

## CHANGE REQUEST

⌘ **33.220 CR 038** ⌘ 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

<b>Title:</b>	⌘ Fetching of one AV only on each Zh run between BSF and HSS		
<b>Source:</b>	⌘ SA WG3		
<b>Work item code:</b>	⌘ SEC1-SC	<b>Date:</b>	⌘ 15/11/2004
<b>Category:</b>	⌘ <b>C</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: 2 (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)

<b>Reason for change:</b>	⌘ Fetching of one AV only per Zh run has the following advantages: (i) GUSS stored in BSF for a user has a lifetime of maximal the key lifetime of Ks, and not a lifetime until the last stored AV is used up. This avoids possible long existence of stale GUSS in BSF after change of GUSS in HSS without the need to introduce a GUSS update mechanism over Zh. (ii) NAF may request update of USS at any time by simply indicating "renegotiation request" to UE. Then UE runs Ub, BSF runs Zh receiving current (possibly new) GUSS, and NAF receives new USS on retrieval of new Ks_NAF over Zn. (iii) No special handling of sequence numbers in AuC, in particular if more than one BSF exists in home network. The common reason to send more than one AV in a response is to limit traffic between home network and visited network and to reduce response time to the terminal. This does not apply here, as BSF is always in home network, and as bootstrapping is not as time critical as e.g. voice call setup. Note: Rel. 7 may again introduce the sending of multiple AVs in one Zh run, together with possible introduction of GUSS update procedure over Zh and Zn.
<b>Summary of change:</b>	⌘ Only one authentication vector AV may be fetched by BSF from HSS on each protocol run over Zh reference point.
<b>Consequences if not approved:</b>	⌘ (i) reduced control of GUSS freshness, (ii) Document has to be extended with respect to storage and usage of stored AVs in BSF (missing until now).

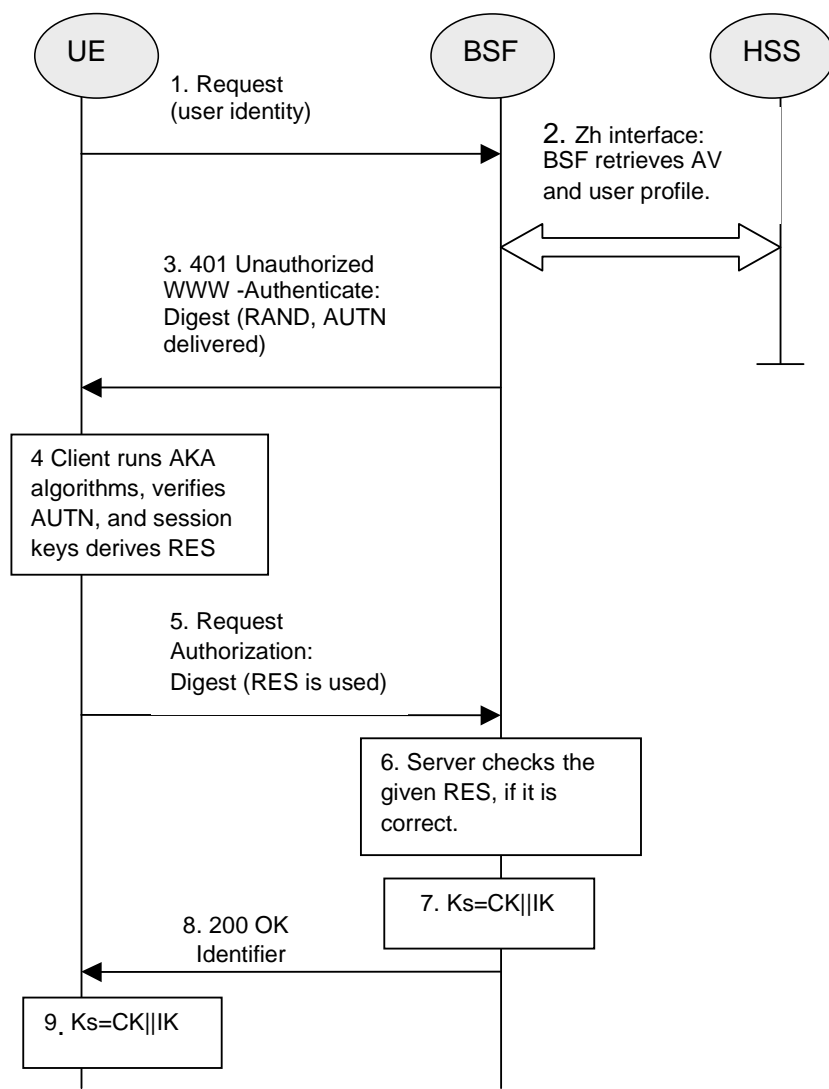
<b>Clauses affected:</b>	⌘ 4.5.2, 5.3.2										
<b>Other specs affected:</b>	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">Y</td> <td style="width: 20px; text-align: center;">N</td> </tr> <tr> <td style="text-align: center;">X</td> <td style="text-align: center;"></td> </tr> <tr> <td style="text-align: center;"></td> <td style="text-align: center;">X</td> </tr> <tr> <td style="text-align: center;"></td> <td style="text-align: center;">X</td> </tr> </table> Other core specifications Test specifications O&M Specifications	Y	N	X			X		X	⌘ TS 29.109	
Y	N										
X											
	X										
	X										
<b>Other comments:</b>	⌘ -										

\*\*\*\*\* **begin 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 Ks 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 Ks. The key material Ks 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.

\*\*\*\*\* **begin 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.

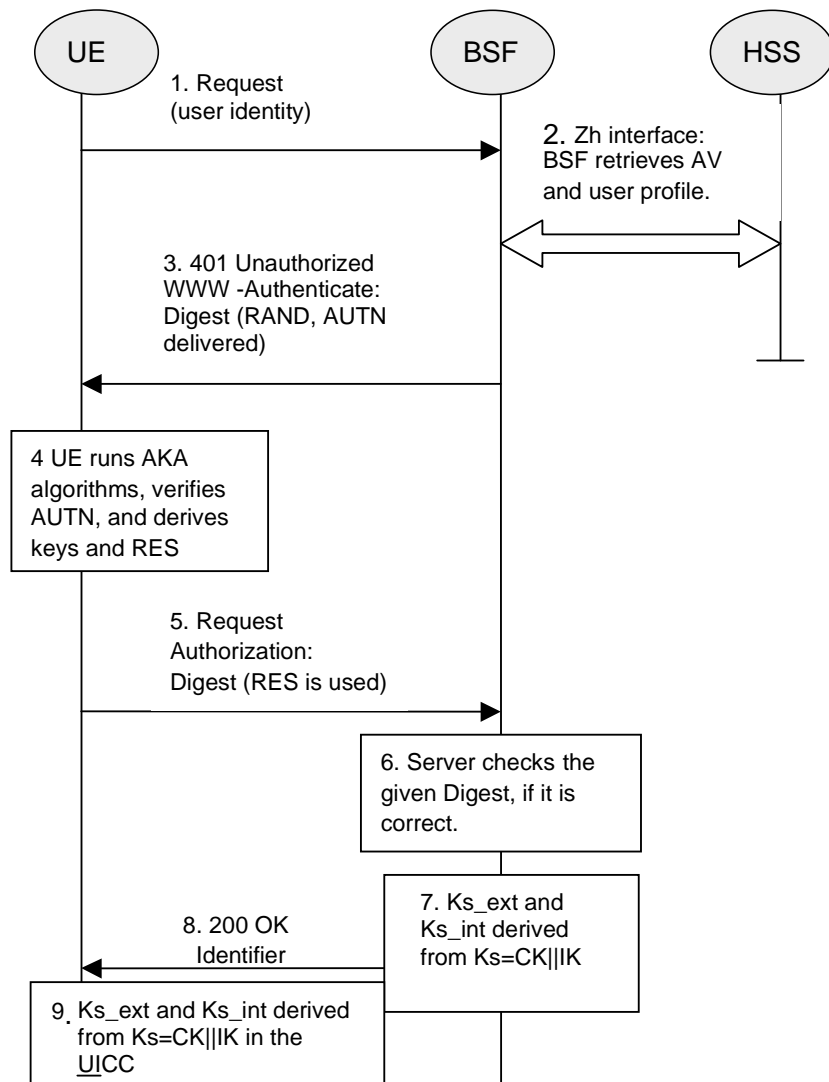


Figure 5.1: The bootstrapping procedure with UICC-based enhancements

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,  $AV = RAND || AUTN || XRES || CK || IK$ ) over the Zh reference point from the HSS. The BSF can then decide to perform GBA\_U, 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  $IK = IK1 || IK2$  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  $AUTN^* = SQN \oplus AK || AMF || MAC^*$ ) 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.  $SQN \oplus AK || AMF || MAC$ ) 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.  $h1(Ks, h1 \text{ key derivation parameters}) = Ks\_ext || Ks\_int$  (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.

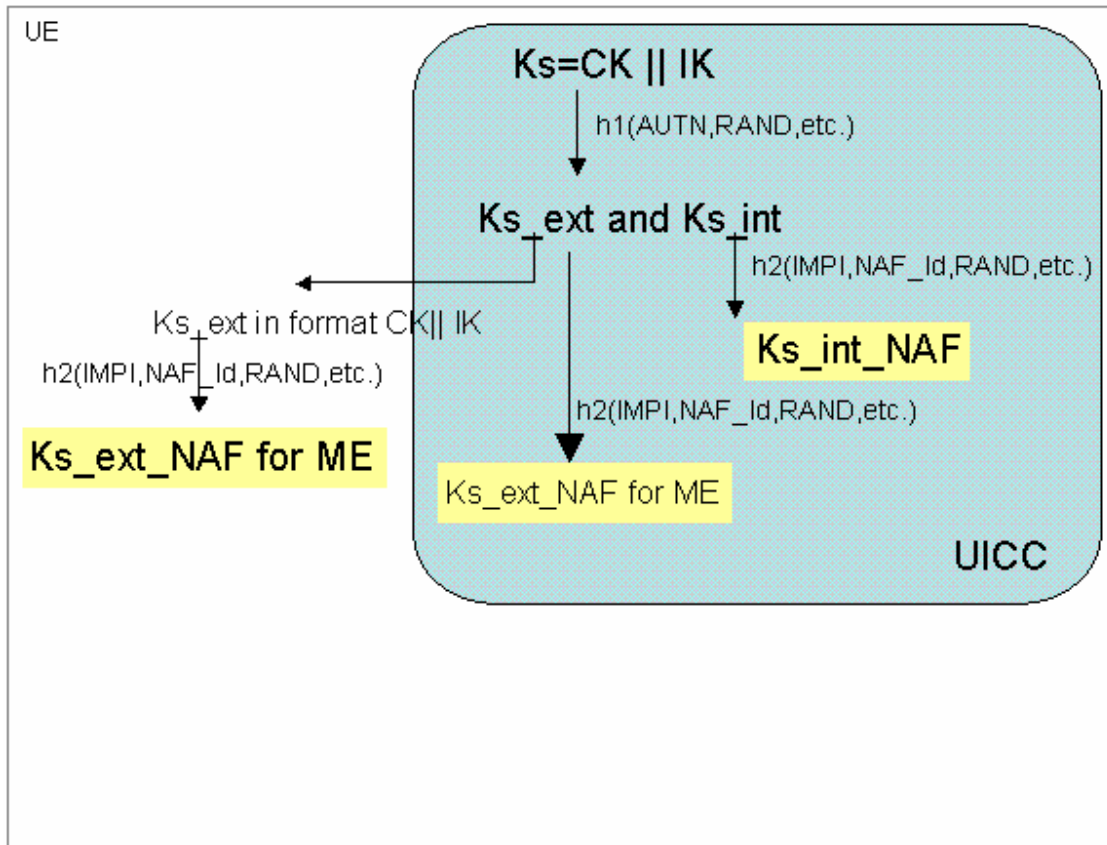


Figure 5.2: Key derivation for GBA-aware UICC when GBA-run was triggered

\*\*\*\*\* end change \*\*\*\*\*