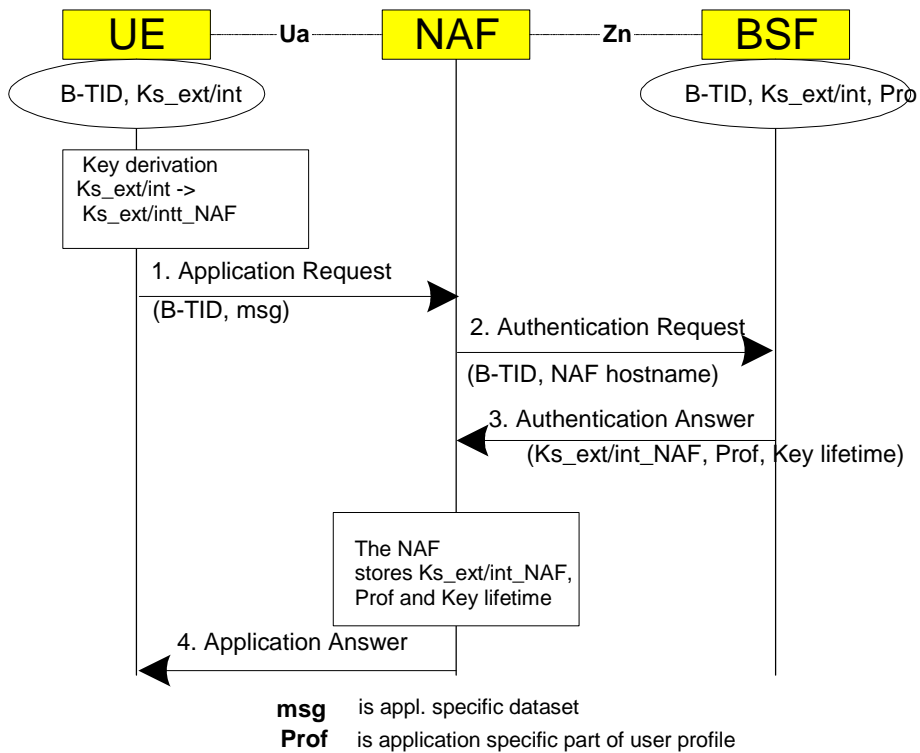


Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements