

CR-Form-v7

CHANGE REQUEST

33.220 CR CRNum # rev - # 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:	#	B-TID generation	
Source:	#	Nokia	
Work item code:	#	GBA and SSC	Date: # 29/06/2004
Category:	#	C	Release: # Rel-6
		Use <u>one</u> of the following categories: <i>F</i> (correction) <i>A</i> (corresponds to a correction in an earlier release) <i>B</i> (addition of feature), <i>C</i> (functional modification of feature) <i>D</i> (editorial modification) Detailed explanations of the above categories can be found in 3GPP TR 21.900 .	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)

Reason for change:	#	Currently the B-TID is generated from RAND and BSF's hostname. Since RAND is in binary format this results to a binary B-TID. For example, in HTTP Digest based Ua reference point this is not usable, and an additional transformation to ascii format is needed. To avoid the extra transforms in Ua interface, the B-TID should be ascii based, and this is achieved by base64 encoding the RAND part of the B-TID: "base64encode(RAND)@BSF_servers_domain_name". Also, B-TID abbreviation is taken into use in the TS.
Summary of change:	#	B-TID is generated from base64 encoded RAND and BSF's hostname.
Consequences if not approved:	#	B-TID is in binary format, and its extra transformation to ascii format is needed in certain application protocols implementing the Ua reference point (such as HTTP Digest).

Clauses affected:	#	2, 3.1, 4.4.4, 4.4.6, 4.4.7, 4.5.2, 4.5.3, 5.3.2, 5.3.3										
Other specs affected:		<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>	Y	N	X			X		X	Other core specifications	# TS 24.109
	Y	N										
	X											
	X											
	X											
		Test specifications										
		O&M Specifications										
Other comments:	#											

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

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

- [1] 3GPP TS 31.102: "3rd Generation Partnership Project; Technical Specification Group Terminals; Characteristics of the USIM application".
- [2] 3GPP TS 33.102: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security architecture".
- [3] Franks J., et al.: "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.
- [4] A. Niemi, et al.: "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)", RFC 3310, September 2002.
- [5] 3GPP TS 33.221: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA); Support for Subscriber Certificates".
- [6] T. Dierks, et al.: "The TLS Protocol Version 1.0", RFC 2246, January 1999.
- [7] OMA: "Provisioning Content Version 1.1", Version 13-Aug-2003. Open Mobile Alliance.
- [8] 3GPP TS 23.228: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS); Stage 2 (Release 6)".
- [9] IETF RFC 3546 (2003): "Transport Layer Security (TLS) Extensions".
- [10] [IETF RFC 3548 \(2003\): "The Base16, Base32, and Base64 Data Encodings"](#).

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

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

Bootstrapping Server Function: BSF is hosted in a network element under the control of an MNO.

Editor's note: Definition to be completed.

ME-based GBA: in GBA_ME, all GBA-specific functions are carried out in the ME. The UICC is GBA-unaware. If the term GBA is used in this document without any further qualification then always GBA_ME is meant, see clause 4 of this specification.

UICC-based GBA: this is a GBA with UICC-based enhancement. In GBA_U, the GBA-specific functions are split between ME and UICC, see clause 5 of this specification.

Network Application Function: NAF is hosted in a network element under the control of an MNO.

~~Editor's note: Definition to be completed.~~

Bootstrapping Transaction Identifier: [the bootstrapping transaction identifier \(B-TID\) is used to bind the subscriber identity to the keying material in reference points Ua, Ub and Zn.](#)

~~Editor's note: Definition to be completed.~~

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

4.4.4 Requirements on reference point Ub

The requirements for reference point Ub are:

- the BSF shall be able to identify the UE;
- the BSF and the UE shall be able to authenticate each other based on AKA;
- the BSF shall be able to send a [bootstrapping T](#)transaction [I](#)identifier to the UE.

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

4.4.6 Requirements on reference point Zn

The requirements for reference point Zn are:

- mutual authentication, confidentiality and integrity shall be provided;

NOTE: This requirement may be fulfilled by physical or proprietary security measures if BSF and NAF are located within the same operator's network.

Editors' Note: In the visited NAF scenario, it should be decided how the communication between a D-Proxy and a BSF is secured. The possible solutions for securing this link include TLS and IPsec.

- The BSF shall verify that the requesting NAF is authorised;
- The NAF shall be able to send a key material request to the BSF, containing NAF's public hostname used by the UE's corresponding request. The BSF shall be able to verify that a NAF is authorized to use this hostname, i.e. the FQDN used by UE when it contacts the NAF;
- The BSF shall be able to send the requested key material to the NAF;
- The NAF shall be able to get the subscriber profile information needed for security purposes from BSF;
- The BSF shall be able to indicate to the NAF the lifetime of the key material.

Editor's note: Relationship between ~~B-TID~~Transaction Identifier and subscriber identity is ffs. In the case of Presence reference point Ut, there are several potential identities that are related to ~~Transaction Identifier~~B-TID, i.e. IMPI and IMPUs. The subscriber may have several Presence accounts related to same IMPI. ~~Transaction Identifier~~B-TID does not carry enough information on which IMPU the end-user is trying to use.

4.4.7 Requirements on Bootstrapping Transaction Identifier

Bootstrapping ~~Transaction identifier~~ (B-TID) shall be used to bind the subscriber identity to the keying material in reference points Ua, Ub and Zn.

Requirements for ~~Transaction Identifier~~B-TID are:

- ~~Transaction Identifier~~B-TID shall be globally unique;
- ~~Transaction Identifier~~B-TID shall be usable as a key identifier in protocols used in the reference point Ua;
- NAF shall be able to detect the home network and the BSF of the UE from the ~~Transaction Identifier~~B-TID.

NOTE 1: NAF can remove the security association based on deletion conditions after the key has become invalid.

NOTE 2: Care has to be taken that the parallel use of GBA and non-GBA authentication between UE and NAF does not lead to conflicts, e.g. in the name space. This potential conflict cannot be resolved in a generic way as it is dependent on specific protocol and authentication mechanism used between UE and application server. It is therefore out of scope of this specification. For the example of HTTP Digest authentication used between UE and NAF, parallel use is possible as the following applies: <username,password>-pairs must be unique to one realm only. As the NAF controls the realm names, it has to ensure that only the GBA based realm is named with the reserved 3GPP realm name. In the special case that the NAF wants to allow non GBA based authentication in the GBA realm also, it has to ensure that no usernames in the format of a ~~Transaction Identifier~~B-TID are used outside GBA based authentication.

===== BEGIN NEXT 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.

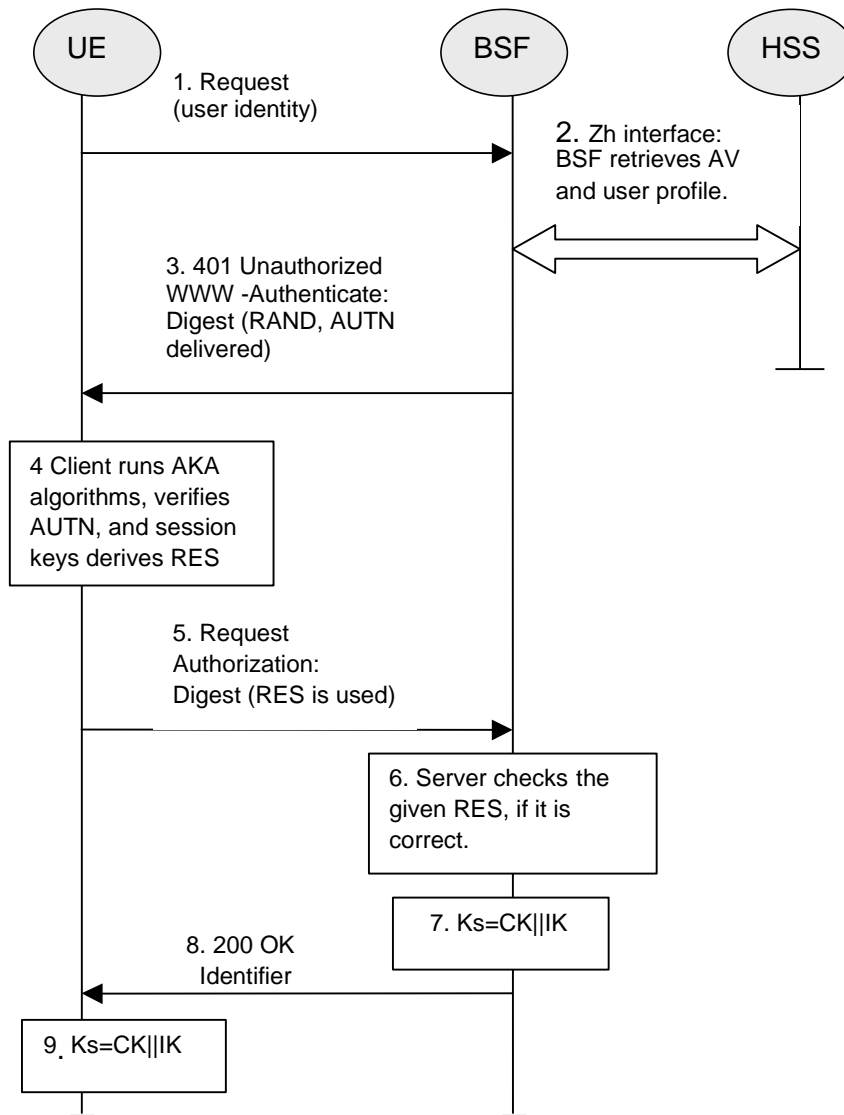


Figure 4.3: The bootstrapping procedure

1. The UE sends an HTTP request towards the BSF.
2. BSF retrieves the user profile 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 ~~Transaction Identifier~~**B-TID** value shall be also generated in format of NAI by taking the [base64 encoded \[10\]](#) 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 ~~Transaction Identifier~~**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 ~~Transaction Identifier~~ B-TID for further use, until the lifetime of Ks has expired, or until the key Ks is updated.

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 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 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 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 section 4.5.2). For each protocol used over Ua it shall be specified if only cases (1) and (2) of NOTE 2 of section 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 ~~Transaction Identifier~~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;

NAF starts communication over reference point Zn with BSF

- The NAF requests key material corresponding to the ~~Transaction Identifier~~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);
- 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;
- 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 ~~Transaction Identifier~~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.

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.

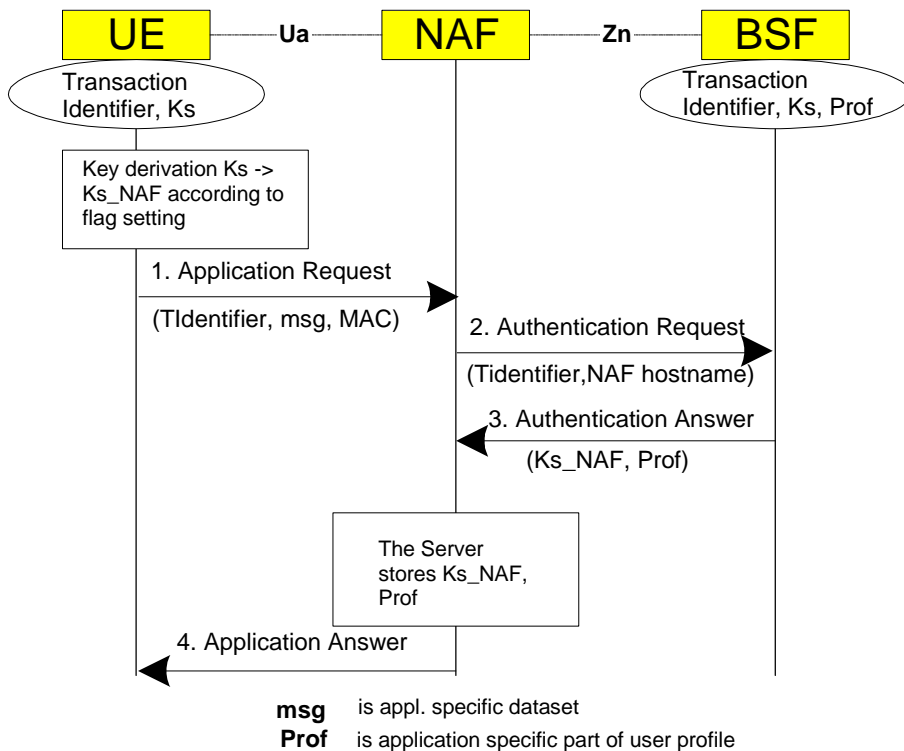


Figure 4.4: The bootstrapping usage procedure

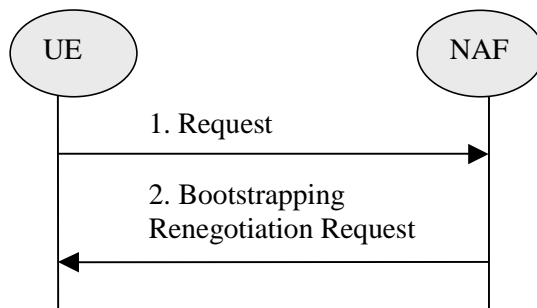


Figure 4.5: Bootstrapping renegotiation request

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

5.3.2 Bootstrapping procedure

The procedure specified in this clause differs from the procedure specified clause 4.5.2 in the generation of the Authentication Vector in the HSS and the local handling of keys 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.

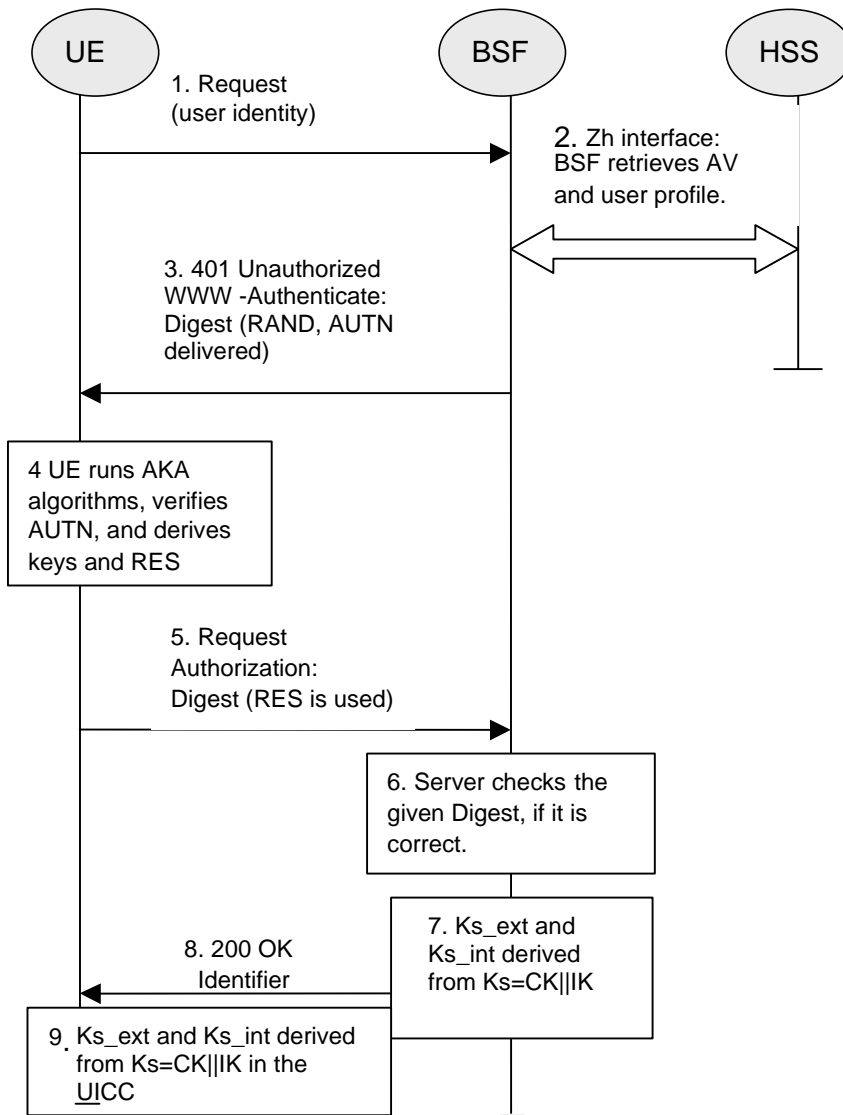


Figure 5.1: The bootstrapping procedure with UICC-based enhancements

1. The ME sends an HTTP request towards the BSF.
2. The BSF retrieves the user profile 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 HSS recognises that the UICC is GBA_U aware and that the request for AVs came from a GBA_U aware BSF, and generates a GBA_U-AV. If the BSF received GBA_U-AVs then it stores the XRES after flipping the least significant bit.

Editors' Note: The GBA_U-AV will be described within Annex D of this specification.

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 ME sends RAND and AUTN to the UICC. The UICC checks AUTN to verify that the challenge is from an authorised network; the UICC also calculates CK, IK and RES. This will result in session keys CK and IK in both BSF and UICC.
5. The UICC checks if a GBA_U-AV was received as specified in step 2 of this clause. If this is not the case, the UICC transfers RES, CK and IK to the ME, and the ME proceeds according to the procedures specified in section 4 of this document, without involving the UICC any further. If a GBA_U-AV was received, 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 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. The BSF checks if the AV was a GBA_U- AV as specified in step 2 of this clause. If this is not the case, the BSF applies the procedures specified in clause 4 of this document. If the GBA_U-AV was recognized 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 ~~Transaction Identifier~~**B-TID** value shall be also generated in format of NAI by taking the [base64 encoded \[10\]](#) 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 ~~Transaction Identifier~~**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 ~~Transaction Identifier~~**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.

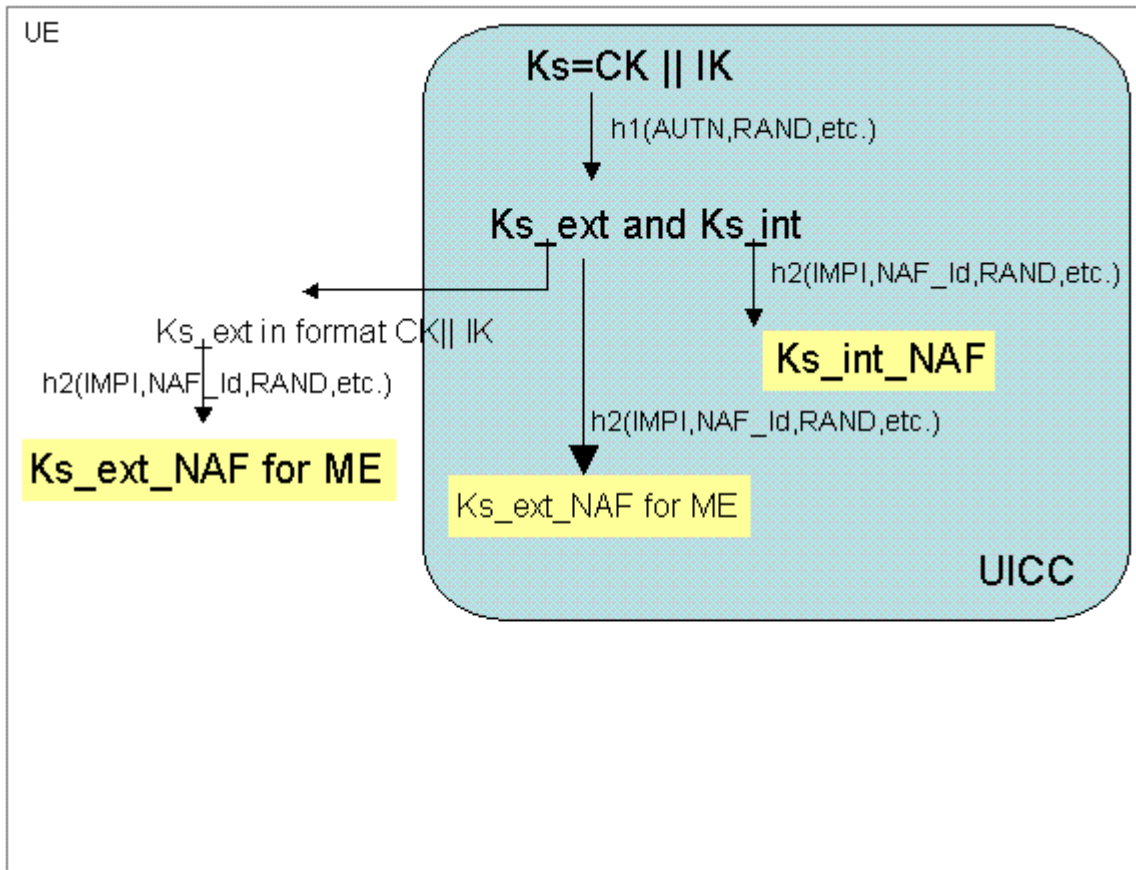


Figure 5.2: Key derivation for GBA-aware UICC when GBA-run was triggered

===== 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, 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. How this agreement is reached is application-specific and is not within the scope of this document.

NOTE 1: Such an agreement could e.g. be reached by manual configuration, or by an application-specific protocol step.

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} 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 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 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 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.

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 ~~Transaction Identifier~~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 ~~transaction identifier~~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 ~~transaction identifiers~~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 ~~Transaction Identifier~~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).
- 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 ~~Transaction Identifier~~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 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.

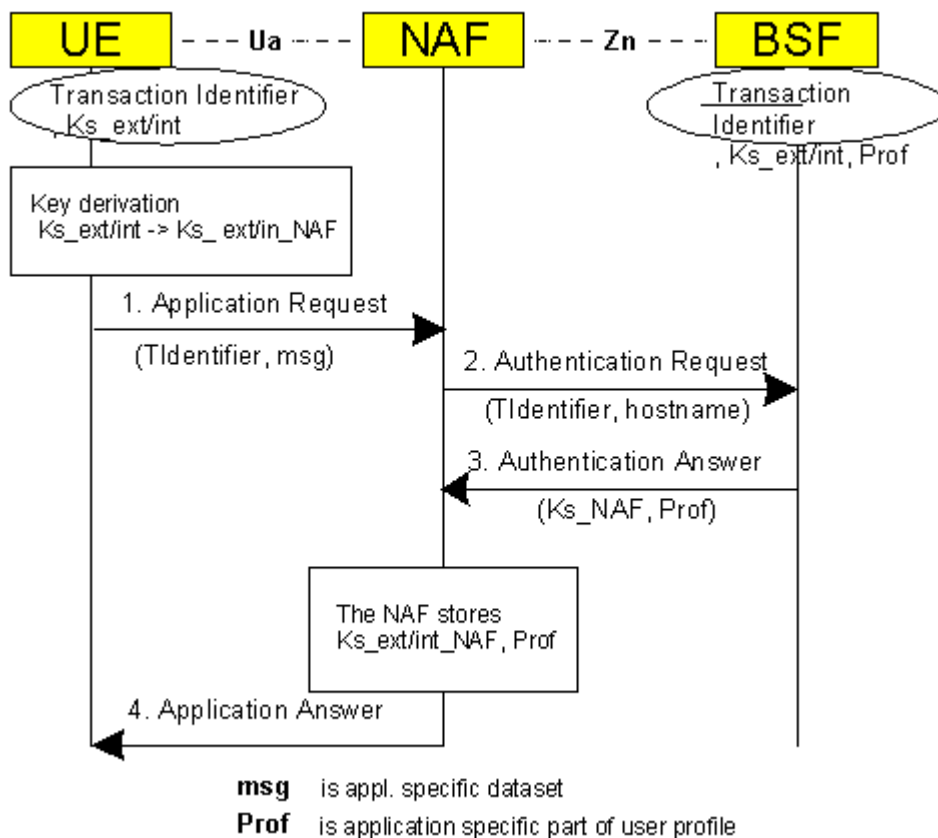


Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

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