Source: **SA WG3**

Title: **17 CRs to 33.220: (Rel-6)**

Document for: **Approval**

Agenda Item: **7.3.3**

The following CRs have been agreed by SA WG3 and are presented to TSG SA for approval.

TSG SA Doc number	Spec	CR	Rev	Phase	Subject	Cat	Version-Current	SA WG3 Doc number	Work item
SP-040855	33.220	018	1	Rel-6	BSF discovery using default domain method	C	6.2.0	S3-040831	SEC1-SC
SP-040855	33.220	019	1	Rel-6	Local validity condition set by NAF	F	6.2.0	S3-040828	SEC1-SC
SP-040855	33.220	020	3	Rel-6	GBA User Security Settings (GUSS) usage in GAA and Introduction of NAF groups	C	6.2.0	S3-041135	SEC1-SC
SP-040855	33.220	021	2	Rel-6	Details of USIM/ISIM selection in GAA	C	6.2.0	S3-041085	SEC1-SC
SP-040855	33.220	023	-	Rel-6	TLS profile for securing Zn' reference point	C	6.2.0	S3-040756	SEC1-SC
SP-040855	33.220	025	2	Rel-6	Optimization of the GBA_U key derivation procedure	C	6.2.0	S3-041136	SEC1-SC
SP-040855	33.220	027	2	Rel-6	Requirement on ME capabilities for GBA_U	B	6.2.0	S3-041080	SEC1-SC
SP-040855	33.220	034	1	Rel-6	Adding a note about replay protection	F	6.2.0	S3-041087	SEC1-SC
SP-040855	33.220	035	1	Rel-6	Complete the MAC modification for GBA_U	F	6.2.0	S3-041078	SEC1-SC
SP-040855	33.220	036	1	Rel-6	Removal of unnecessary editor's notes	D	6.2.0	S3-041082	SEC1-SC
SP-040855	33.220	038	1	Rel-6	Fetching of one AV only on each Zh run between BSF and HSS	C	6.2.0	S3-041090	SEC1-SC
SP-040855	33.220	039	1	Rel-6	Clean up of TS 33.220	F	6.2.0	S3-041083	SEC1-SC
SP-040855	33.220	040	1	Rel-6	New key management for ME based GBA keys	C	6.2.0	S3-041084	SEC1-SC
SP-040855	33.220	041	1	Rel-6	Key derivation function	B	6.2.0	S3-041081	SEC1-SC
SP-040855	33.220	042	1	Rel-6	Re-negotiation of keys	F	6.2.0	S3-041140	SEC1-SC
SP-040855	33.220	043	1	Rel-6	No GUSS/USS update procedures in Release-6	C	6.1.0	S3-041089	GBA-SSC
SP-040855	33.220	044	1	Rel-6	Clarify the number of NAF-specific keys stored in the UE per NAF-Id	D	6.1.0	S3-041137	SEC1-SC

3GPP TSG-SA WG3 Meeting S3#35
St Paul's Bay, Malta, October 5-8, 2004

Tdoc ⌘S3-040831

CR-Form-v7.1

CHANGE REQUEST

⌘ **33.220** **CR** **018** ⌘ **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:	⌘	BSF discovery using default domain method
Source:	⌘	SA WG3
Work item code:	⌘	SEC1-SC
Date:	⌘	06/09/2004
Category:	⌘	**C**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
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)

Reason for change:	⌘	BSF discovery using default domain method is added.
Summary of change:	⌘	The BSF address is derived from either IMSI or IMPI depending on the UICC application that was used in the bootstrapping. The old service discovery methods are deleted.
Consequences if not approved:	⌘	

Clauses affected:	⌘	4.5.4

Other specs affected:	⌘	Y	N		⌘	
		X		Other core specifications		24.109
			X	Test specifications		
			X	O&M Specifications		

Other comments: ⌘

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

4.5.4 Procedure related to service discovery

~~To enable the bootstrapping procedure, a procedure needs to be described on how to discover the location of BSF. It shall be possible to enable the terminal to be configured either manually or automatically via one of the following approaches:~~

The UE shall discover the address of the BSF the from the identity information related to the UICC application that is used during bootstrapping procedure, i.e., IMSI for USIM, or IMPI for ISIM the following way:

- In the case where the USIM is used in bootstrapping, the address information shall be derived as follows:

  1. take the first 5 or 6 digits, depending on whether a 2 or 3 digit MNC is used (see 3GPP TS 31.102 [1]) and separate them into MCC and MNC; if the MNC is 2 digits then a zero shall be added at the beginning;
  2. use the MCC and MNC derived in step 1 to create the "mnc<MNC>.mcc<MCC>.3gppnetwork.org" domain name;
  3. add the label "bsf." to the beginning of the domain.

  Example 1: If IMSI in use is "234150999999999", where MCC=234, MNC=15, and MSIN=0999999999, the BSF address would be "bsf.mnc015.mcc234.3gppnetwork.org".

- In the case where ISIM is used in bootstrapping, the address information shall be derived as follows:

  1. extract the domain name from the IMPI;
  2. add the label "bsf." to the beginning of the domain.

  Example 2: If the IMPI in use is "user@operator.com", the BSF address would be "bsf.operator.com ".

- ~~The address information shall be published via reliable channel. Subscribers shall store all the parameters as part of the initial establishment of IP connectivity. The addresses need to be input only once;~~

- ~~The address information shall be pushed automatically to the UE over the air interface when the subscription to bootstrapping service is accepted. All the parameters shall be saved in the UE and used the same manner as above. The procedure is specified in [7];~~

- ~~The location information shall be discovered automatically based on DHCP, after the IP connectivity has been established. The DHCP server shall provide the UE with the domain name of a BSF and the address of a Domain Name Server (DNS) that is capable of resolving the Fully Qualified Domain Name (FQDN) of the BSF. The procedure is specified in TS 23.228 [8].~~

~~NOTE: The location of DHCP server may be pushed to UE through the procedure specified in [7].~~

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

3GPP TSG-SA WG3 SECURITY Meeting #35
St Paulís Bay, Malta, 5-8 October 2004

Tdoc ⌘ S3-040828

CR-Form-v7.1

CHANGE REQUEST

⌘ **33.220** **CR** **019** ⌘ **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.*

St Paulís Bay, Malta

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Local validity condition set by NAF
Source:	⌘	SA WG3
Work item code:	⌘	SEC1-SC
Date:	⌘	7/10/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
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)*

Reason for change:	⌘	The key leaking out is an potential security threaten during the service, e.g. leakage of key with not well safed local connections in UE, then a frequent stolen service request may happen with a very short time. Application(NAF) should have a mechanism to limit the abnormal using of the shared secret and take action to mitigate it as much as possible.
Summary of change:	⌘	NAF set the local validity condition of the shared key material. E.g. a limited number of times that TID and key material can be used. When NAF receive the userís request including the TID, the NAF can check the local validity conditions set by itself to avoid the service stolen.
Consequences if not approved:	⌘	The NAF miss the important feature that can avoid the some possible attacks

Clauses affected:	⌘	4.2.2, 4.5.3, 5.3.3
Other specs affected:	⌘	Y N X Other core specifications X Test specifications X O&M Specifications
	⌘	
Other comments:	⌘	

**********Begin of change **********

4.2.2 Network application function (NAF)

After the bootstrapping has been completed, the UE and an operator-controlled NAF can run some application specific protocol where the authentication of messages will be based on those session keys generated during the mutual authentication between UE and BSF.

General assumptions for the functionality of an operator-controlled NAF are:

- there is no previous security association between the UE and the NAF;
- NAF shall be able to locate and communicate securely with the subscriber's BSF;
- NAF shall be able to acquire a shared key material established between UE and the BSF during the run of the application-specific protocol;
- NAF shall be able to acquire an (application-specific) user security setting from the HSS via the BSF;
- NAF shall be able to set the local validity condition of the shared key material according to the local policy;
- NAF shall be able to check lifetime and local validity condition of the shared key material.

**********End of change **********

**********Begin of 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 keyís lifetime has expired or the key can not meet the NAF local validity condition, 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;

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 NAFís 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. ~~The NAF can further set the local validity condition of the Ks_NAF according to the local policy, for example a limitation of reuse times of a Ks_NAF.~~ 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:The NAF can further set the local validity condition of the Ks_NAF according to the local policy,for example a limitation of reuse times of a Ks_NAF.

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.

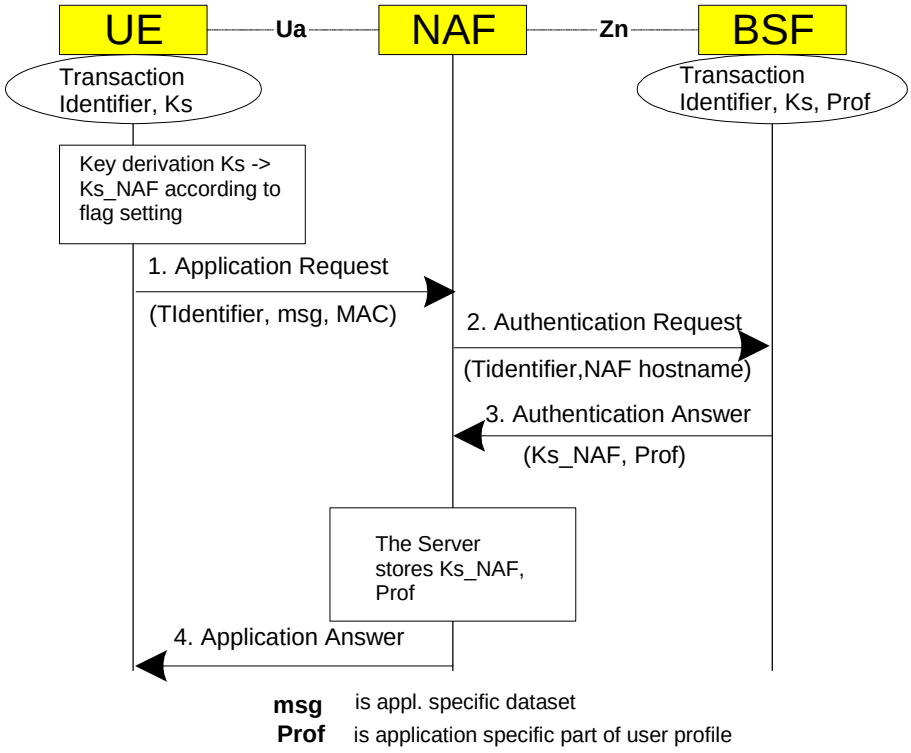


Figure 4.4: The bootstrapping usage procedure

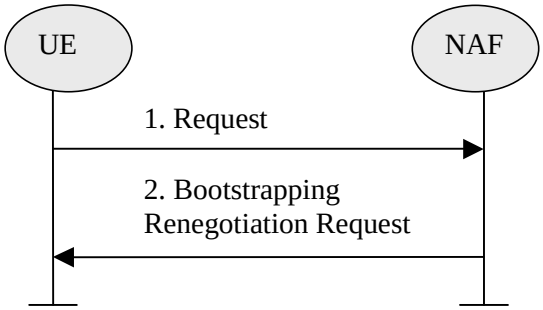


Figure 4.5: Bootstrapping renegotiation request

**********End of change **********

**********Begin of 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.

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.

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 NAFís 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 can further set the local validity condition of the Ks_NAF according to the local policy, for example a limitation of reuse times of a Ks_NAF.

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 BSFís 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.

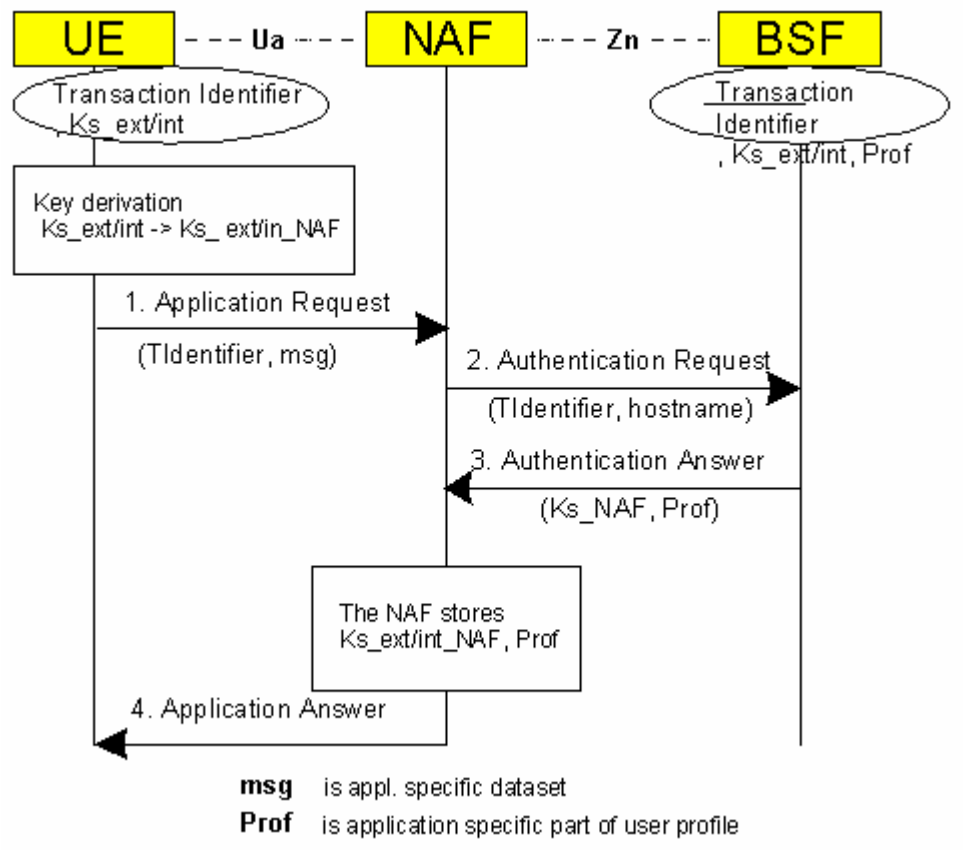


Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

**********End of change **********

3GPP TSG-SA WG3 Meeting S3#36
Shenzhen, China, November 23-26, 2004

Tdoc ⌘S3-041135

CR-Form-v7.1

CHANGE REQUEST

⌘ **33.220** **CR** **020** ⌘ **rev** **3** ⌘ 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 [X]

Title:	⌘	GBA User Security Settings (GUSS) usage in GAA and Introduction of NAF groups
Source:	⌘	SA WG3
Work item code:	⌘	SEC1-SC
Date:	⌘	12/11/2004
Category:	⌘	**C**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
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)*

Reason for change:	⌘	The details of GBA user security settings (GUSS) are used is missing. - The MNO may have a need to supply different USSs to NAFs for the same service, dependent on particular NAF, e.g. if located in home or visited network. For this purpose NAF groups are introduced.
Summary of change:	⌘	- The BSF may require that one or more USSs shall be present in subscriber's GUSS for a particular NAF. If one or more of these required USSs are missing from the GUSS, the BSF will not provide bootstrapping information to the NAF. (This method is used for the home operator control on whether the subscriber may use service in the visited network, i.e, visited NAF.) - If a NAF requests USSs from the BSF and they are not present in user GBA user security settings, it will not cause an error, provided the conditions of the local policy of the BSF are fulfilled. The BSF will then send only the requested and found USSs to the NAF. - GUSS may be used to transfer subscriber specific parameters intended for the BSF only (i.e., the type of subscriber's UICC and subscriber specific key lifetime). - The complete set of application-specific user security settings are named GUSS and application-specific user security setting are named USS in the specification for clarity reasons. - NAF grouping for better controllability of GBA usage is introduced.
Consequences if not approved:	⌘	The details of how GBA user security settings (GUSS) are used are missing. Difficult handling of varying access controls on GBA usage.

Clauses affected:	⌘	3.1, 3.2, 4.2.1, 4.2.2, 4.2.3, 4.4.3, 4.4.6, 4.5.3, 5.3.3
Other specs affected:	⌘	Y N X Other core specifications ⌘ TS 29.109 CR009 X Test specifications X O&M Specifications
Other comments:	⌘	

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

3.1 Definitions

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

Application: In all places in this document where the term application is used to refer to a service offered by the MNO or a third party to the mobile subscriber, then it always denotes the type of application and not the actual instance of an application installed on an application server.

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.

~~**GBA** **User Security Setting:** An application specific parameter set describing the security related usage of bootstrapping function by the BSF and, optionally, some types of NAFs in the context of an application and in relation to a subscriber. A user security setting~~ A USS is an application and subscriber specific parameter set that defines~~has~~ two parts, an authentication part, which contains the list of identities of the user needed for the application (e.g. IMPI, IMPUs, MSISDN, pseudonyms), and an authorisation part, which contains the user permission flags (e.g. access to application allowed, type of certificates which may be issued). Sometimes also called application-specific user security setting. The USS is delivered to the BSF as a part of GUSS from the HSS, and from the BSF to the NAF if requested by the NAF.

GBA User Security Settings: GUSS contains the BSF specific information element and the set of all application-specific ~~user security settings~~USSs.

NAF Group: A grouping of NAFs to allow assignment of different USSs to NAFs representing the same application. This grouping is done in each home network separately, i.e. one NAF contacting BSFs in different home networks belongs to different groups in every home network.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AK	Anonymity Key
AKA	Authentication and Key Agreement
B-TID	Bootstrapping Transaction Identifier
BSF	Bootstrapping Server Function
CA	Certificate Authority
FQDN	Fully Qualified Domain Name
GAA	Generic Authentication Architecture
GBA	Generic Bootstrapping Architecture
GBA_ME	ME-based GBA
GBA_U	GBA with UICC-based enhancements
GUSS	GBA User Security Settings
HSS	Home Subscriber System
IK	Integrity Key
KDF	Key Derivation Function
Ks_int	Derived key in GBA_U which remains on UICC
Ks_ext	Derived key in GBA_U
MNO	Mobile Network Operator
NAF	Network Application Function
PKI	Public Key Infrastructure
USS	~~GBA~~ User Security Setting

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

4.2.1 Bootstrapping server function (BSF)

A generic Bootstrapping Server Function (BSF) and the UE shall mutually authenticate using the AKA protocol, and agree on session keys that are afterwards applied between UE and a~~n operator controlled~~ Network Application Function (NAF). The BSF shall restrict the applicability of the key material to a specific NAF by using a suitable key derivation procedure. The key derivation procedure may be used with multiple NAFs during the lifetime of the key material. The lifetime of the key material is set according to the local policy of the BSF. The generation of key material is specified in clause 4.5.2.

The BSF shall be able to acquire the GBA user security settings (GUSS) from the HSS.

The BSF shall be able to keep a list, which assigns NAFs to NAF Groups. This list is used to select if any and which application-specific USS within GUSS is valid for a certain NAF.

NOTE 1: The operator does the assignment of NAFs to NAF Groups. NAF Group definitions in HSS and all connected BSFs belonging to the same operator's network shall be equal (cf., clause 4.2.3). As these network elements belong to the same operator's network, standardisation of the NAF Group definitions themselves is not necessary in 3GPP.

NOTE 2: The NAF grouping may be e.g. "home" and "visited". It allows the BSF to send USSs for the same application with e.g. different authorization flags to different NAFs, e.g., in home network and visited networks. The NAF e.g. in visited network indicates only the requested application, but it is unaware of the grouping in home network of the subscriber.

4.2.2 Network application function (NAF)

After the bootstrapping has been completed, the UE and a~~n operator controlled~~ NAF can run some application specific protocol where the authentication of messages will be based on those session keys generated during the mutual authentication between UE and BSF.

General assumptions for the functionality of a~~n operator controlled~~ NAF are:

- there is no previous security association between the UE and the NAF;
- NAF shall be able to locate and communicate securely with the subscriber's BSF;
- NAF shall be able to acquire a shared key material established between UE and the BSF during the run of the application-specific protocol;
- NAF shall be able to acquire ~~an~~zero or more ~~(~~application-specific~~) user security setting~~ USSs from the HSS via the BSF;
- NAF shall be able to check lifetime of the shared key material.

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

4.2.3 HSS

The set of all user security settings (USSs), i.e. GUSS, is stored in the HSS. ~~There shall be at most one USS per application stored in the HSS.~~ In the case where the subscriber has multiple subscriptions, i.e. multiple ISIM or USIM applications on the UICC, the HSS shall contain one or more ~~subscriber profiles~~GUSSs that can be mapped to one or more private identities, i.e. IMPIs and IMSIs.

~~Editor's note: Needed new subscriber profile parameters, i.e. GBA user security settings, are FFS.~~

The requirements on the HSS are:

- HSS shall provide the only persistent storage for G~~BA~~ USSs;
- G~~BA~~ USS shall be defined in such a way that interworking of different operators for standardised application profiles is possible;
- G~~BA~~ USS shall be defined in such a way that profiles for operator specific applications and extensions to existing application profiles are supported without need for standardisation of these elements.
- GUSS shall be able to contain application-specific USSs that contain parameters that are related to identification or authorization information of one or more applications hosted by one ore more NAFs. Any other types of parameters are not allowed in the application-specific USS.

NOTE 1: The necessary subscriber profile data may be fetched by the NAF directly from HSS or from its local database using identity information provided by the application-specific USS.

NOTE 2: The HSS may temporarily remove an application-specific USS from the GUSS if the service is temporarily revoked from the subscriber.

- GUSS shall be able to contain parameters intended for the BSF usage:
  - the type of the UICC the subscriber is issued (i.e., is it GBA_U aware or not, cf. subclause 5);
  - subscriber specific key lifetime.

NOTE 3: These parameters are optional and if they are missing from subscriber's GUSS or subsciber does not have GUSS then the BSF will use the default values in the BSF local policy defined by the particular MNO.

- HSS shall be able to assign application-specific USSs to a NAF Group. This shall be defined in such a way that different USSs for the same application, but for different groups of NAFs, are possible. The restrictions on the number of USSs per GUSS are dependent on the usage of NAF Groups by the operator:
  - if no NAF Groups are defined for this application then at most one USS per application is stored in GUSS;

- if NAF Groups are defined for this application then at most one USS per application and NAF Group is stored in GUSS.

- NAF Group definitions in the HSS and all connected BSFs belonging to the same operator's network shall be equal.

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

4.4.3 Roaming

The requirements on roaming are:

- The roaming subscriber shall be able to utilize the bootstrapping function in the home network. The subscriber shall be able to utilize network application function that is in a visited network.

- The home network shall be able to control whether its subscriber is authorized to use the service in the visited network.

===== 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;

- If the BSF and the NAF are located within the same operator's network, the Zn reference point shall be secured according to NDS/IP [13];

- If the BSF and the NAF are located in different operators' networks, the Zn' reference point between the D-Proxy and the BSF shall be secured using TLS as specified in RFC 2246 [6];

Editorís Note: The TLS Certificate profiling needs to be completed and will be added into an Annex.

- 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 a selected set of application-specific ~~user security settings~~USSs from the BSF, depending on the policy of the BSF and the application indicated in the request from the NAF over Zn;

- The NAF shall be able to indicate to the BSF the single application or several applications it requires ~~user security settings~~USS for;

NOTE 1: If some application needs only a subset of an application-specific ~~user security setting~~USS, e.g. only one IMPU, the NAF selects this subset from the complete set of ~~user security settings~~USSs sent from BSF.

- If a NAF requests USSs from the BSF and they are not present in subscriber's GUSS, it shall not cause an error, provided the conditions of the local policy of the BSF are fulfilled. The BSF shall then send only the requested and found USSs to the NAF;

- The BSF shall be able to be configured on a per NAF or per application basis if private subscriber identity and which ~~user security settings~~application-specific USSs may be sent to a NAF;

- The BSF shall be able to be configured locally by the MNO in such a way that the BSF is able to decide on a per NAF basis if one or more application-specific USSs shall be present in subscriber's GUSS, and to reject the request from the NAF in case the conditions are not fulfilled;

- The BSF shall be able to indicate to the NAF the lifetime of the key material. The key lifetime sent by the BSF over Zn shall indicate the expiry time of the key, and shall be identical to the key lifetime sent by the BSF to the UE over Ub.

NOTE 2: This does not preclude a NAF to refresh the key before the expiry time according to the NAF's local policy.

Editor's note: It is ffs which actions are to be taken over Zn when the BSF receives a user security settings update from the HSS over Zh.

===== BEGIN 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.

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 keyís 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;

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 one or more application-specific ~~user security settings~~USSs for the applications, which the request received over Ua from UE may access;

- With the key material request, the NAF shall supply NAFís 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, and the requested application-specific and potentially NAF group specific USSs if they are available in subscriber's GUSS and if the NAF is authorized to receive the requested USSs. 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 require that one or more application-specific and potentially NAF group specific USSs shall be present in subscriber's GUSS for the NAF (cf. subclause 4.4.6). If one or more of these required settings are missing from the GUSS, the BSF shall indicate this in the reply to the NAF.

- The BSF may also send the private user identity (IMPI) and requested ~~user security settings~~USSs 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.

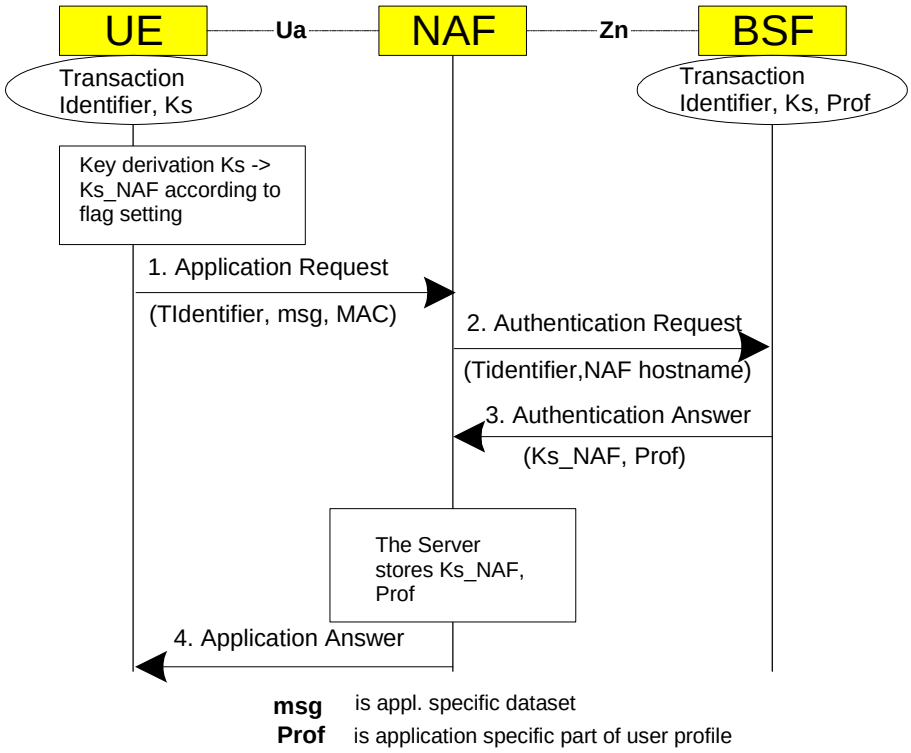


Figure 4.4: The bootstrapping usage procedure

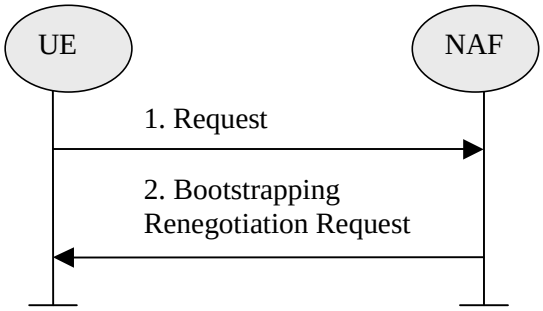


Figure 4.5: Bootstrapping renegotiation request

===== 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 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.

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.

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 one or more application-specific ~~user security settings~~USSs 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 NAFís 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, and the requested application-specific and potentially NAF group specific USSs if they are available in subscriber's GUSS and if the NAF is authorized to receive the requsted USSs. 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 9: 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 require that one or more application-specific and potentially NAF group specific USSs shall be present in subscriber's GUSS for the NAF (cf. subclause 4.4.6). If one or more of these required settings are missing from the GUSS, the BSF shall indicate this in the reply to the NAF.

- The BSF may also send the private user identity (IMPI) and requested ~~user security settings~~USSs to NAF according to the BSFís 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.

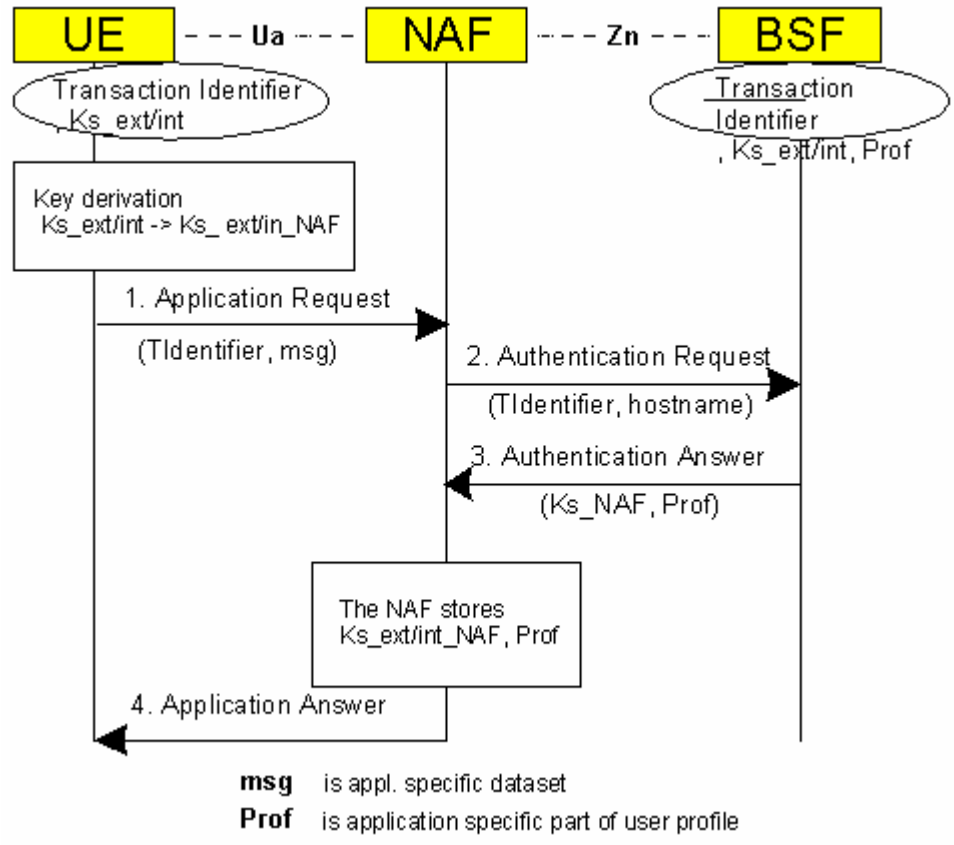


Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

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

3GPP TSG-SA WG3 Meeting S3#36
Shenzhen, China, November 23-26, 2004

Tdoc ⌘S3-041085

CR-Form-v7

CHANGE REQUEST

⌘ **33.220** **CR** **021** ⌘**rev** **2** ⌘ 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:	⌘	Details of USIM/ISIM selection in GAA
Source:	⌘	SA WG3
Work item code:	⌘	SEC1-SC
Date:	⌘	16/11/2004
Category:	⌘	**C**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one 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:	⌘	At SA3#34 a new section 4.4.8 of TS 33.220 dealing with selection of UICC application for GBA was introduced (approved CR S3-040648). This document points to a necessary correction and, in addition, proposes improvements to the selection process as defined at SA3#34.
Summary of change:	⌘	The correction concerns the fact that a ìdefault USIMî is not defined in 3GPP specifications, and that the term ìselectionî is used in a way not compatible with other 3G specifications. The two main goals of the improvements are (i) the optional possibility for a Ua application to choose a particular UICC application (not only UICC type) and (ii) more deterministic behaviour and better understandability of the selection process by the user. Requirement regarding to name of the UICC application as an indication is added to subclause 4.2.4.
Consequences if not approved:	⌘	Specification stays inconsistent with regard to the corrections. Sub-optimal behaviour of UICC application selection.

Clauses affected:	⌘	2, 3.1, 4.2.4, 4.4.8

Other specs affected:	⌘	Y	N		⌘
			X	Other core specifications	
			X	Test specifications	
			X	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] 3GPP TS 31.103: "3rd Generation Partnership Project; Technical Specification Group Terminals; Characteristics of the IP Multimedia Services Identity Module (ISIM) application".

[11] 3GPP TS 23.003: "3rd Generation Partnership Project; Technical Specification Group Core Network; Numbering, addressing and identification".

[12] IETF RFC 3548 (2003): "The Base16, Base32, and Base64 Data Encodings".

[13] 3GPP TS 33.210: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Network domain security; IP network layer security".

[14] IETF RFC 3588 (2003): "Diameter Base Protocol".

[15] 3GPP TS 31.101: "3rd Generation Partnership Project; Technical Specification Group Terminals; UICC-terminal interface; Physical and logical characteristics ".

[16] 3GPP TS 33.203: ì3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G security; Access security for IP-based services (Release 6)î

===== 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.

GBA User Security Setting: An application-specific parameter set describing the security related usage of bootstrapping function by the BSF and, optionally, some types of NAFs in the context of an application and in relation to a subscriber. A user security setting has two parts, an authentication part, which contains the list of identities of the user needed for the application (e.g. IMPI, IMPUs, MSISDN, pseudonyms), and an authorisation part, which contains the user permission flags (e.g. access to application allowed, type of certificates which may be issued). Sometimes also called application-specific user security setting.

GBA User Security Settings: the set of all application-specific user security settings.

Bootstrapping Usage Procedure: A procedure using bootstrapped security association over Ua reference point.

Ua Application: An application on the ME intended to run bootstrapping usage procedure with a NAF.

GBA Function: A function on the ME executing the bootstrapping procedure with BSF (i.e. supporting the Ub reference point) and providing Ua applications with security association to run bootstrapping usage procedure. GBA function is called by a Ua application when a Ua application wants to use bootstrapped security association.

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

4.2.4 UE

The required functionalities from the UE are:

- the support of HTTP Digest AKA protocol;
- the capability to use both a USIM and an ISIM in bootstrapping;
- the capability to select either a USIM or an ISIM to be used in bootstrapping, when both of them are present;
- the capability for an Ua application on the ME ~~using the shared secret~~ to indicate to the GBA Function on the ME the type or the name of UICC application to use in bootstrapping (~~i.e., ISIM or USIM~~cf. subclause 4.4.8);
- the capability to derive new key material to be used with the protocol over Ua interface from CK and IK;
- support of NAF-specific application protocol (For an example see TS 33.221 [5]).

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

4.4.8 Requirements on selection of UICC application and related keys

When several applications are present on the UICC, which are capable of running AKA, then the ME shall ~~select~~choose one of these UICC applications for performing the GBA procedures specified in this document in the following order of preference:

1. The UE determines which UICC application is to be involved:

   a. the application on the ME that needs Ks_NAF (Ua application) may indicate to the GBA ~~application~~ support function (GBA function) the type or the name of the UICC application: no preference, USIM, ~~or~~ ISIM, or the ì Labelî (see definition in TS 31.101 [15]) of the UICC application.

   If the application on the ME indicated a ì Labelî of the UICC application, step b below shall be executed.

   If the application on the ME indicated that the UICC application type should be:

   - the USIM on the UICC; step b below is skipped and in steps c and d only USIM applications are considered.

   - the ISIM on the UICC; step ~~c~~ b below is skipped and in steps c and d only ISIM applications are considered.

   If the application on the ME did not indicate a preference, step b below is skipped and the selection process is executed as described below starting with step c,

   b. if a ì Labelî was indicated in step a, the GBA function shall select (see definition in TS 31.102 [1]) the UICC application with the ì Labelî indicated; if selection of this UICC application does not succeed the selection procedure fails;

   c. if no ì Labelî was indicated in step a, the ~~ME~~ GBA function shall ~~select~~ choose among the active ~~ISIMs~~UICC applications; if there is more than one active ~~ISIM~~UICC application, the ~~UE~~ GBA function may show a~~n~~ ~~ISIM~~ UICC application ~~selection~~ choosing dialogue to the end user (the list contains the "Labels" from the application list of the UICC), from which the end user chooses~~selects~~ the UICC application to be selected~~ISIM~~; if no dialogue is shown the ~~ME~~ GBA function shall select ~~any one of the active ISIMs~~the ì last selectedî active UICC application; in case the Ua application indicated ì no preferenceî and both a ì last selectedî USIM and a ì last selectedî ISIM are active, then the ì last selectedî USIM is selected.

   ~~c. the ME shall select among the active USIMs; if there is more than one active USIM, the UE may show a USIM selection dialog to the end user (the list contains the "Labels" from the application list of the UICC), from which the end user selects the USIM; if no dialogue is shown the ME shall select any one of the active USIMs.~~

   d. if there are no UICC applications active:

      - if there is only one UICC application, the ~~UE~~GBA function ~~activates~~ selects it, if possible~~, and selects it~~;

      - if there is more than one UICC application, the ~~UE~~ GBA function may show a UICC application ~~selection~~ choosing dialogue to the end user (the list contains the "Labels" from the application list of the UICC), from which the end user ~~selects~~ chooses the UICC application to be ~~activated~~selected; if no dialogue is shown the ~~ME~~ GBA function shall ~~activate~~ select the ~~default USIM~~ì last selectedî UICC application, if possible~~, and select it~~.

   e. if the type indicated in step a and used in step d was ISIM, but there was no ISIM to select, then step d is repeated with type USIM; otherwise the selection process fails.

   NOTE 1: Step e is required for the case that an ISIM as defined in TS 33.203 [16] may be realised using a USIM application on the UICC.

2. If there already is a key Ks derived from the chosen~~selected~~ UICC application, the UE takes this key to derive Ks_NAF.

3. If there is no such key Ks, the UE first runs the Ub protocol involving the selected UICC application and then goes to step 2.

If a USIM is chosen~~selected~~, the IMPI obtained from the IMSI stored on the USIM as specified in 3GPP TS 23.003 section 13.3 [11], is used in the protocol run over Ub.

NOTE ~~1~~2: Strictly speaking, an IMPI, and the derivation of an IMPI from an IMSI as in 3GPP TS 23.003 section 13 [11] are only defined in the context of the IMS. For the purposes of this specification, however, an identifier obtained from an IMSI as specified in 3GPP TS 23.003 section 13.3 [11] is also called an IMPI, even if the user has no IMS subscription.

If an ISIM is selected, the IMPI stored on the ISIM is used in the protocol run over Ub.

Whenever a~~n~~ UICC application is successfully selected or terminated~~ISIM or a USIM is activated or deactivated~~, the rules in this subsection for ~~selecting~~ choosing the UICC application are re-applied and, consequently, the ~~selected~~ UICC application chosen for GBA may change.

Whenever a UICC application is terminated ~~de-selected~~ the shared key Ks established from it in the protocol over the Ub reference point (according to sections 4.5.2 and 5.3.2) shall be deleted.

NOTE ~~2~~3: At any one time, there is at most one UICC application chosen~~selected~~ for performing the GBA procedures.

NOTE ~~3~~4: The Ua applications ~~on the ME~~ can continue using the NAF specific keys derived also after the shared key Ks itself has been deleted until the key lifetime expires.

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

3GPP TSG-SA WG3 Meeting S3#35
St Paul's Bay, Malta, October 5-8, 2004

Tdoc ⌘S3-040756

CR-Form-v7.1

CHANGE REQUEST

⌘ **33.220** **CR** **023** ⌘ **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 **X**

Title:	⌘	TLS profile for securing Zn' reference point
Source:	⌘	SA WG3
Work item code:	⌘	SEC1-SC
Date:	⌘	27/09/2004
Category:	⌘	**C**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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)*

Reason for change:	⌘	TLS profile that is used for securing the Zn' reference is defined.
Summary of change:	⌘	TLS profile is defined by using existing TLS profile specified in RFC 3588 with an addition that the client certificate of the D-Proxy shall contain FQDNs of the NAFs behind the D-Proxy. FQDNs are specified either by full FQDN or by using wildcard character as specified in RFC 2818. Also an informative annex is added to describe how TLS certificates may be enrolled and revocated.
Consequences if not approved:	⌘	TLS profile is not defined.

Clauses affected:	⌘	2, 4.4.6, annex D (new), annex E (new)

Other specs affected: ⌘	Y	N		⌘
		X	Other core specifications	
		X	Test specifications	
		X	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] 3GPP TS 31.103: "3rd Generation Partnership Project; Technical Specification Group Terminals; Characteristics of the IP Multimedia Services Identity Module (ISIM) application".

[11] 3GPP TS 23.003: "3rd Generation Partnership Project; Technical Specification Group Core Network; Numbering, addressing and identification".

[12] IETF RFC 3548 (2003): "The Base16, Base32, and Base64 Data Encodings".

[13] 3GPP TS 33.210: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Network domain security; IP network layer security".

[14] IETF RFC 3588 (2003): "Diameter Base Protocol".

[15] IETF RFC 3280 (2002): "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

[16] IETF RFC 2818 (2000): "HTTP over TLS".

[17] 3GPP TS 33.310: "3rd Generation Partnership Project; Technical Specification Group Service and System Aspects; Network Domain Security (NDS); Authentication Framework (AF)".

[18] IETF RFC 2560 (1999): "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP".

===== 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;
- If the BSF and the NAF are located within the same operator's network, the Zn reference point shall be secured according to NDS/IP [13];
- If the BSF and the NAF are located in different operators' networks, the Zn' reference point between the D-Proxy and the BSF shall be secured using TLS as specified in RFC 2246 [6];

~~Editorís Note: The TLS Certificate profiling needs to be completed and will be added into an Annex.~~

NOTE: Annex D specifies the TLS profile that is used for securing the Zn' reference point.

- 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 a selected set of application-specific user security settings from the BSF, depending on the policy of the BSF and the application indicated in the request from the NAF over Zn;
- The NAF shall be able to indicate to the BSF the single application or several applications it requires user security settings for;

NOTE: If some application needs only a subset of an application-specific user security setting, e.g. only one IMPU, the NAF selects this subset from the complete set of user security settings sent from BSF.

- The BSF shall be able to configure on a per NAF or per application basis if private subscriber identity and which user security settings may be sent to a NAF;
- The BSF shall be able to indicate to the NAF the lifetime of the key material. The key lifetime sent by the BSF over Zn shall indicate the expiry time of the key, and shall be identical to the key lifetime sent by the BSF to the UE over Ub.

NOTE: This does not preclude a NAF to refresh the key before the expiry time according to the NAF's local policy.

Editor's note: It is ffs which actions are to be taken over Zn when the BSF receives a user security settings update from the HSS over Zh.

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

Annex D (normative): TLS profile for securing Zn' reference point

The TLS profile for securing the Zn' reference point is specified in RFC 3588 [14] section 13.2.

In addition, the D-Proxy certificate, i.e., the client certificate used in TLS handshake shall contain the subjectAltName extension as specified in RFC 3280 [15]. The subjectAltName extension shall contain one or more dNSName names. The dNSName name may contain the wildcard character '*' and the matching is performed as specified in RFC 2818 [16] section 3.1.

NOTE: The D-Proxy certificate shall contain all the NAF_IDs of NAFs that may send a request for NAF specific shared secret through the D-Proxy to the subscriber's home BSF. If new NAF is added, the new NAF_ID is either covered in the certificate by using the wildcard character approach (e.g., "*.operator.com"), or a new dNSName name needs to be added to the certificate. In the latter case, new certificate is needed for the D-Proxy.

Annex E (informative): Handling of TLS certificates

An authentication framework as available for IPsec [17] is not available for TLS certificates. The purpose of this Annex is to provide guidelines for TLS certificate handling for use on the Zní reference point in the absence of a framework for TLS certificates.

Within this Annex following abbreviations are used: CA_A is the certification authority in Aís network and CA_B is the certification authority in Bís network. $Cert_A$ is the certificate of A and $Cert_B$ is the certificate of B. I_A is the set of identifiers that A may use as NAF_ID. T_B is the set of peers trusted by B

E.1 TLS certificate enrollment

Mutual authentication in TLS is achieved based on public key technology and certificates. Both TLS peers A and B need to contain a certificate store and there shall be at least one certification authority CA that can issue certificates within the security domains in with A and B are part of. $Cert_A$ contains the set I_A of Aís identifiers. Each identifier is in the form of fully qualified domain name (FQDN). Similarly, Bís certificate is $Cert_B$.

The certificates in the store of B define the group T_B of peers trusted by B. There are several options for creation and enrollment of certificates, three of which are described below.

1. In one option there is a certification authority, CA_B, only in the network of B. CA_B issues a certificate $Cert_B$ to B and a certificate $Cert_A$ to A. The certificates are delivered from CA_B to A and B in a secure way ëout of bandí. Both A and B then add their peer into the group of their trusted peers by inserting that peerís certificate into the certificate store: A inserts $Cert_B$ into Aís certificate store and B inserts $Cert_A$ into Bís certificate store. This insertion is typically manual and the details depend on the implementation of the management interface to the certificate store.

2. In another option both Aís and Bís networks contain certification authorities, CA_B and CA_A, respectively. CA_B issues a certificate $Cert_B$ to B and CA_A issues a certificate $Cert_A$ to A. The certificates are delivered from CA_B to A and from CA_A to B in a secure way ëout of bandí. Both A and B then add their peer into the group of their trusted peers by inserting that peerís certificate into the certificate store: A inserts $Cert_B$ into Aís certificate store and B inserts $Cert_A$ into Bís certificate store.

3. In a third option the CA certificates of both sides are exchanged: the certificate of CA_B is delivered to A and the certificate of CA_A is delivered to B in a secure way ëout of bandí, inserted to the certificate store, and marked trusted. The validation of $Cert_A$ and $Cert_B$ that are exchanged during TLS handshake, is based on the presence of the corresponding CA certificates in the certificate store.

NOTE: In options 1 and 2 the need for certification authority may be avoided if the peers generate self signed certificates and exchange them in a secure way, "out of band". Also, instead of certificates themselves, certificate fingerprints may be exchanged "out of band" in those options.

E.2 TLS Certificate revocation

In the absence of PKI-revocation interfaces, certificate revocation needs to be performed manually. The revocation operation involves the removal of A from the group T_B of peers trusted by B. In the first two enrollment options described above the revocation happens by B removing the certificate of A, $Cert_A$, from its certificate store. This removal can be done manually. In the third option the certificate of A, $Cert_A$, is not in Bís certificate store. For that reason B has to have a way to check the validity of $Cert_A$ with the issuer of the certificate. (Also in the first two enrollment options the amount of manual maintenance operations will decrease if B can check the validity of $Cert_A$ with

the issuer of the certificate.) This check may be done by using Online Certificate Status Protocol (OCSP) [18] or by using Certificate Revocation Lists (CRLs) [15] published by the issuer of $Cert_A$.

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

3GPP TSG SA WG3 Security ñ S3#36
Shenzhen, 23-26 November 2004

Tdoc ⌘ S3-041136

CR-Form-v7.1

CHANGE REQUEST

⌘ **33.220** **CR** **025** ⌘ **rev** **2** ⌘ 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⌘ [X] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Optimization of the GBA_U key derivation procedure
Source:	⌘	SA WG3
Work item code:	⌘	SEC1-SC
Date:	⌘	12/11/2004
Category:	⌘	**C**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
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)*

Reason for change:	⌘	The current version of TS 33.220 requires the UICC and the BSF to perform four key derivation procedures to calculate Ks_int_NAF and Ks_ext_NAF keys (first the UICC performs Ks derivation, then performs Ks_int and Ks_ext derivation, then performs Ks_int_NAF and Ks_ext_NAF derivation). This procedure can be optimized by reducing the number of key derivations from four to three, while achieving the same level of security (Ks_int_NAF and Ks_ext_NAF can be directly derived from Ks). Besides, the changes proposed in this CR are in line with SA3#35 decision on the storage of Ks_ext. In fact, SA3#35 decided that "If the UICC supports GBA_U, Ks_ext shall not leave the UICC" (see SA3#35 meeting report) The solution proposed in this document presents the following benefits: - Reduce significantly the bootstrapping time, as the UICC would have to perform one key derivation for the bootstrapping instead of two. This will lead to better performance in the UICC and BSF (less network resources consumption/less booting time). Reduce significantly the implementation complexity in the BSF, as the GBA_U procedures will be similar to the GBA_ME procedures. The only difference between the two procedures will be in the handling of the modified MAC and the derivation of Ks_int_NAF, which is very similar to the derivation of Ks_ext_NAF (the later is also very similar to Ks_NAF derivation).
Summary of change:	⌘	- Optimization of the bootstrapping procedure by removing Ks_int and Ks_ext derviation. This CR proposes to derive Ks_int_NAF/Ks_ext_NAF directly from Ks. - The description of the UICC-ME interface is added as normative annex.
Consequences if not approved:	⌘	Unecessary complexity in the implementation of GBA_U. Description of the solution is not complete. In particular, the location of Ks_ext will remain unspecified for GBA_U.

Clauses affected:	⌘	3.2, 5, Annex D (new)

	Y	N		
Other specs Affected:	⌘		X	Other core specifications ⌘
			X	Test specifications
			X	O&M Specifications

Other comments:	⌘	

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AK	Anonymity Key
AKA	Authentication and Key Agreement
B-TID	Bootstrapping Transaction Identifier
BSF	Bootstrapping Server Function
CA	Certificate Authority
FQDN	Fully Qualified Domain Name
GAA	Generic Authentication Architecture
GBA	Generic Bootstrapping Architecture
GBA_ME	ME-based GBA
GBA_U	GBA with UICC-based enhancements
HSS	Home Subscriber System
IK	Integrity Key
KDF	Key Derivation Function
Ks_int_NAF	Derived key in GBA_U which remains on UICC
Ks_ext_NAF	Derived key in GBA_U
MNO	Mobile Network Operator
NAF	Network Application Function
PKI	Public Key Infrastructure
USS	GBA User Security Setting

5 UICC-based enhancements to Generic Bootstrapping Architecture (GBA_U)

It is assumed that the UICC, BSF, and HSS involved in the procedures specified in this clause are capable of handling the GBA_U specific enhancements. The procedures specified in this clause also apply if NAF is not GBA_U aware~~, but, of course, in that case there are no benefits of the GBA_U specific enhancements~~.

5.1 Architecture and reference points for bootstrapping with UICC-based enhancements

The text from clause 4.4 of this specification applies also here, with the addition that the interface between the ME and the UICC, as specified in TS 31.102 [1] and TS 31.103 [10], needs to be enhanced with GBA_U specific commands. The requirements on these commands can be found in clause 5.2.1, details on the procedures are in clause 5.3.

5.2 Requirements and principles for bootstrapping with UICC-based enhancements

The requirements and principles from clause 4.3 also apply here with the following addition:

5.2.1 Requirements on UE

The 3G AKA keys CK and IK resulting from a run of the protocol over the Ub reference point shall not leave the UICC.

The UICC shall be able to distinguish between authentication requests for GBA_U, and authentication requests for other 3G authentication domains.

Upon an authentication request from the ME, which the UICC recognises as related to GBA_U, the UICC shall derive the bootstrapping key~~two keys from CK and IK~~. All 3G MEs are capable of such a request.

Upon request from the ME, the UICC shall be able to derive further NAF-specific keys from the derived key stored on the UICC. Only GBA_U-aware 3G MEs are capable of such a request.

~~Editors' Note: The location (whether in the UICC or in the ME) of the storage of Ks_ext is ffs.~~

5.2.2 Requirements on BSF

BSF shall support both GBA_U and GBA_ME bootstrapping procedures. The decision on running one or the other shall be based on subscription information (i.e. UICC capabilities).

The BSF shall be able to acquire the UICC capabilities related to GBA as part of the GBA user security settings received from the HSS.

5.3 Procedures for bootstrapping with UICC-based enhancements

5.3.1 Initiation of bootstrapping

The text from clause 4.5.1 of this document applies also here.

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.

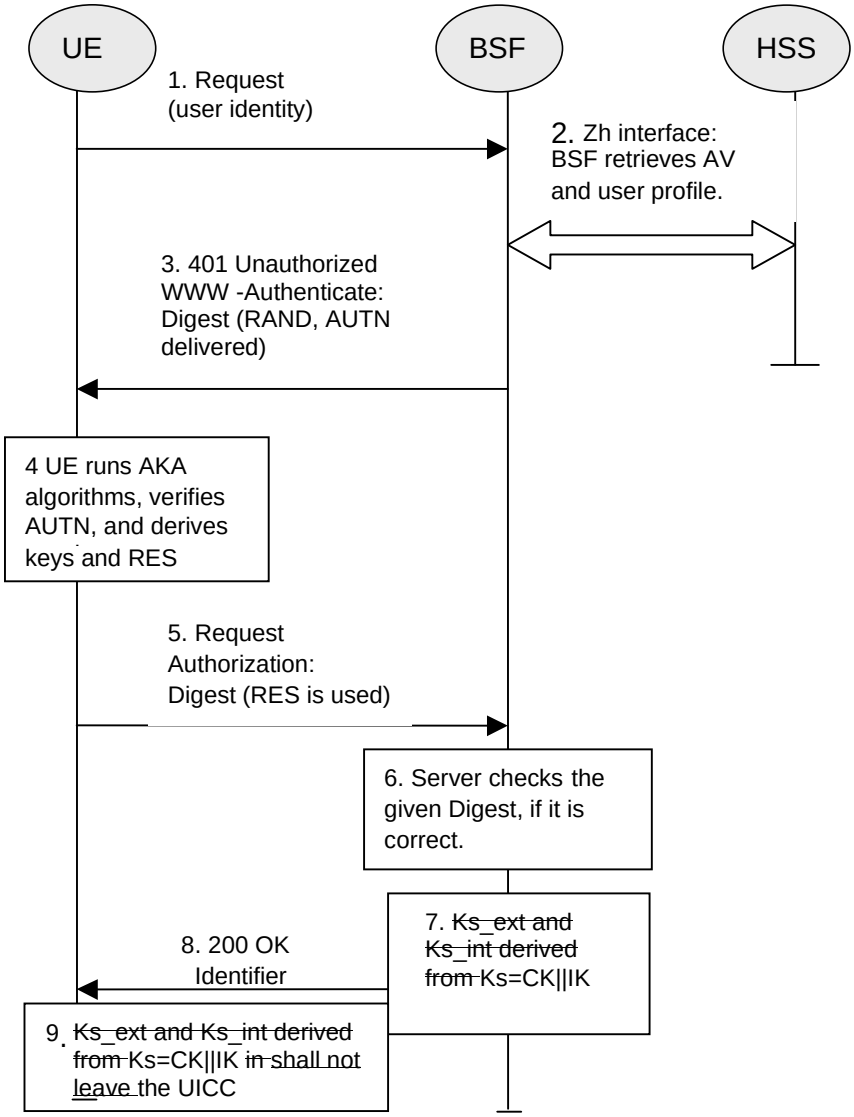


Figure 5.1: The bootstrapping procedure with UICC-based enhancements

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 □ 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 ⊕ 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* ⊕ SHA-1(IK1)). Then the UICC checks AUTN(i.e. SQN ⊕ 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. The UICC then transfers RES (after flipping the least significant bit) to the ME and stores Ks, which is the concatenation of CK and IK, on the 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 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~~5. The ME sends another HTTP request, containing the Digest AKA response (calculated using RES), to the BSF.

~~7~~6. The BSF authenticates the UE by verifying the Digest AKA response.

~~8~~7. 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~~8. 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 key~~s~~ Ks~~_ext and Ks_int, The lifetimes of the keys Ks_ext and Ks_int shall be the same~~.

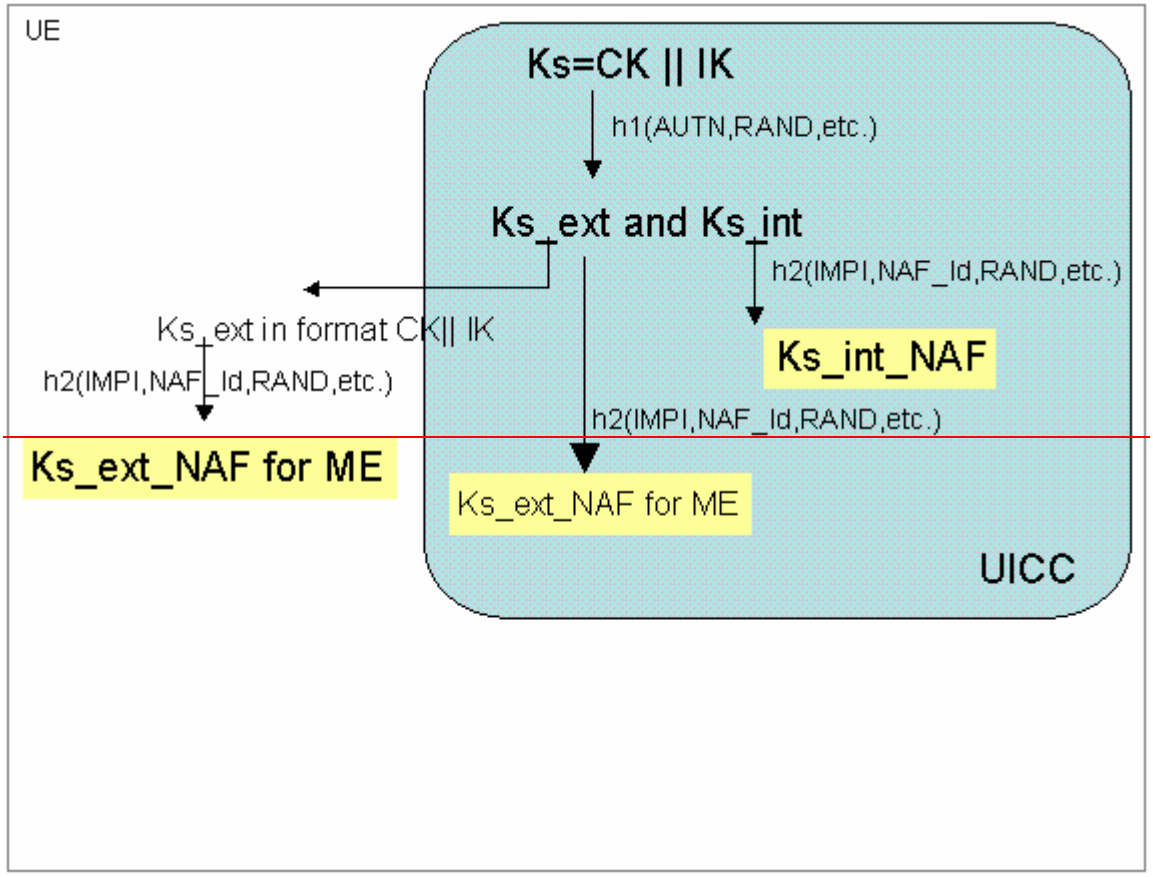
~~10~~9. Both the UICC and the BSF shall use the Ks to derive NAF-specific keys Ks_ext_NAF and Ks_int_NAF during the procedures as specified in clause 5.3.3, if applicable. ~~The BSF shall use the keys Ks_ext and Ks_int to derive the NAF specific keys Ks_ext_NAF and Ks_int_NAF, if requested by a NAF 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 in the UICC as Ks_ext_NAF = ~~h2~~ KDF (Ks~~_ext~~, h1~~2~~-key derivation parameters), and Ks_int_NAF is computed in the UICC as Ks_int_NAF = ~~h2~~ KDF (Ks~~_int~~, h1~~2~~-key derivation parameters), where ~~h2~~ KDF is ~~a~~ the ~~suitable~~ key derivation function as specified in Annex B, 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. The key derivation parameters used for Ks_ext_NAF derivation must be different from those used for Ks_int_NAF derivation. This is done by adding a static string "gba-me" in Ks_ext_NAF and "gba-u" in Ks_int_NAF as an input parameter to the key derivation function.

~~Editors' Note: The definition of the h21 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 key~~s~~ 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 key~~s~~ Ks~~_ext and Ks_int are~~ is updated.



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

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_NAF and Ks_int_NAF 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-unaware 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~~UICC, the ~~UE~~ ME requests the UICC to derive~~s~~ 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 key~~s~~ 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~~ is not available in the UE, the UE first agrees on a new key~~s~~ 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 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, a new key Kss ~~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 key~~s~~ Ks~~_ext and Ks_int are~~ is 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 NAFís 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 BSFís 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.

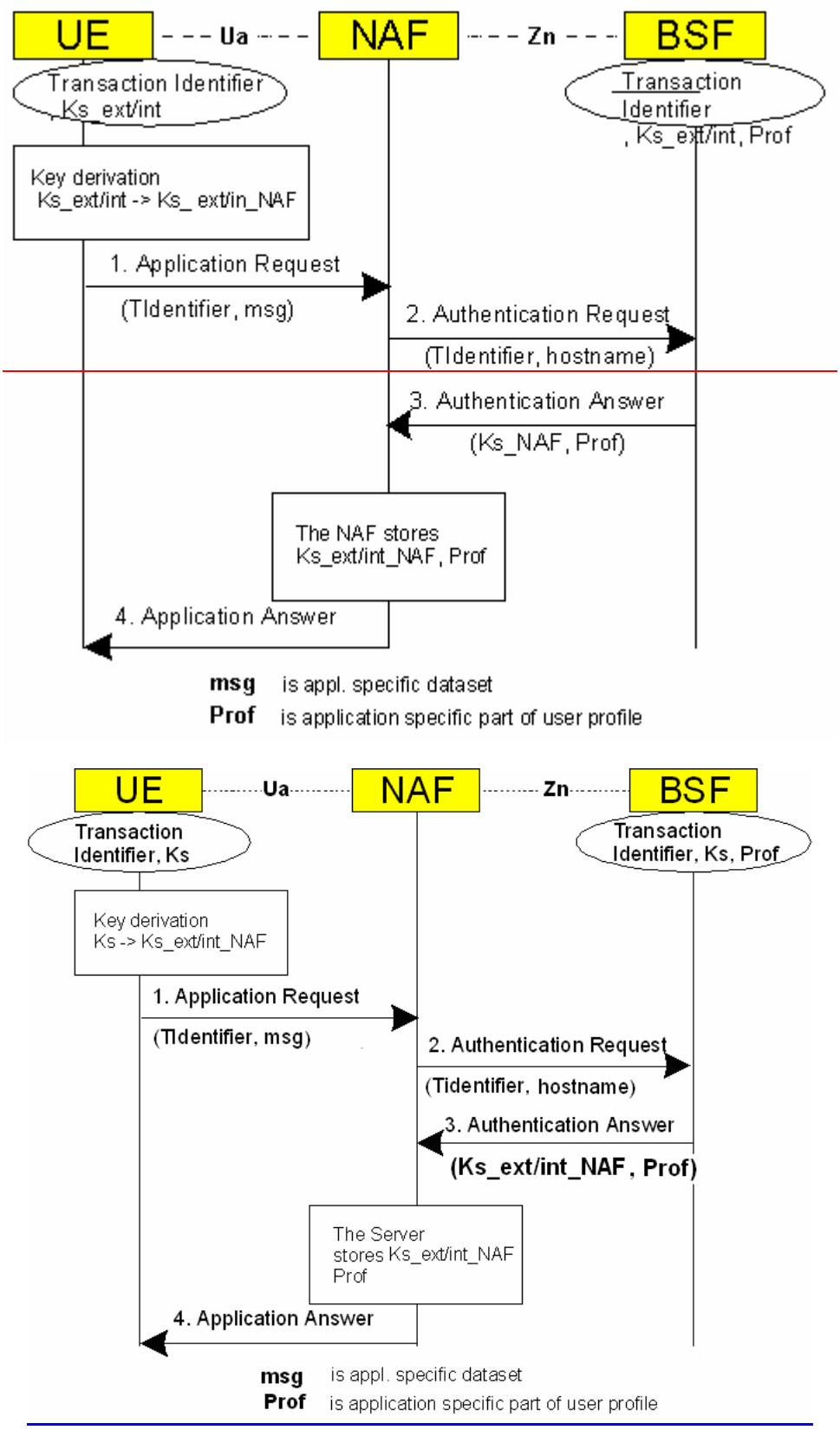


Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

5.3.4 Procedure related to service discovery

The text from clause 4.5.4 of this document applies also here.

Annex D (normative): GBA_U UICC-ME interface

This section describes the UICC-ME interface to be used when a GBA_U aware UICC application is active and the ME is involved in a GBA bootstrapping procedure. When the UICC application is not GBA_U aware, the ME uses AUTHENTICATE command in non-GBA_U security context (i.e. UMTS security context in case of USIM application and IMS security context in case of the ISIM) as defined in 31.102 [1] and 31.103 [xx].

D.1. GBA_U Bootstrapping procedure

This procedure is part of the Bootstrapping procedure as described in section 5.3.2

The ME sends RAND and AUTN to the UICC, which performs the Ks derivation as described in 5.3.2.

The UICC then stores Ks. The UICC also stores the used RAND to identify the current bootstrapped values. RAND value in the UICC shall be further accessible by the ME.

The ME then, finalizes the Bootstrapping procedure and stores in the UICC the Transaction Identifier (B-TID) and Key Life Time associated with the previous bootstrapped keys (i.e. Ks). Transaction Identifier and Key Life Time values in the UICC shall be further accessible by the ME.

At the end of the GBA_U bootstrapping procedure the UICC stores Ks, Transaction Identifier, Key Life Time and the RAND.

The UICC sends RES to the ME.

A new bootstrapping procedure replaces Ks, B-TID, Key LifeTime and RAND values of the previous bootstrapping procedure.

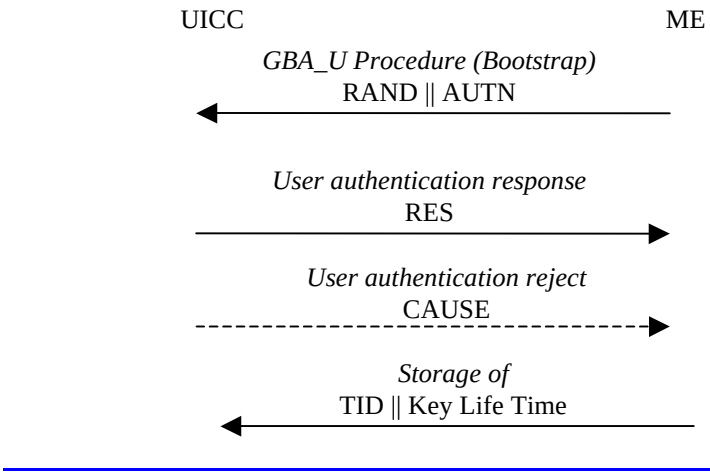


Figure x: GBA_U Bootstrap Procedure

D.2. GBA_U NAF Derivation procedure

This procedure is part of the Procedures using bootstrapped Security Association as described in section 5.3.3

The ME sends NAF_ID and IMPI to the UICC. The UICC then performs Ks_ext_NAF and Ks_int_NAF derivation as described in 5.3.2. The UICC uses the RAND and Ks values stored from the previous bootstrapping procedure. The UICC returns Ks_ext_NAF to the ME and stores Ks_int_NAF together with NAF_Id.

Note: A previous GBA_U Bootstrap needs to be undertaken before. If Ks is not available in the UICC, the command will answer with the appropriate error message.

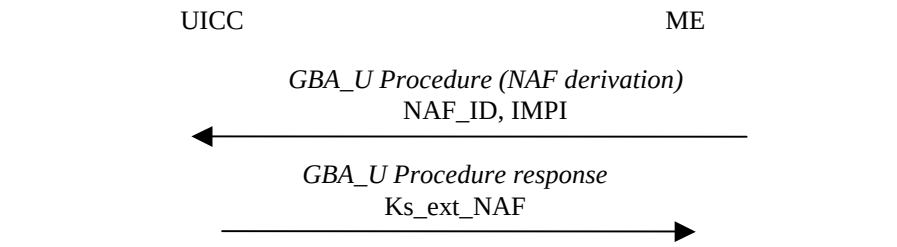


Figure x: GBA_U NAF derivation procedure

3GPP TSG SA WG3 Security ó S3#36 **S3-041080**
Shenzhen, 23 ñ 26 November 2004

CR-Form-v7

CHANGE REQUEST

⌘ **33.220** **CR** **027** ⌘ **rev** **2** ⌘ 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 [X] Radio Access Network ☐ Core Network ☐

Title:	⌘	Requirement on ME capabilities for GBA_U
Source:	⌘	SA WG3
Work item code:	⌘	SEC1-SC
Date:	⌘	25/11/2004
Category:	⌘	**B**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one 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:	⌘	At SA3#35 meeting SA3 agreed that if the UICC supports GBA_U, the boostrapping keys shall not leave the UICC and as a direct consequence, a GBA terminal will behave differently if a GBA-capable UICC is inserted, to when a legacy UICC without GBA support is insertedî. This agreement implies that all GBA-aware MEs shall support both GBA_ME and GBA_U procedures. The execution of GBA_U procedure shall not depend on the ME capabilities.
Summary of change:	⌘	Addition of a requirement on GBA-aware ME to support GBA_U procedure.
Consequences if not approved:	⌘	The current version of the TS does not reflect the agreement reached at SA3#35 meeting and a complete GBA_U bootstrap will not be possible depending on ME capabilities.

Clauses affected:	⌘	4.2.4, 5

	Y	N			
Other specs affected:	X		Other core specifications	⌘	TS 31.102, TS 31.103
		X	Test specifications		
		X	O&M Specifications		

Other comments:	⌘	-

BEGIN OF CHANGE

4.2.4 UE

The required functionalities from the UE are:

- the support of HTTP Digest AKA protocol;
- the capability to use both a USIM and an ISIM in bootstrapping;
- the capability to select either a USIM or an ISIM to be used in bootstrapping, when both of them are present;
- the capability for an application on the ME using the shared secret to indicate the type of UICC application to use in bootstrapping (i.e., ISIM or USIM);
- the capability to derive new key material to be used with the protocol over Ua interface from CK and IK;
- support of NAF-specific application protocol (For an example see TS 33.221 [5]);

A GBA-aware ME shall support both GBA_U, as specified in clause 5.2.1 and GBA_ME procedures, as specified in clause 4.5.

END OF CHANGE

BEGIN OF CHANGE

5 UICC-based enhancements to Generic Bootstrapping Architecture (GBA_U)

It is assumed that the UICC, BSF, and HSS involved in the procedures specified in this clause are capable of handling the GBA_U specific enhancements. The procedures specified in this clause also apply if NAF is not GBA_U aware, but, of course, in that case there are no benefits of the GBA_U specific enhancements.

5.1 Architecture and reference points for bootstrapping with UICC-based enhancements

The text from clause 4.4 of this specification applies also here, with the addition that the interface between the ME and the UICC, as specified in TS 31.102 [1] and TS 31.103 [10], needs to be enhanced with GBA_U specific commands. The requirements on these commands can be found in clause 5.2.1, details on the procedures are in clause 5.3.

5.2 Requirements and principles for bootstrapping with UICC-based enha ncements

The requirements and principles from clause 4.3 also apply here with the following addition:

5.2.1 Requirements on UE

The 3G AKA keys CK and IK resulting from a run of the protocol over the Ub reference point shall not leave the UICC.

The UICC shall be able to distinguish between authentication requests for GBA_U, and authentication requests for other 3G authentication domains.

Upon an authentication request from the ME, which the UICC recognises as related to GBA_U, the UICC shall derive two keys from CK and IK. ~~All 3G MEs are capable of such a request.~~

Upon request from the ME, the UICC shall be able to derive further NAF-specific keys from the derived key stored on the UICC. ~~Only GBA_U-aware 3G MEs are capable of such a request.~~

All GBA-aware MEs shall support procedures for the two previous requests.

~~Editors' Note: The location (whether in the UICC or in the ME) of the storage of Ks_ext is ffs.~~

5.2.2 Requirements on BSF

BSF shall support both GBA_U and GBA_ME bootstrapping procedures. The decision on running one or the other shall be based on subscription information (i.e. UICC capabilities).

The BSF shall be able to acquire the UICC capabilities related to GBA as part of the GBA user security settings received from the HSS.

5.3 Procedures for bootstrapping with UICC-based enhancements

5.3.1 Initiation of bootstrapping

The text from clause 4.5.1 of this document applies also here.

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.

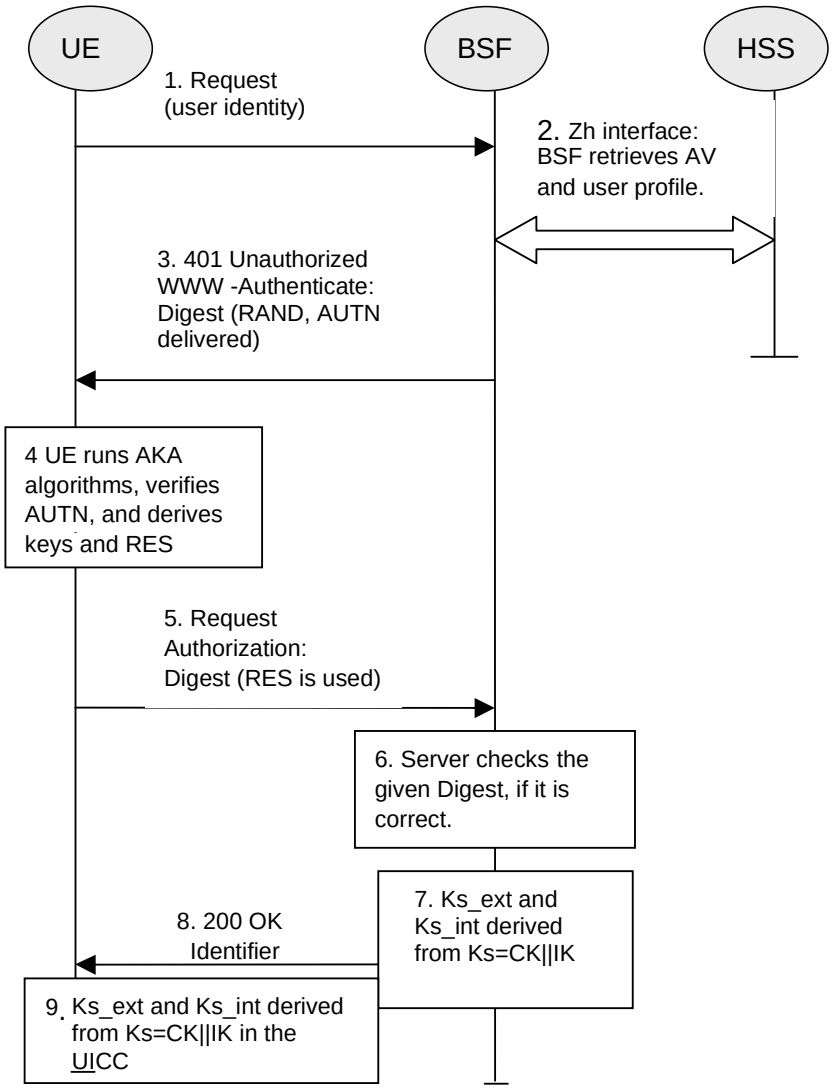


Figure 5.1: The bootstrapping procedure with UICC-based enhancements

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 □ 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 ⊕ 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 UICCcalculates IK and MAC (by performing MAC= MAC* ⊕ SHA-1(IK1)). Then the UICC checks AUTN(i.e. SQN ⊕ 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 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/K~~k~~s_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-key derivation parameters), and Ks_int_NAF is computed in the UICC as Ks_int_NAF = h2 (Ks_int, h2-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.

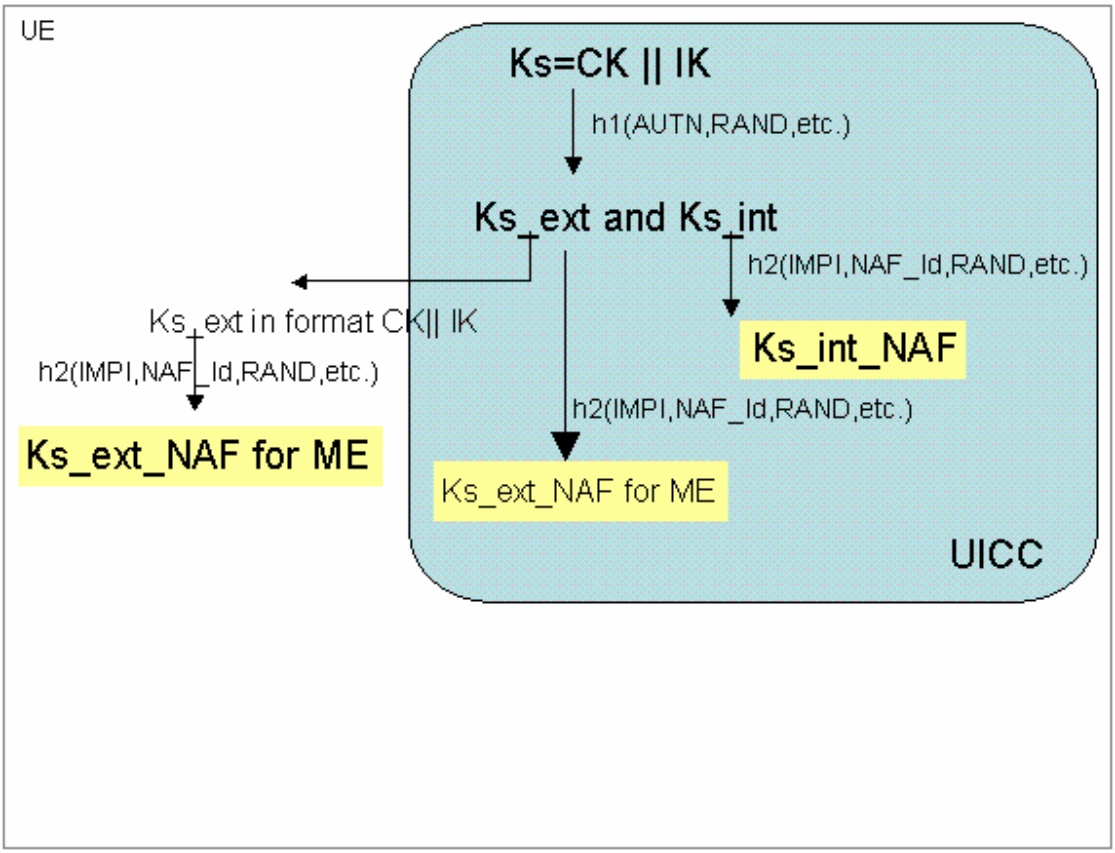


Figure 5.2: Key derivation for GBA-aware UICC when GBA-run was triggered

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.

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.

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 NAFís 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 BSFís 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.

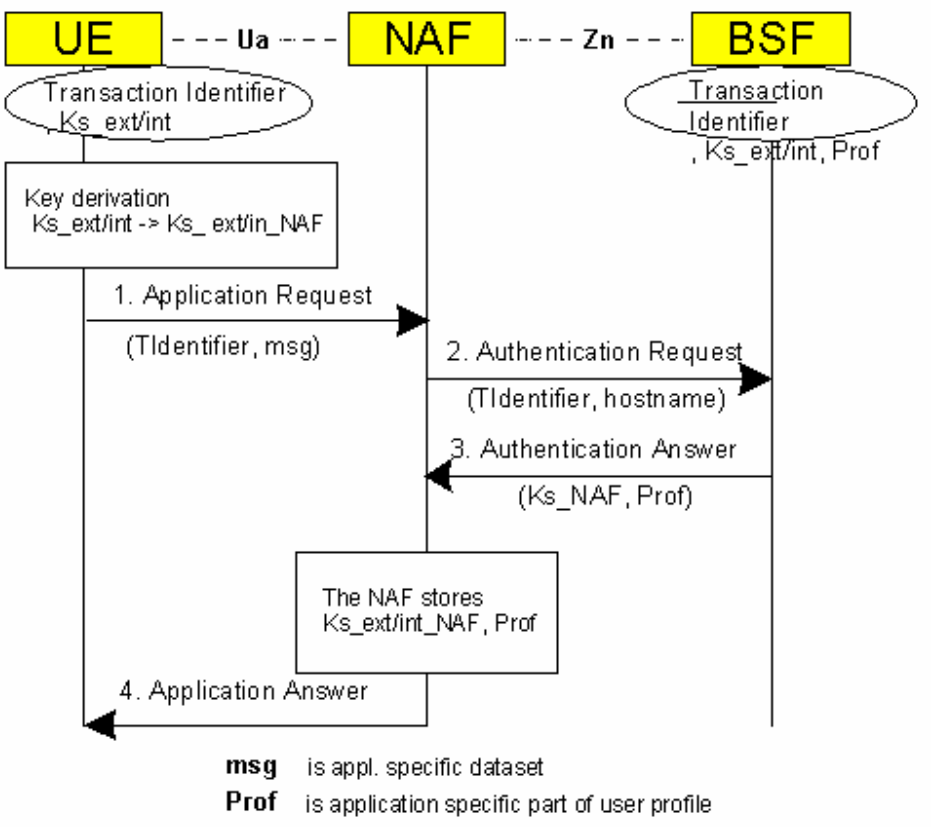


Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

5.3.4 Procedure related to service discovery

The text from clause 4.5.4 of this document applies also here.

END OF CHANGE

3GPP TSG-SA WG3 Meeting S3#35
Shenzen, China, November 23rd- 26th, 2004

Tdoc ⌘S3-041087

CR-Form-v7.1

CHANGE REQUEST

⌘ **33.220** **CR** **034** ⌘ **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 **X**

Title:	⌘	Adding a note about replay protection
Source:	⌘	**SA WG3**
Work item code:	⌘	SEC1-SC
Date:	⌘	16/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
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)

Reason for change:	⌘	For Ua protocols that have no intrinsic replay protection, implementers should be aware that GBA does not guarantee key freshness without forcing a re-run of AKA.
Summary of change:	⌘	A note that warns about the dangers of re-using keys with some Ua protocols is added.
Consequences if not approved:	⌘	An implementation of a Ua protocol without intrinsic replay protection may allow re-use of a key, which could lead to the replay attacks being possible.

Clauses affected:	⌘	4.2.2

Other specs affected: ⌘	Y	N	
		X	Other core specifications
		X	Test specifications
		X	O&M Specifications

⌘

Other comments: ⌘

4.2.2 Network application function (NAF)

After the bootstrapping has been completed, the UE and an operator-controlled NAF can run some application specific protocol where the authentication of messages will be based on those session keys generated during the mutual authentication between UE and BSF.

General assumptions for the functionality of an operator-controlled NAF are:

- there is no previous security association between the UE and the NAF;
- NAF shall be able to locate and communicate securely with the subscriber's BSF;
- NAF shall be able to acquire a shared key material established between UE and the BSF during the run of the application-specific protocol;
- NAF shall be able to acquire an (application-specific) user security setting from the HSS via the BSF;
- NAF shall be able to check lifetime of the shared key material.

NOTE: Without additional measures, GBA does not guarantee the freshness of the key, Ks(_int/ext)_NAF in the sense that it does not guarantee that the key was not used in a previous run of the Ua protocol. The additional measures which may be taken by the UE and the NAF to ensure key freshness in GBA are:
1) enforce a new run of the Ub protocol (thus generating a new Ks) before deriving a new Ks_NAF.
2) store previously used keys Ks(_int/ext)_NAF, or the corresponding key identifiers B-TID, until the end of their lifetime.
A UE and a NAF that support a Ua protocol that does not provide replay protection over unconnected runs of the protocol, will need to take corresponding action to avoid replay attacks if desired.

3GPP TSG-SA3 Meeting #36
Shenzhen, China, 23-26 October 2004

Tdoc ⌘S3-041078

CR-Form-v7

CHANGE REQUEST

⌘ **33.220** **CR** **035** ⌘**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⌘ [X] ME [ ] Radio Access Network [ ] Core Network [X]

Title:	⌘	Complete the MAC modification for GBA_U
Source:	⌘	SA WG3
Work item code:	⌘	SEC1-SC
Date:	⌘	15/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one 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:	⌘	- Correction of wrong symbols - Removal of editorís Note by proposing a hash truncation
Summary of change:	⌘	Complete the MAC modification for GBA_U
Consequences if not approved:	⌘	Incomplete specification

Clauses affected:	⌘	2, 5.3.2, New clauses 3.3 and 3.4			
Other specs affected:	⌘	**Y**	**N**		⌘
			X	Other core specifications	
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

** FIRST 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] 3GPP TS 31.103: "3rd Generation Partnership Project; Technical Specification Group Terminals; Characteristics of the IP Multimedia Services Identity Module (ISIM) application".

[11] 3GPP TS 23.003: "3rd Generation Partnership Project; Technical Specification Group Core Network; Numbering, addressing and identification".

[12] IETF RFC 3548 (2003): "The Base16, Base32, and Base64 Data Encodings".

[13] 3GPP TS 33.210: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Network domain security; IP network layer security".

[14] IETF RFC 3588 (2003): "Diameter Base Protocol".

[15] FIPS PUB 180-2 (2002): "Secure Hash Standard".

** END OF CHANGE ***

** NEXT CHANGE ***

3.3 Symbols

For the purposes of the present document, the following symbols apply:

|| Concatenation
⊕ Exclusive or

3.4 Conventions

All data variables in this specification are presented with the most significant substring on the left hand side and the least significant substring on the right hand side. A substring may be a bit, byte or other arbitrary length bitstring. Where a variable is broken down into a number of substrings, the leftmost (most significant) substring is numbered 0, the next most significant is numbered 1, and so on through to the least significant.

** END OF CHANGE ***

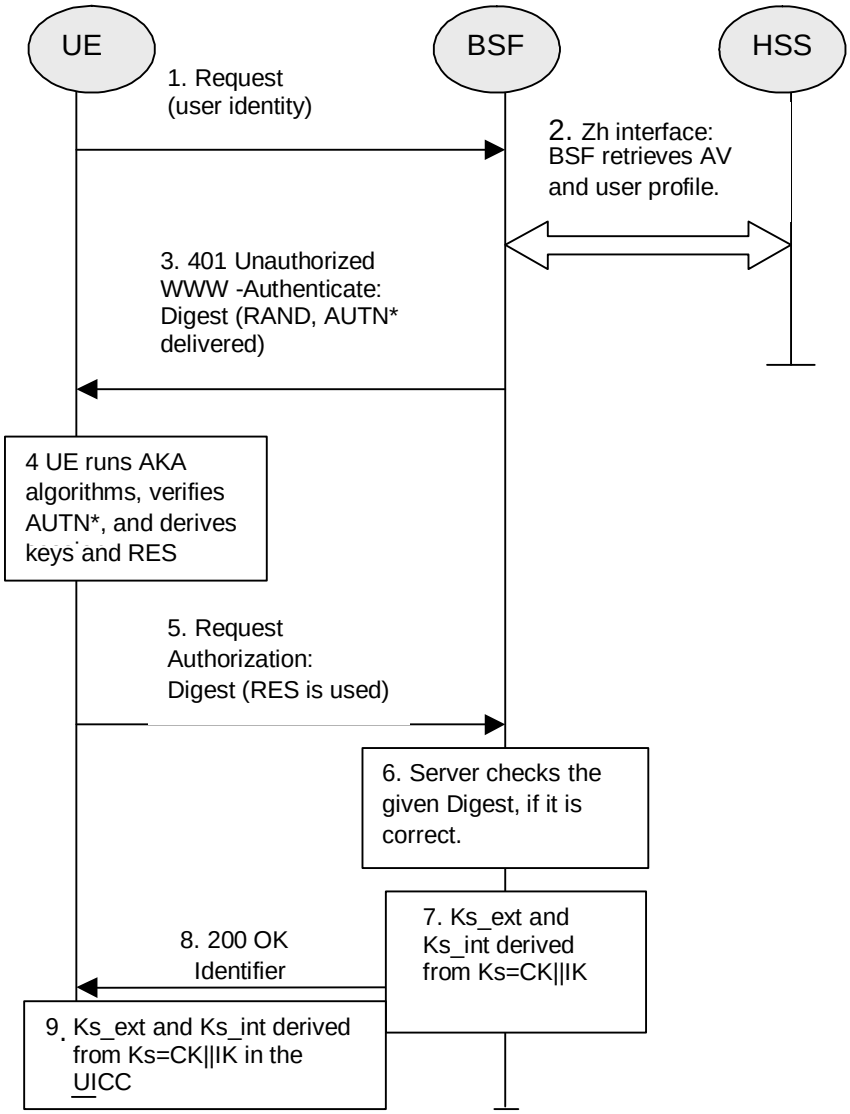
** LAST CHANGE ***

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.


UE
BSF
HSS
1. Request
(user identity)
2. Zh interface:
BSF retrieves AV
and user profile.
3. 401 Unauthorized
WWW -Authenticate:
Digest (RAND, AUTN*
delivered)
4 UE runs AKA
algorithms, verifies
AUTN*, and derives
keys and RES
5. Request
Authorization:
Digest (RES is used)
6. Server checks the
given Digest, if it is
correct.
7. Ks_ext and
Ks_int derived
from Ks=CK||IK
8. 200 OK
Identifier
9. Ks_ext and Ks_int derived
from Ks=CK||IK in the
UICC

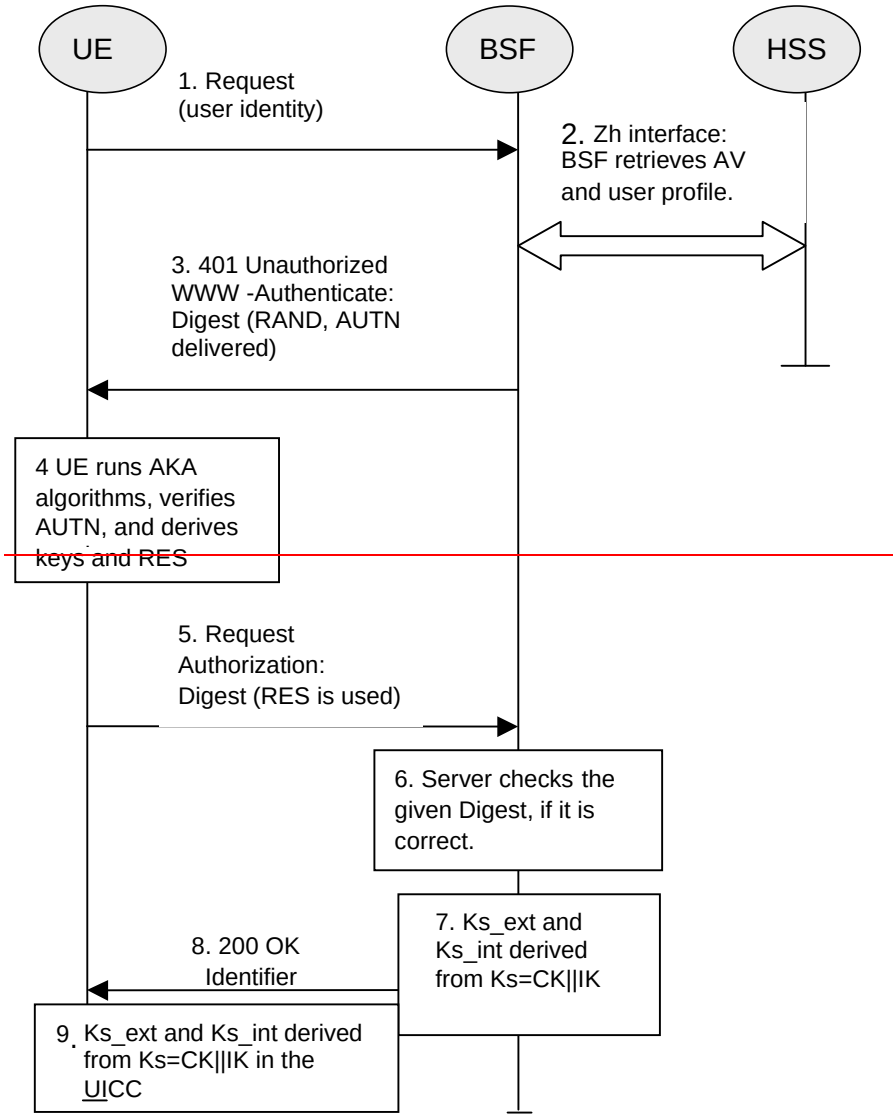


Figure 5.1: The bootstrapping procedure with UICC-based enhancements

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⊕ Trunc(~~□~~ SHA-1(IK~~1~~)) ~~(where IK= IK1|| IK2 and * is a exclusive or as described in TS 33.102 [2])~~

NOTE: Trunc denotes that from the 160 bit output of SHA-1[15], the 64 bits numbered as [0] to [63] are used within the ⊕ operation to MAC.

~~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 ⊕ 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* ⊕ Trunc(SHA-1(IK~~1~~))). Then the UICC checks AUTN(i.e. SQN ⊕ 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 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-key derivation parameters), and Ks_int_NAF is computed in the UICC as Ks_int_NAF = h2 (Ks_int, h2-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.

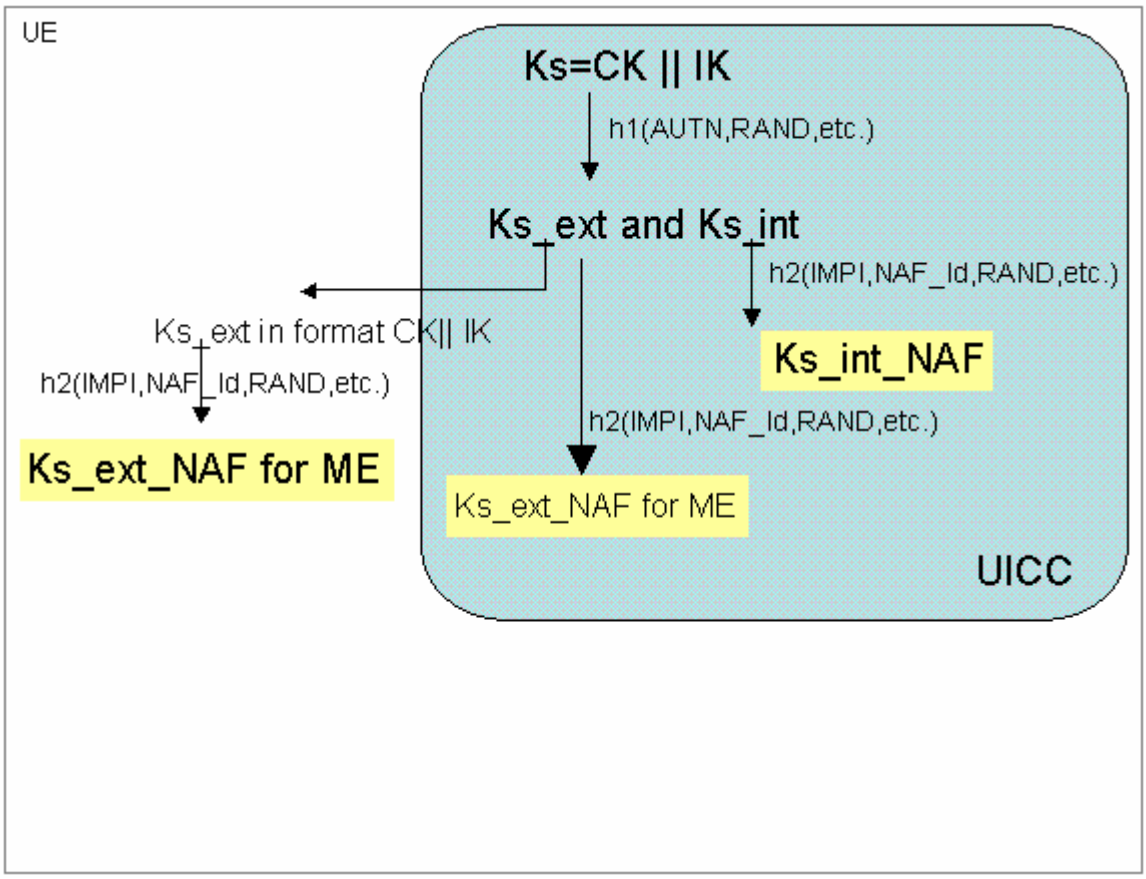


Figure 5.2: Key derivation for GBA-aware UICC when GBA-run was triggered

** END OF CHANGE ***

3GPP TSG-SA WG3 Meeting #36
Shenzhen, China, November 23-26, 2004

Tdoc ⌘S3-041082

CR-Form-v7.1

CHANGE REQUEST

⌘ **33.220** **CR** **036** ⌘ **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:	⌘	Removal of unnecessary editor's notes
Source:	⌘	SA WG3
Work item code:	⌘	SEC1-SC
Date:	⌘	16/11/2004
Category:	⌘	**D**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
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)

Reason for change:	⌘	Unnecessary editor's notes are deleted.
Summary of change:	⌘	- 3.1: definitions are completed - 4.3.3 and 4.3.4: USS is capable of transferring the authorization part. It is up to application itself whether the authorization part is used or not.
Consequences if not approved:	⌘	

Clauses affected:	⌘	3.1, 4.3.3, 4.3.4

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

==== *BEGIN 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. BSF, HSS, and UEs participate in GBA in which a shared secret is established between the network and a UE by running the bootstrapping procedure. The shared secret can be used between NAFs and UEs, for example, for authentication purposes.

~~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~~. GBA may be used between NAFs and UEs for authentication purposes, and for securing the communication path between the UE and the NAF.

~~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.

GBA User Security Setting: An application-specific parameter set describing the security related usage of bootstrapping function by the BSF and, optionally, some types of NAFs in the context of an application and in relation to a subscriber. A user security setting has two parts, an authentication part, which contains the list of identities of the user needed for the application (e.g. IMPI, IMPUs, MSISDN, pseudonyms), and an authorisation part, which contains the user permission flags (e.g. access to application allowed, type of certificates which may be issued). Sometimes also called application-specific user security setting.

GBA User Security Settings: the set of all application-specific user security settings.

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

4.3.3 Reference point Zh

The reference point Zh used between the BSF and the HSS allows the BSF to fetch the required authentication information and all GBA user security settings from the HSS. The interface to the 3G Authentication Centre is HSS-internal, and it need not be standardised as part of this architecture.

~~Editorís note: It is ffs, jointly with CN4 and SA2, whether the authorisation part of all USSs is transferred over Zh, or by other means. SA3 expresses a preference for Release 6, however, to transfer the authorisation part of the USSs for, at least, the GBA specific entities PKI portal (cf. TS. 33.221) and Authentication Proxy (TS 33.222) over Zh.~~

4.3.4 Reference point Zn

The reference point Zn is used by the NAF to fetch the key material agreed during a previous HTTP Digest AKA protocol run over the reference point Ub from the UE to the BSF. It is also used to fetch application-specific user security settings from the BSF, if requested by the NAF.

~~Editorís note: It is ffs, jointly with CN4 and SA2, whether the authorisation part of the application-specific USSs is transferred over Zn, or by other means. SA3 expresses a preference for Release 6, however, to transfer also the authorisation part of the application specific USSs for, at least, the GBA specific entities PKI portal (cf. TS. 33.221) and Authentication Proxy (TS 33.222) over Zn.~~

==== *END CHANGE* ====

3GPP TSG SA WG3 Security ó S3#36 **S3-041090**
November 23-26, 2004, Shenzhen, China

CR-Form-v7

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 **X**

Title:	⌘	Fetching of one AV only on each Zh run between BSF and HSS
Source:	⌘	SA WG3
Work item code:	⌘	SEC1-SC
Date:	⌘	15/11/2004
Category:	⌘	**C**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one 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:	⌘	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 ìrenegotiation requestî 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.
Summary of change:	⌘	Only one authentication vector AV may be fetched by BSF from HSS on each protocol run over Zh reference point.
Consequences if not approved:	⌘	(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).

Clauses affected:	⌘	4.5.2, 5.3.2
Other specs affected:	⌘	Y N X ☐ Other core specifications ⌘ TS 29.109 ☐ X Test specifications ☐ X O&M Specifications
Other comments:	⌘	-

********** 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.

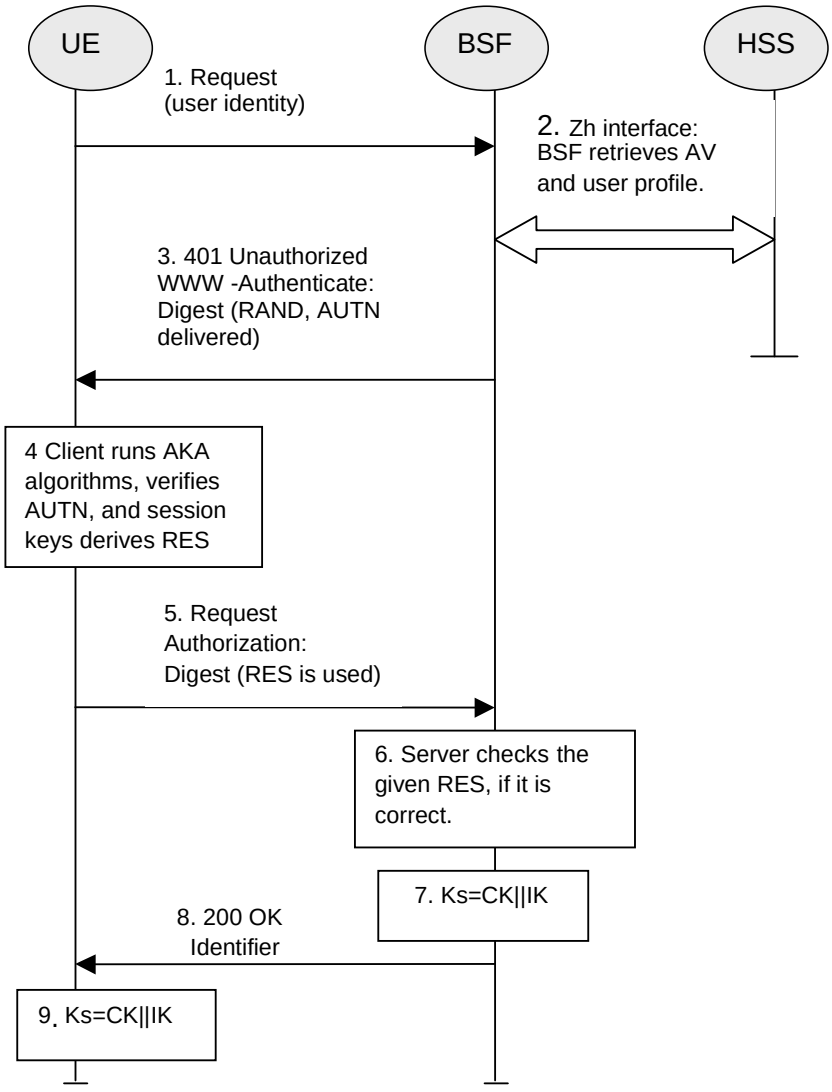


Figure 4.3: The bootstrapping procedure

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 Vector~~s~~ (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, 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.

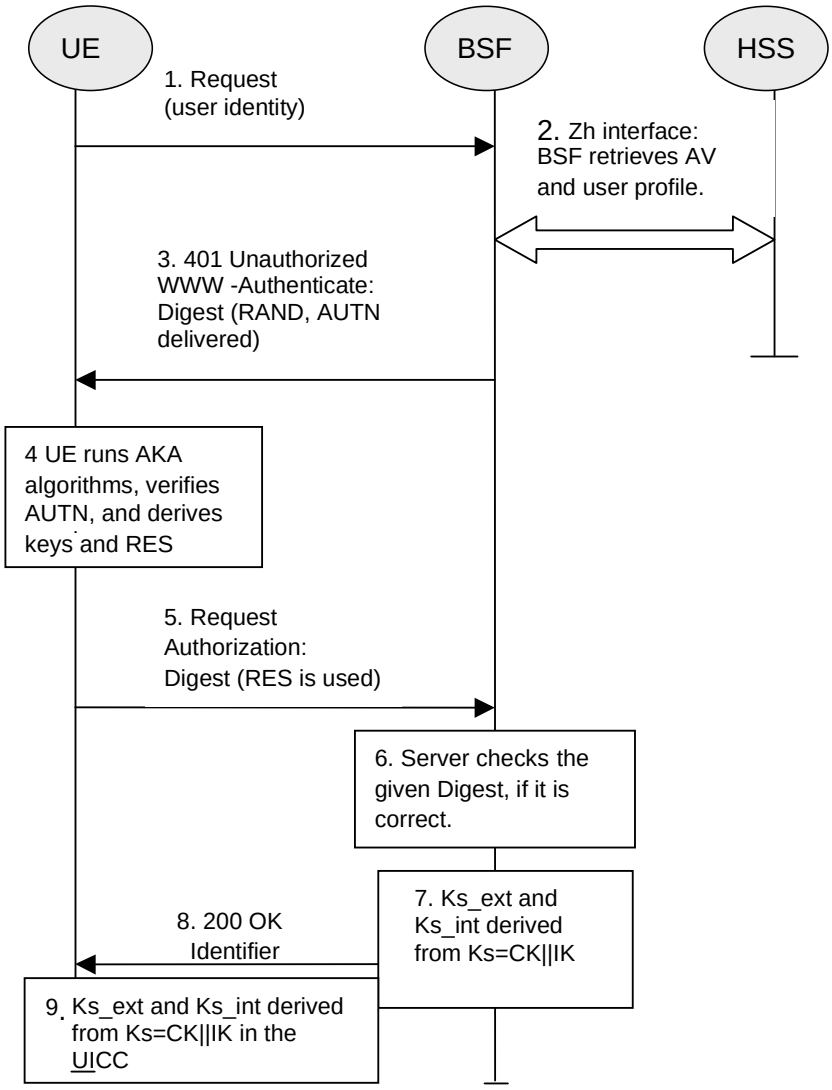


Figure 5.1: The bootstrapping procedure with UICC-based enhancements

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 Vector~~s~~
(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 □ 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 ⊕ 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 UICCcalculates IK and MAC (by performing MAC= MAC* ⊕ SHA-1(IK1)). Then the UICC checks AUTN(i.e. SQN ⊕ 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 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-key derivation parameters), and Ks_int_NAF is computed in the UICC as Ks_int_NAF = h2 (Ks_int, h2-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.

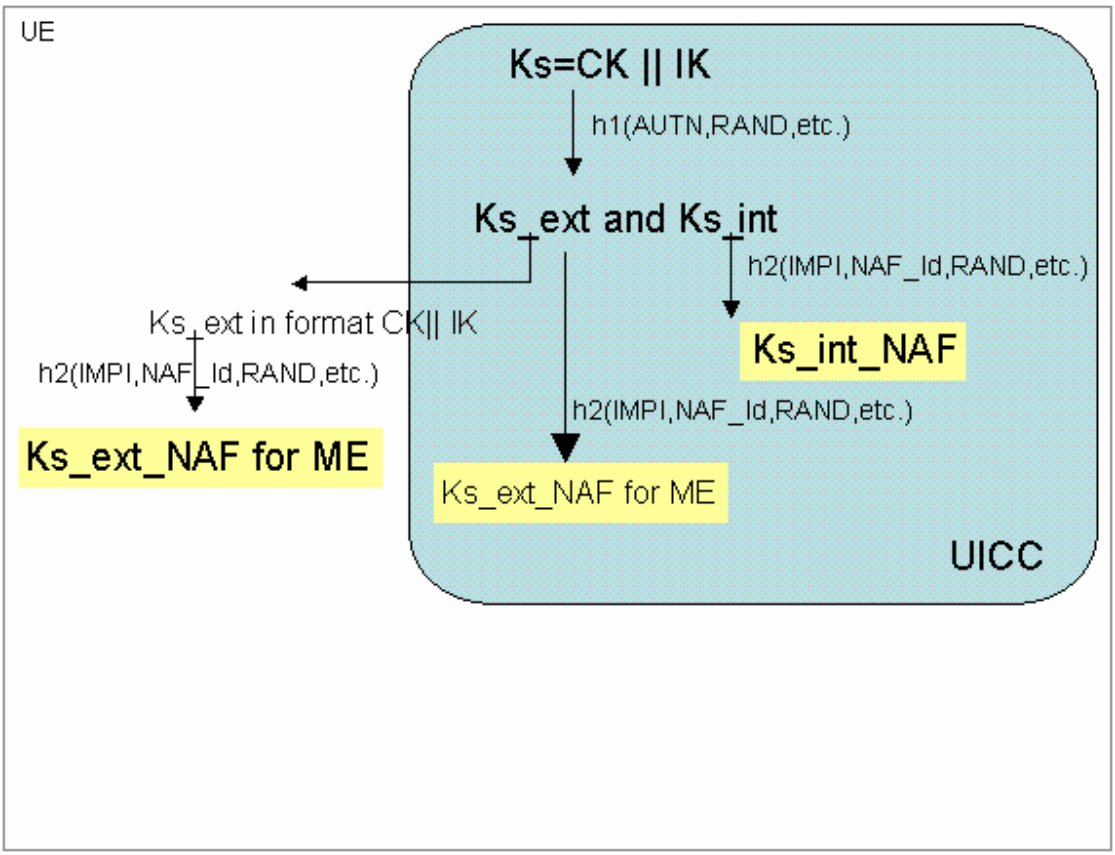


Figure 5.2: Key derivation for GBA-aware UICC when GBA-run was triggered

********** end change **********

3GPP TSG-SA WG3 Meeting #36
Shenzhen, China, 23 ñ 26 November 2004

Tdoc ⌘S3-041083

CR-Form-v7.1

CHANGE REQUEST

⌘ **33.220** **CR** **039** ⌘ **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 [X] Radio Access Network ☐ Core Network [X]

Title:	⌘	Clean up of TS 33.220
Source:	⌘	SA WG3
Work item code:	⌘	SEC1-SC
Date:	⌘	22/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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)*

Reason for change:	⌘	Cleaning up TS 33.220 for rel-6
Summary of change:	⌘	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.
Consequences if not approved:	⌘	Unclear specification.

Clauses affected:	⌘	4.5.2, 4.5.3, 5.3.2, 5.3.3

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

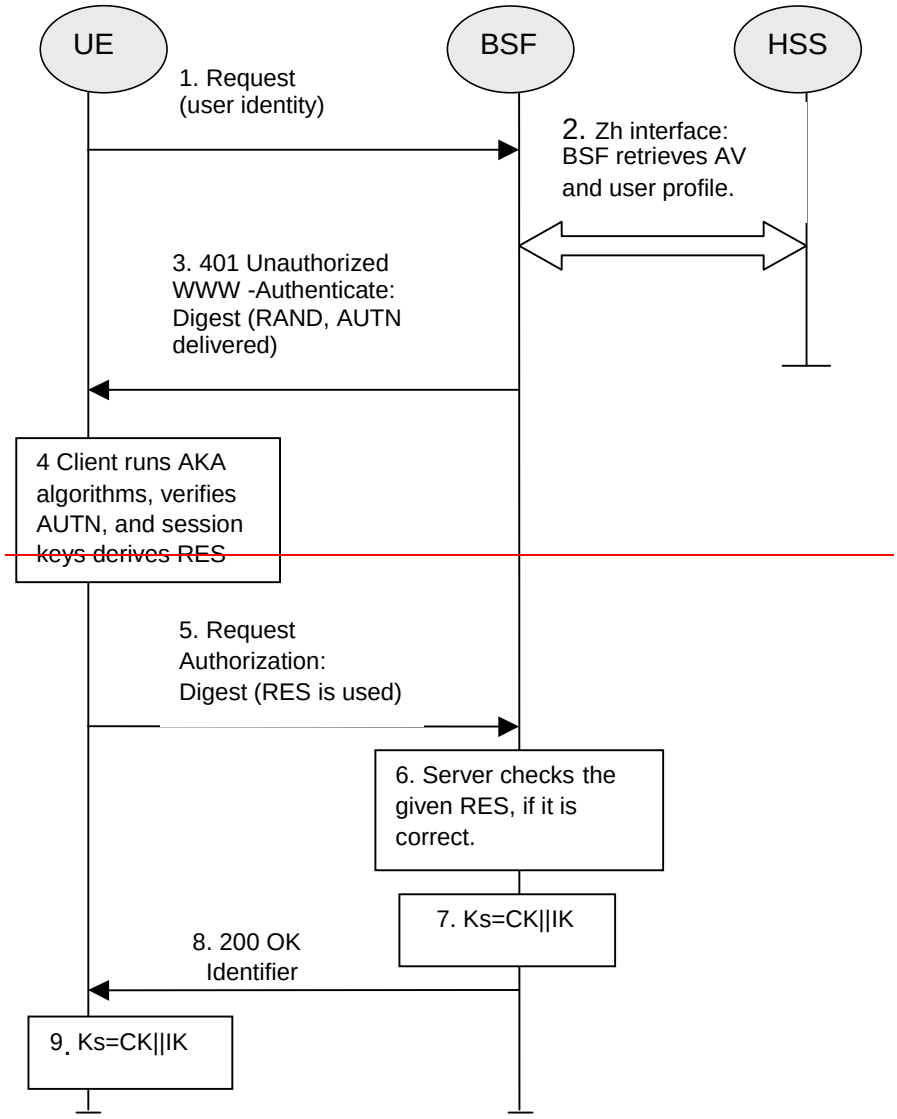
Other comments:	⌘	

***** 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.


UE
BSF
HSS
1. Request
(user identity)
2. Zh interface:
BSF retrieves AV
and user profile.
3. 401 Unauthorized
WWW -Authenticate:
Digest (RAND, AUTN
delivered)
4 Client runs AKA
algorithms, verifies
AUTN, and session
keys derives RES
5. Request
Authorization:
Digest (RES is used)
6. Server checks the
given RES, if it is
correct.
7. Ks=CK||IK
8. 200 OK
Identifier
9. Ks=CK||IK

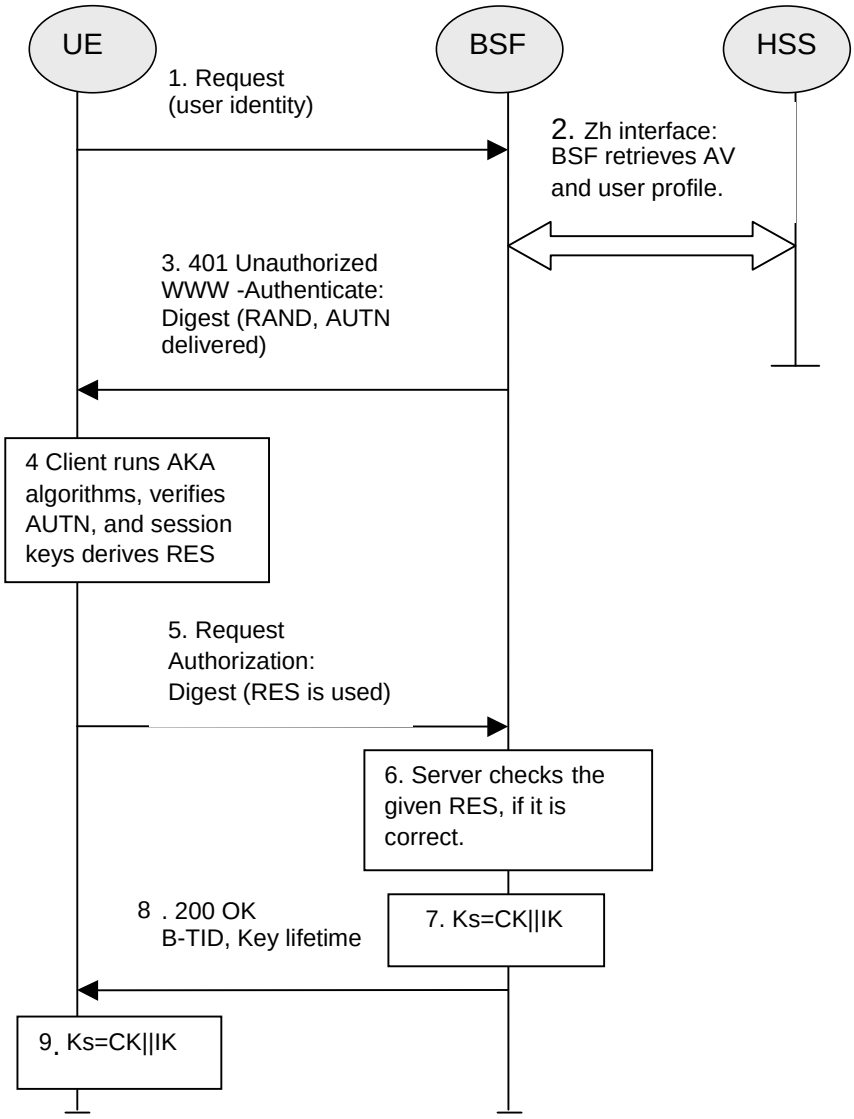


Figure 4.3: The bootstrapping procedure

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, 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 keyís 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 NAFís 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;

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.

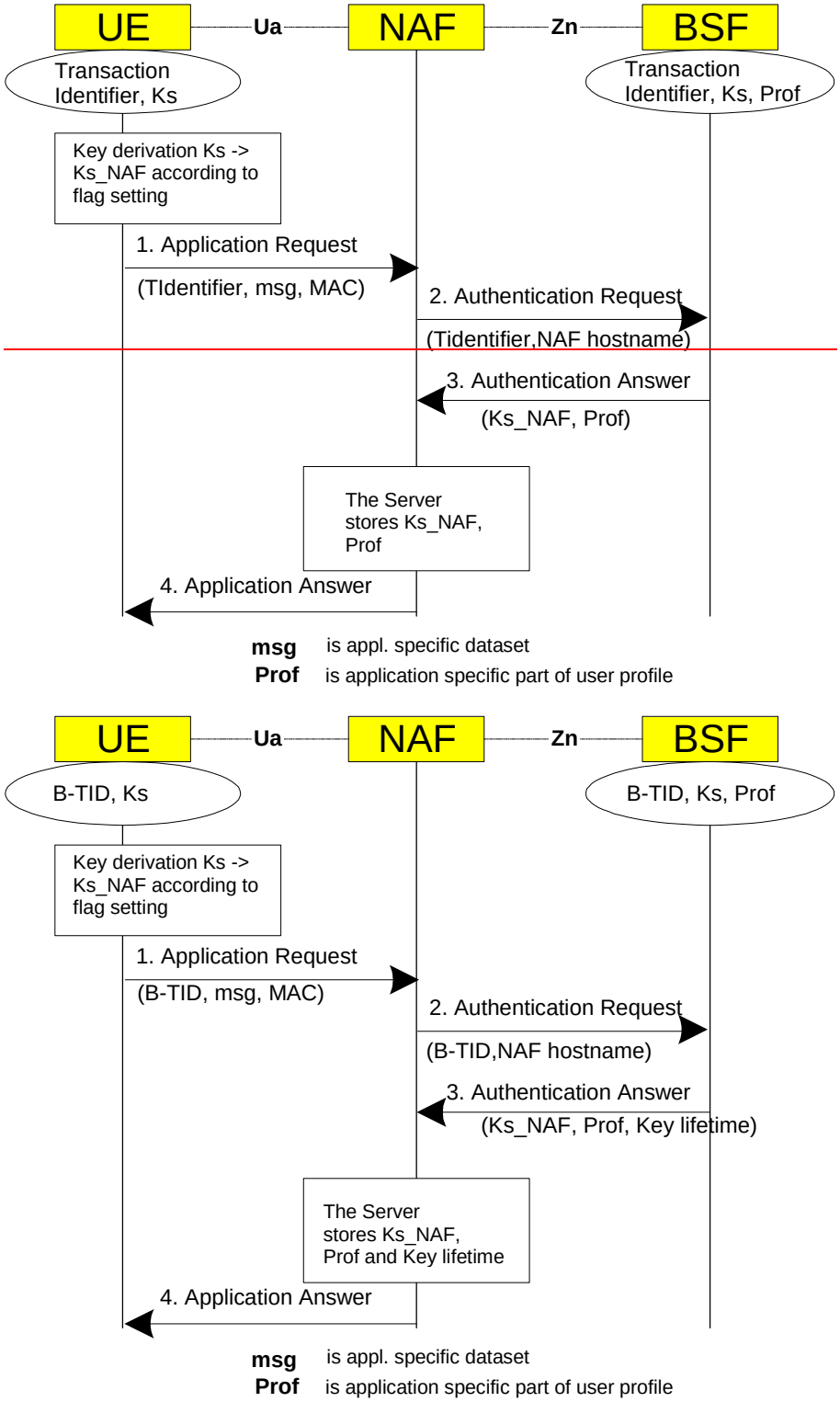


Figure 4.4: The bootstrapping usage procedure

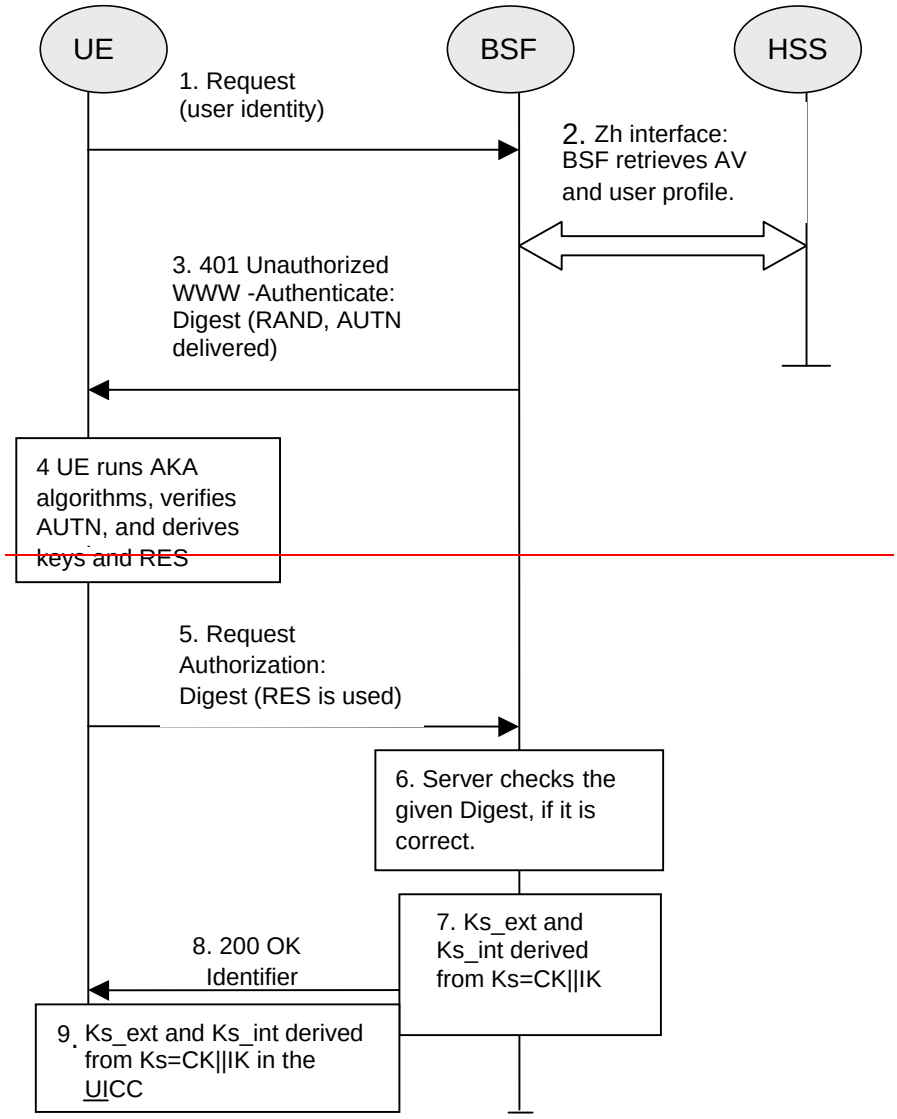
***** Next Change *****

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.


UE
BSF
HSS
1. Request
(user identity)
2. Zh interface:
BSF retrieves AV
and user profile.
3. 401 Unauthorized
WWW -Authenticate:
Digest (RAND, AUTN
delivered)
4 UE runs AKA
algorithms, verifies
AUTN, and derives
keys and RES
5. Request
Authorization:
Digest (RES is used)
6. Server checks the
given Digest, if it is
correct.
7. Ks_ext and
Ks_int derived
from Ks=CK||IK
8. 200 OK
Identifier
9. Ks_ext and Ks_int derived
from Ks=CK||IK in the
UICC

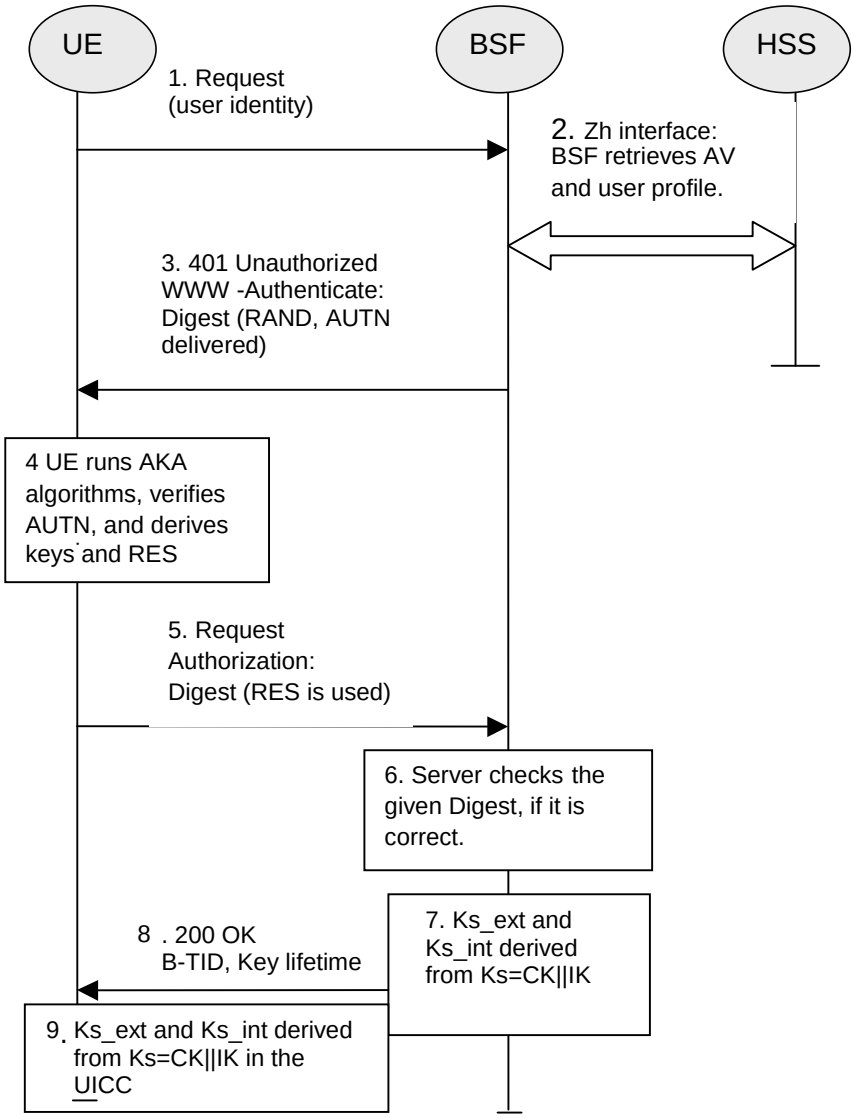


Figure 5.1: The bootstrapping procedure with UICC-based enhancements

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 □ 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 ⊕ 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 UICCcalculates IK and MAC (by performing MAC= MAC* ⊕ SHA-1(IK1)). Then the UICC checks AUTN(i.e. SQN ⊕ 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 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~~5. The ME sends another HTTP request, containing the Digest AKA response (calculated using RES), to the BSF.

~~7~~6. The BSF authenticates the UE by verifying the Digest AKA response.

~~8~~7. 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~~8. 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~~9. 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-key derivation parameters), and Ks_int_NAF is computed in the UICC as Ks_int_NAF = h2 (Ks_int, h2-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 NAFís 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.

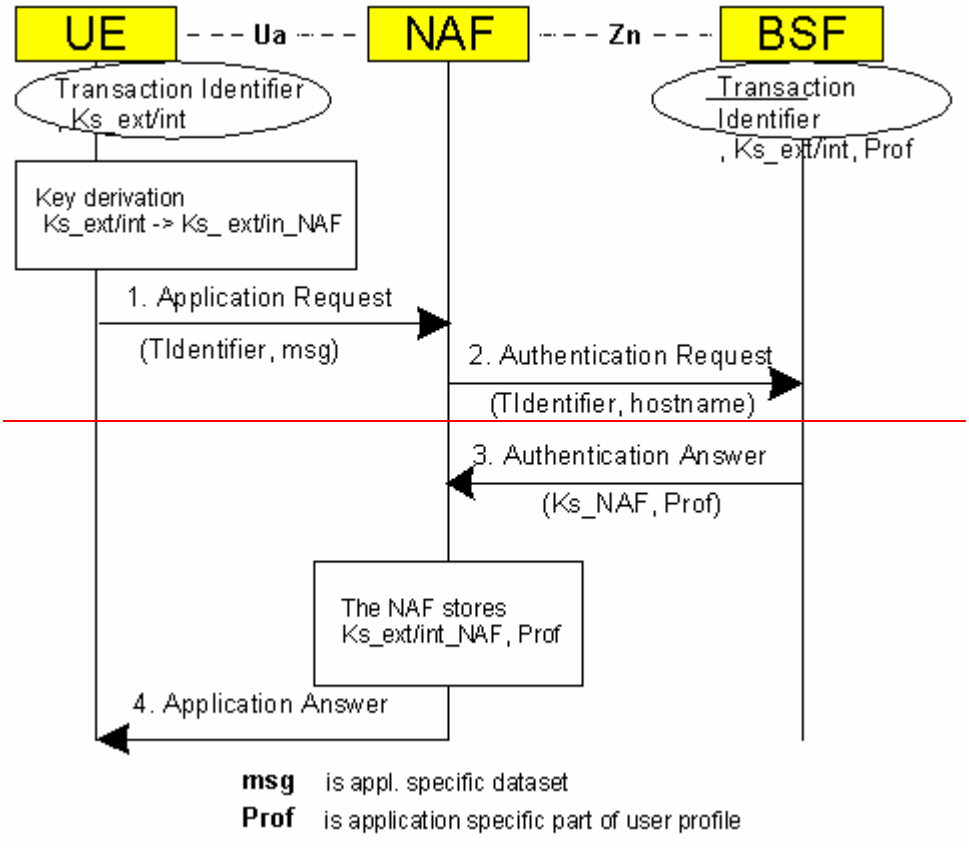
3. The BSF derives the keys Ks_ext_NAF, and Ks_int_NAF (if additionally required), as specified in clause 5.3.2. If the NAF indicated in its request that it is GBA_U aware, the BSF supplies to NAF both keys, Ks_ext_NAF, and Ks_int_NAF, otherwise the BSF supplies only Ks_ext_NAF. In addition, the BSF 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 BSFís 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.



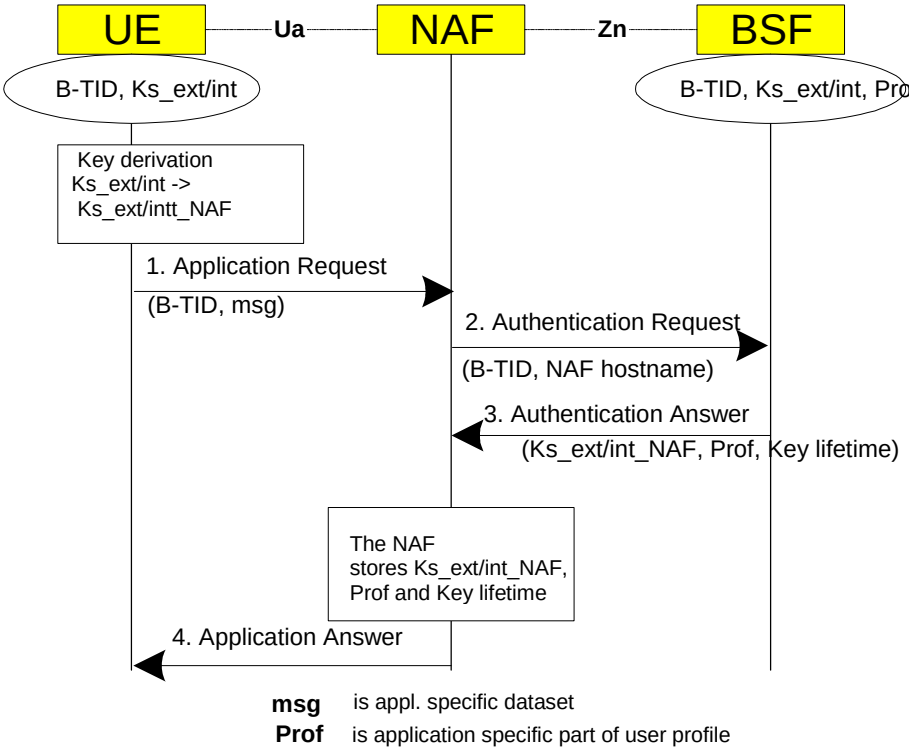


Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

3GPP TSG-SA WG3 Meeting S3#36
Shenzhen, China, November 23-26, 2004

Tdoc ⌘S3-041084

CR-Form-v7.1

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

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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)*

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:	⌘	**Y**	**N**		⌘
			X	Other core specifications	
			X	Test specifications	
			X	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 keyís 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 NAFís 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.

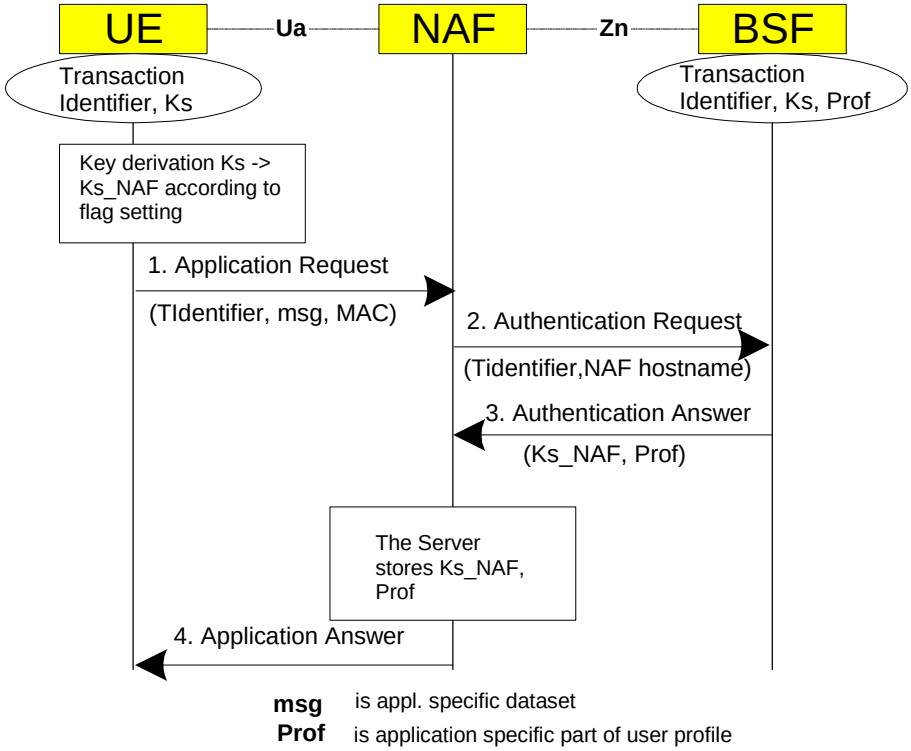


Figure 4.4: The bootstrapping usage procedure

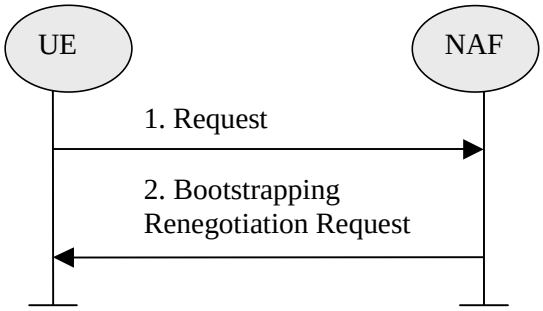


Figure 4.5: Bootstrapping renegotiation request

===== 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 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.

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 NAFís 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 BSFís 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.

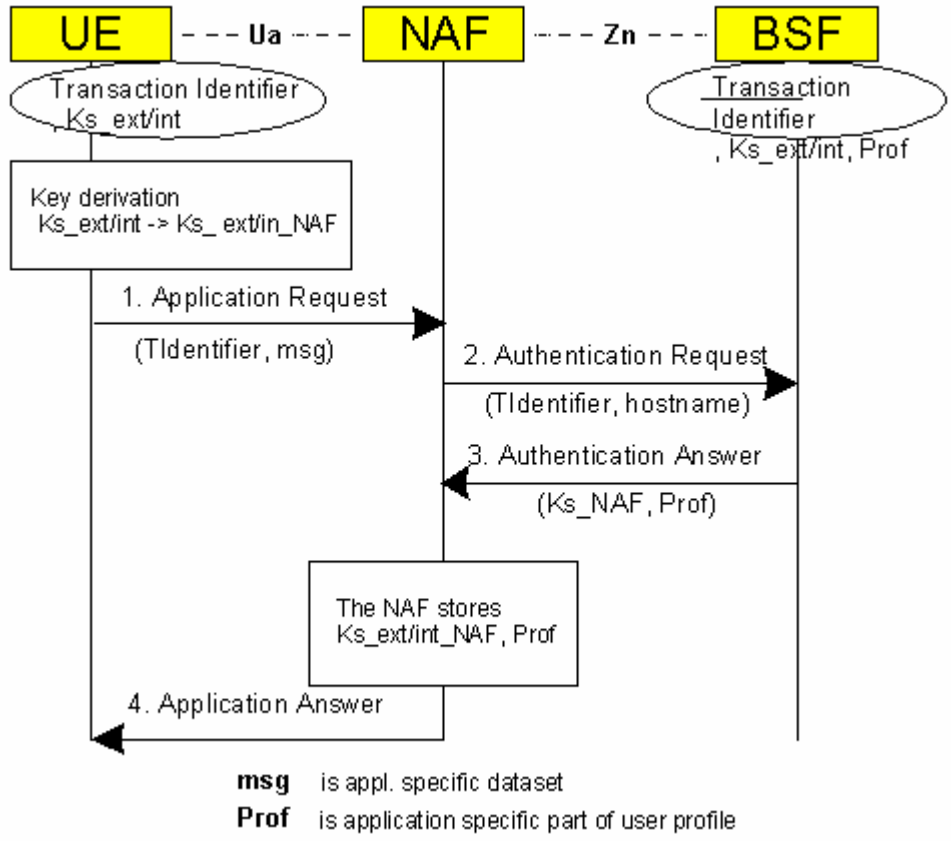


Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

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

3GPP TSG-SA WG3 Meeting #36
Shenzhen, China, November 23-26, 2004

Tdoc ⌘S3-041081

CR-Form-v7.1

CHANGE REQUEST

⌘ **33.220** **CR** **041** ⌘ **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⌘ [X] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Key derivation function
Source:	⌘	SA WG3
Work item code:	⌘	SEC1-SC
Date:	⌘	16/11/2004
Category:	⌘	**B**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
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)*

Reason for change:	⌘	SAGE has defined a key derivation function (KDF) in their LS to SA3 (S3-040914 updated in S3-040937). This CR implements the KDF to TS 33.220.
Summary of change:	⌘	Annex B contains the KDF as specified by SAGE. Subclauses 4.2.1 and 4.5.2 refer to Annex B.
Consequences if not approved:	⌘	The KDF is not specified.

Clauses affected:	⌘	2, 4.2.1, 4.5.2, Annex B

Other specs affected:	⌘	Y	N		⌘
			X	Other core specifications	
			X	Test specifications	
			X	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] 3GPP TS 31.103: "3rd Generation Partnership Project; Technical Specification Group Terminals; Characteristics of the IP Multimedia Services Identity Module (ISIM) application".

[11] 3GPP TS 23.003: "3rd Generation Partnership Project; Technical Specification Group Core Network; Numbering, addressing and identification".

[12] IETF RFC 3548 (2003): "The Base16, Base32, and Base64 Data Encodings".

[13] 3GPP TS 33.210: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Network domain security; IP network layer security".

[14] IETF RFC 3588 (2003): "Diameter Base Protocol".

[15] IETF RFC 2104 (1997): "HMAC: Keyed-Hashing for Message Authentication".

[16] ISO/IEC 10118-3:2004 Information Technology ñ Security techniques ñ Hash-functions ñ Part 3: Dedicated hash-functions

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

4.2.1 Bootstrapping server function (BSF)

A generic Bootstrapping Server Function (BSF) and the UE shall mutually authenticate using the AKA protocol, and agree on session keys that are afterwards applied between UE and an operator-controlled Network Application Function (NAF). The BSF shall restrict the applicability of the key material to a specific NAF by using ~~a suitable~~the key derivation procedure as specified in Annex B. The key derivation procedure may be used with multiple NAFs during the lifetime of the key material. The lifetime of the key material is set according to the local policy of the BSF. The generation of key material is specified in clause 4.5.2.

The BSF shall be able to acquire the GBA user security settings from the HSS.

===== 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.

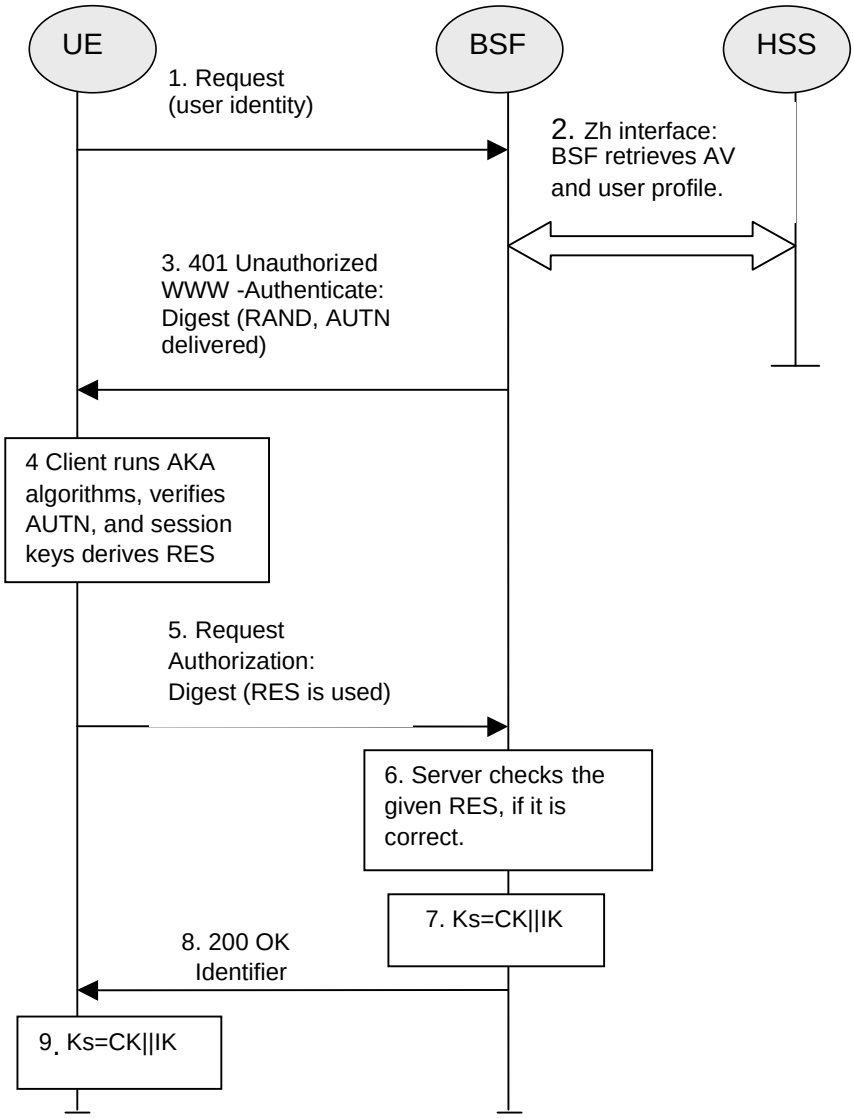


Figure 4.3: The bootstrapping procedure

1. The UE sends an HTTP request towards the BSF.

2. BSF retrieves the 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, "gba-me" || RAND || IMPI || NAF_Id~~key derivation parameters~~), where KDF is ~~a suitable~~the key derivation function as specified in Annex B, 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** =====

Annex B (normative): Specification of the key derivation function KDF

~~Editor's note: The definition of the KDF and the possible inclusion of further key derivation parameters is left to ETSI SAGE.~~

B.1 Introduction

This annex specifies the key derivation function (KDF) that is used in the NAF specific key derivation in both GBA (i.e., GBA_ME) and GBA_U. The key derivation function defined in the annex takes the following assumptions:

1. the input parameters to the key derivation functions are octet strings - not bit strings of arbitrary length:
2. a single input parameter will have lengths no greater than 65535 octets.

B.2 Generic key derivation function

The input parameters and their lengths shall be concatenated into a string S as follows:

1. The length of each input parameter in octets shall be encoded into two-octet string:

   a) express the number of octets in input parameter Pi as a number l in the range $0 \leq l \leq 65535$.

   b) Li is then a two-octet representation of the number l, with the most significant bit of the first octet of Li equal to the most significant bit of l, and the least significant bit of the second octet of Li equal to the least significant bit of l,

   Example: If Pi contains 258 octets then Li will be the two-octet string 0x01 0x02.

2. String S shall be constructed from n input parameters as follows:

   S = FC || P0 || L0 || P1 || L1 || P2 || L2 || P3 || L3 ||... || Pn || Ln

   where

   FC is single octet used to distinguish between different instances of the algorithm,

   P0 is a static ASCII-encoded string,

   L0 is the two octet representation of the length of the P0,

   P1 ... Pn are the n input parameters, and

   L1 ... Ln are the two-octet representations of the corresponding input parameters.

3. The final output, i.e., the derived key is equal to HMAC-SHA-256 (as specified in [15] and [16]) computed on the string S using the key Key:

   derived key = HMAC-SHA-256 (Key , S)

B.3 NAF specific key derivation in GBA and GBA_U

In GBA and GBA_U, the input parameters for the key derivation function shall be the following:

- FC = 0x01,
- P1 = RAND,

- L1 = length of RAND is 16 octets (i.e., 0x00 0x10),
- P2 = IMPI,
- L2 = length of IMPI is variable (not greater that 65535),
- P3 = NAF_ID, and
- L3 = length of NAF_ID is variable (not greater that 65535).

In the key derivation of Ks_NAF as specified in clause 4 and Ks_ext_NAF as specified in clause 5,

- P0 = "gba-me" (i.e., 0x67 0x62 0x61 0x2d 0x6d 0x65), and
- L0 = length of P0 is 6 octets (i.e., 0x00 0x06).

In the key derivation of Ks_int_NAF as specified in clause 5,

- P0 = "gba-u" (i.e., 0x67 0x62 0x61 0x2d 0x75), and
- L0 = length of P0 is 5 octets (i.e., 0x00 0x05).

The Key to be used in key derivation shall be:

- Ks (i.e., CK || IK concatenated) as specified in clauses 4 and 5,

NOTE: In the specification this function is denoted as:
Ks_NAF = KDF (Ks, "gba-me" || RAND || IMPI || NAF_Id),
Ks_ext_NAF = KDF (Ks, "gba-me" || RAND || IMPI || NAF_Id), and
Ks_int_NAF = KDF (Ks, "gba-u" || RAND || IMPI || NAF_Id).

===== **END CHANGE** =====

3GPP TSG SA WG3 Security ó S3#36 **S3-041140**
November 23-26, 2004, Shenzhen, China

CR-Form-v7

CHANGE REQUEST

⌘ **33.220** **CR** **042** ⌘**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 **X** Radio Access Network ☐ Core Network **X**

Title:	⌘	Re-negotation of keys
Source:	⌘	SA WG3
Work item code:	⌘	GBA
Date:	⌘	25/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one 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:	⌘	The current specification mandates that the Ua protocol is terminated when a key is updated. This is too restrictive. Termination is only required when the key lifetime has expired.
Summary of change:	⌘	Add clarifying text
Consequences if not approved:	⌘	Unnecessary restriction, potential interruption of service.

Clauses affected:	⌘	4.5.3, 5.3.3			
Other specs affected:	⌘	**Y**	**N**		⌘
			X	Other core specifications	
			X	Test specifications	
			X	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 keyís lifetime has expired or will expire soon, it shall send a suitable bootstrapping renegotiation request to the UE ~~and terminates the protocol used over reference point Ua~~, see figure 4.5. If the keyís lifetime has expired the protocol used over reference point Ua shall be terminated. 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;

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 NAFís 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.

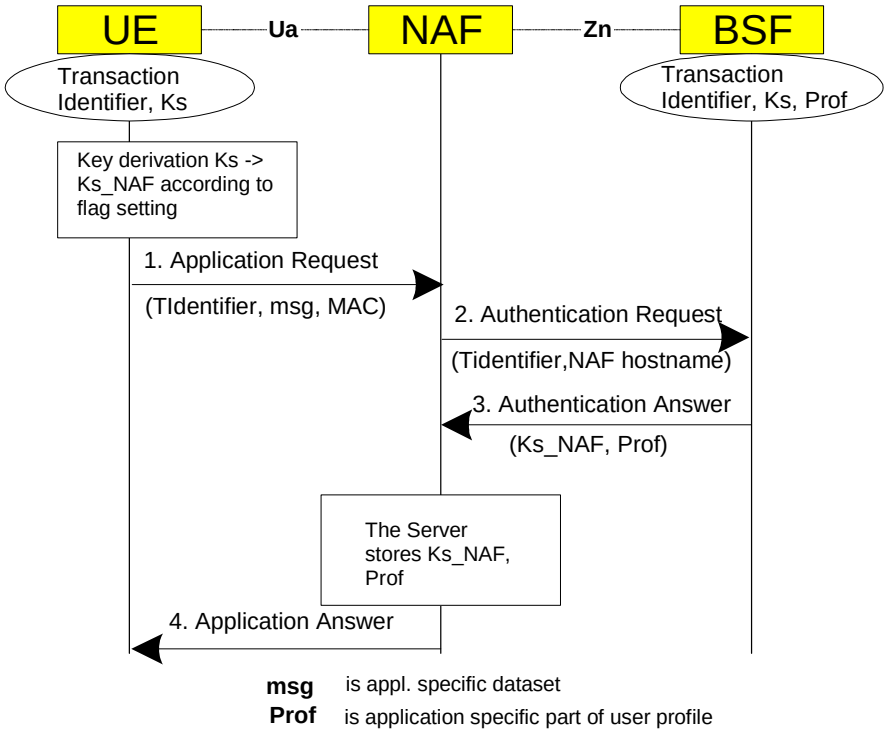


Figure 4.4: The bootstrapping usage procedure

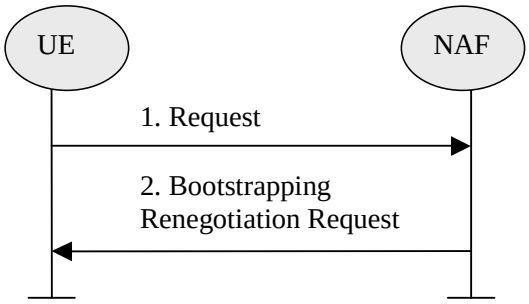


Figure 4.5: Bootstrapping renegotiation request

********** **end change** **********

********** **begin 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~~. If the keyís lifetime has expired the protocol used over reference point Ua shall be terminated. 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 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.

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 NAFís 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 BSFís 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.

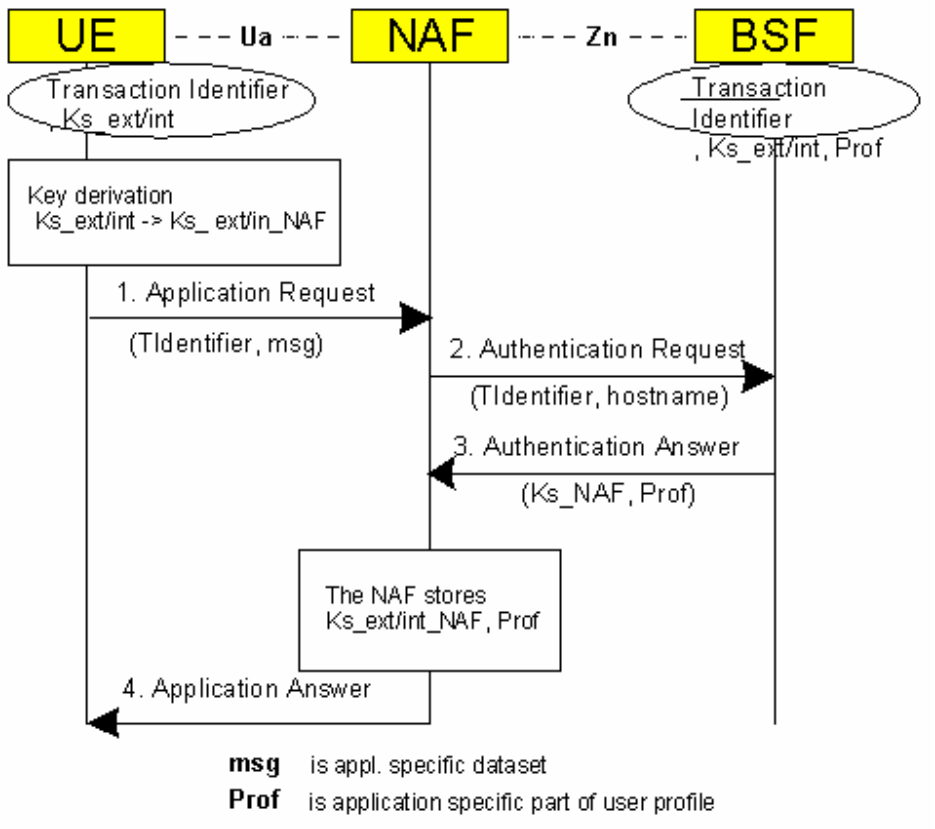


Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

********** end change **********

3GPP TSG-SA WG3 Meeting #36
Shenzhen, China, November 23-26, 2004

Tdoc ⌘S3-041089

CR-Form-v7.1

CHANGE REQUEST

⌘ **33.220** **CR** **043** ⌘ **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:	⌘	No GUSS/USS update procedures in Release-6
Source:	⌘	SA WG3
Work item code:	⌘	GBA-SSC
Date:	⌘	16/11/2004
Category:	⌘	**C**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
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)*

Reason for change:	⌘	The possible GUSS/USS update procedures are mentioned in editor's notes. As the Release-6 is freezing and no studies have been made regarding the details and possible implications of GUSS/USS update procedure, the update procedure is postponed to Release-7. In Release-6, the GUSS in the BSF is updated as part of the bootstrapping procedure with the BSF, and USSs in the NAF when it is fetching a new NAF specific key over Zn reference point.
Summary of change:	⌘	- 4.4.5 and 4.4.6: GUSS update in BSF and USS update in NAF are done by using the existing method, i.e., the BSF gets the updated version of the GUSS when it next time fetches the authentication vectors and GUSS from the HSS, or when NAF fetches a new USS from the BSF when it receives a new B-TID from the UE. The possible update procedure initiated by the HSS may be defined in future releases.
Consequences if not approved:	⌘	

Clauses affected:	⌘	4.4.5, 4.4.6

Other specs affected:	⌘	Y	N		⌘
			X	Other core specifications	
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

==== *BEGIN CHANGE* ====

4.4.5 Requirements on reference point Zh

The requirements for reference point Zh are:

- mutual authentication, confidentiality and integrity shall be provided;

NOTE 1: This requirement may be fulfilled by physical or proprietary security measures if BSF and HSS are located within the same operatorís network.

- the BSF shall be able to send bootstrapping information request concerning a subscriber;
- the HSS shall be able to send 3GPP AKA vectors to the BSF in batches;
- the HSS shall be able to send the complete set of subscriber's GBA user security settings needed for security purposes to the BSF;

~~Editor's note: It's ffs how to proceed in the case where GBA user security settings are updated in HSS after GBA user security settings were forwarded. The question is whether this profile change should be propagated to BSF.~~

NOTE 2: If subscriber's GUSS is updated in HSS, this is not propagated to the BSF. The GUSS in the BSF is updated when the BSF next time fetches the authentication vectors and GUSS from the HSS over Zh reference point as part of the bootstrapping procedure.

- no state information concerning bootstrapping shall be required in the HSS;
- all procedures over reference point Zh shall be initiated by the BSF;

~~Editorís note: This requirement may need to be modified depending on what happens in the case where the GBA user security settings in the HSS is updated.~~

- the number of different interfaces to HSS should be minimized.

4.4.6 Requirements on reference point Zn

The requirements for reference point Zn are:

- mutual authentication, confidentiality and integrity shall be provided;
- If the BSF and the NAF are located within the same operator's network, the Zn reference point shall be secured according to NDS/IP [13];
- If the BSF and the NAF are located in different operators' networks, the Zn' reference point between the D-Proxy and the BSF shall be secured using TLS as specified in RFC 2246 [6];

Editorís Note: The TLS Certificate profiling needs to be completed and will be added into an Annex.

- 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 a selected set of application-specific user security settings from the BSF, depending on the policy of the BSF and the application indicated in the request from the NAF over Zn;
- The NAF shall be able to indicate to the BSF the single application or several applications it requires user security settings for;

NOTE 1: If some application needs only a subset of an application-specific user security setting, e.g. only one IMPU, the NAF selects this subset from the complete set of user security settings sent from BSF.

- The BSF shall be able to configure on a per NAF or per application basis if private subscriber identity and which user security settings may be sent to a NAF;

- The BSF shall be able to indicate to the NAF the lifetime of the key material. The key lifetime sent by the BSF over Zn shall indicate the expiry time of the key, and shall be identical to the key lifetime sent by the BSF to the UE over Ub.

NOTE 2: This does not preclude a NAF to refresh the key before the expiry time according to the NAF's local policy.

~~Editor's note: It is ffs which actions are to be taken over Zn when the BSF receives a user security settings update from the HSS over Zh.~~

NOTE 3: If one or more of the USSs that have been delivered to the NAF has been updated in subscriber's GUSS in the HSS, this change is propagated to the NAF the next time it fetches the USS from the BSF over Zn reference point (provided that the BSF has updated subscriber's GUSS from the HSS over Zh reference point).

==== ***END CHANGE*** ====

3GPP TSG SA WG3 Security ó S3#36 **S3-041137**
Shenzhen, 23 ñ 26 November 2004

CR-Form-v7

CHANGE REQUEST

⌘ **33.220** **CR** **044** ⌘ **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⌘ [X] ME [X] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Clarify the number of NAF-specific keys stored in the UE per NAF-Id
Source:	⌘	SA WG3
Work item code:	⌘	SEC1-SC
Date:	⌘	25/11/2004
Category:	⌘	**D**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one 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:	⌘	The current version of the TS does not explicitly indicate that there is at most one key Ks_NAF for one NAF_Id stored in the UE at a given time
Summary of change:	⌘	Add a clarifying note on the number of NAF-specific stored in the UE per NAF-ID
Consequences if not approved:	⌘	Readers may be unclear about the number of NAF-specific key that can be stored on the UE per NAF-Id.

Clauses affected:	⌘	4.5.3; 5.5.3

	Y	N			
Other specs affected:	X		Other core specifications	⌘	TS 31.102
		X	Test specifications		
		X	O&M Specifications		

Other comments:	⌘	-

BEGIN OF 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 keyís 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.

NOTE 5: According to the procedures defined in sections 4.5.2 and 4.5.3, in the UE there is at most one Ks_NAF key stored per NAF-Id.

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 NAFís 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~~6: 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.

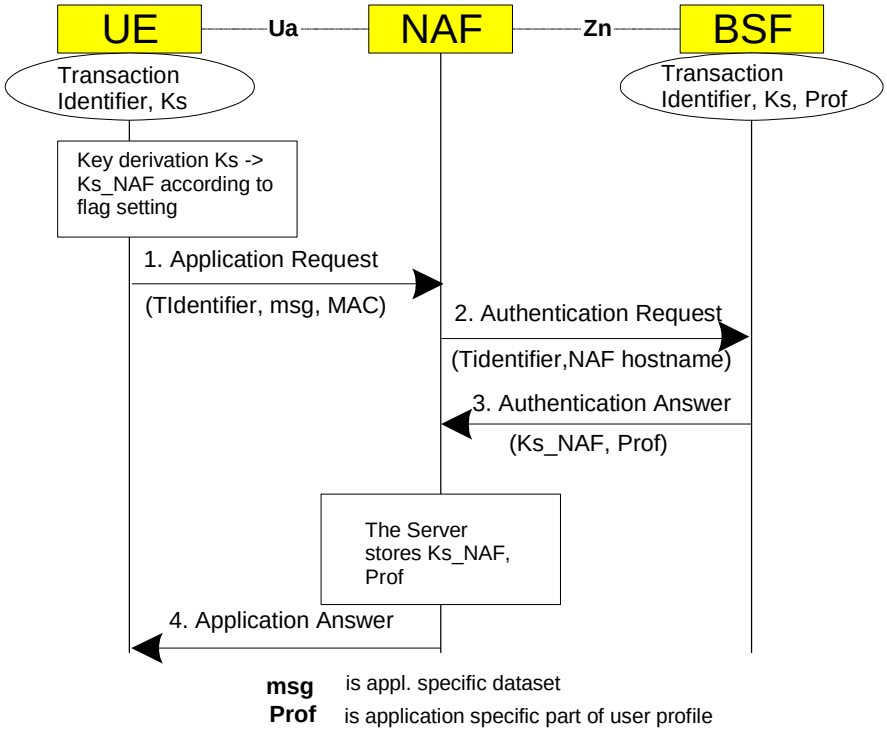


Figure 4.4: The bootstrapping usage procedure

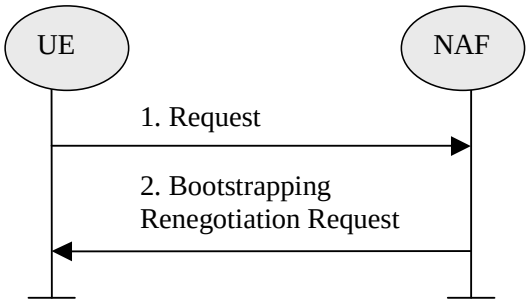


Figure 4.5: Bootstrapping renegotiation request

END OF CHANGE

BEGIN OF 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.

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: According to the procedures defined in sections 5.3.2 and 5.3.3, in the UE there is at most one Ks_int_NAF/Ks_ext_NAF key pair stored per NAF_Id.

NOTE 9~~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 NAFís 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 BSFís 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.

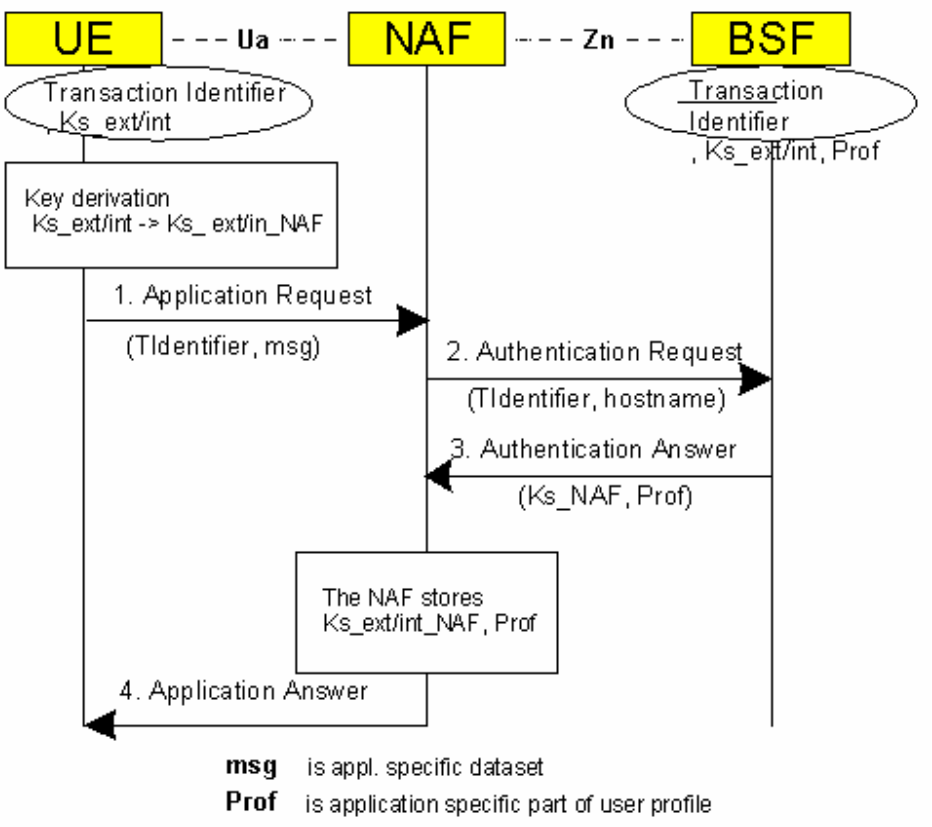


Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

END OF CHANGE