3GPP TSG SA WG3 Security – 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.

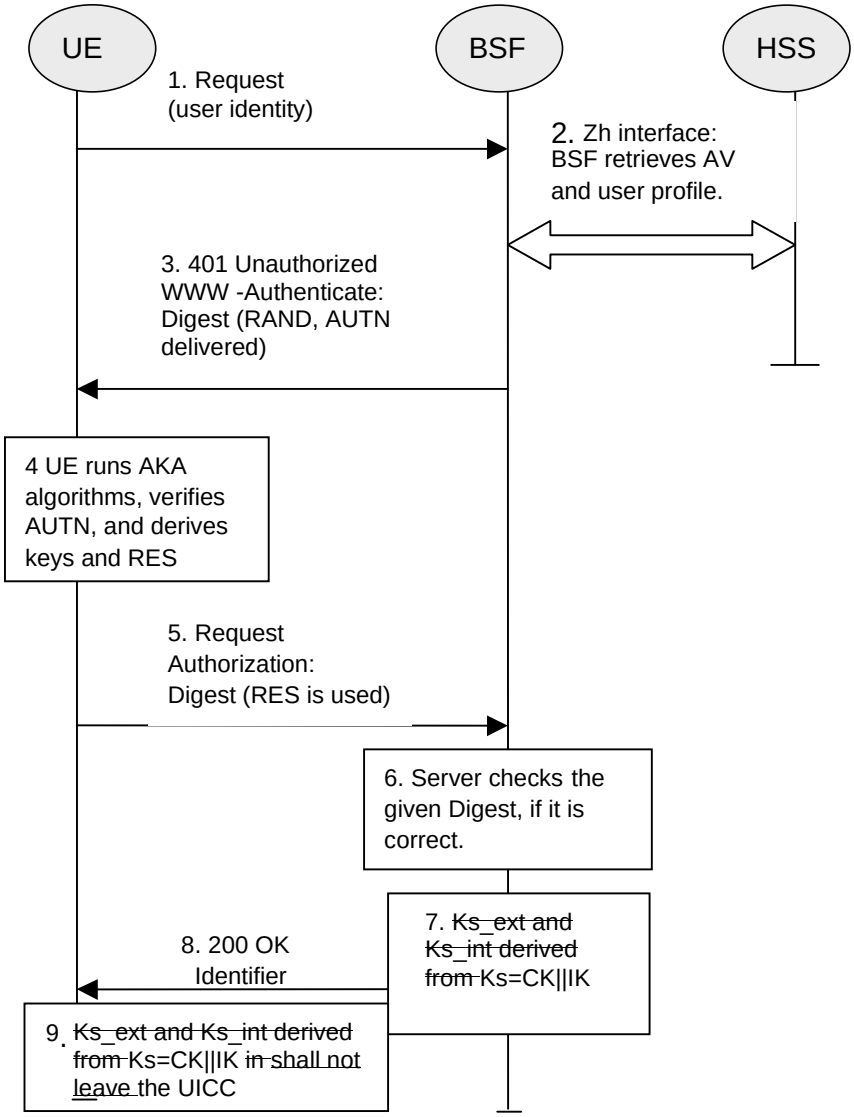


Figure 5.1: The bootstrapping procedure with UICC-based enhancements

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~~.

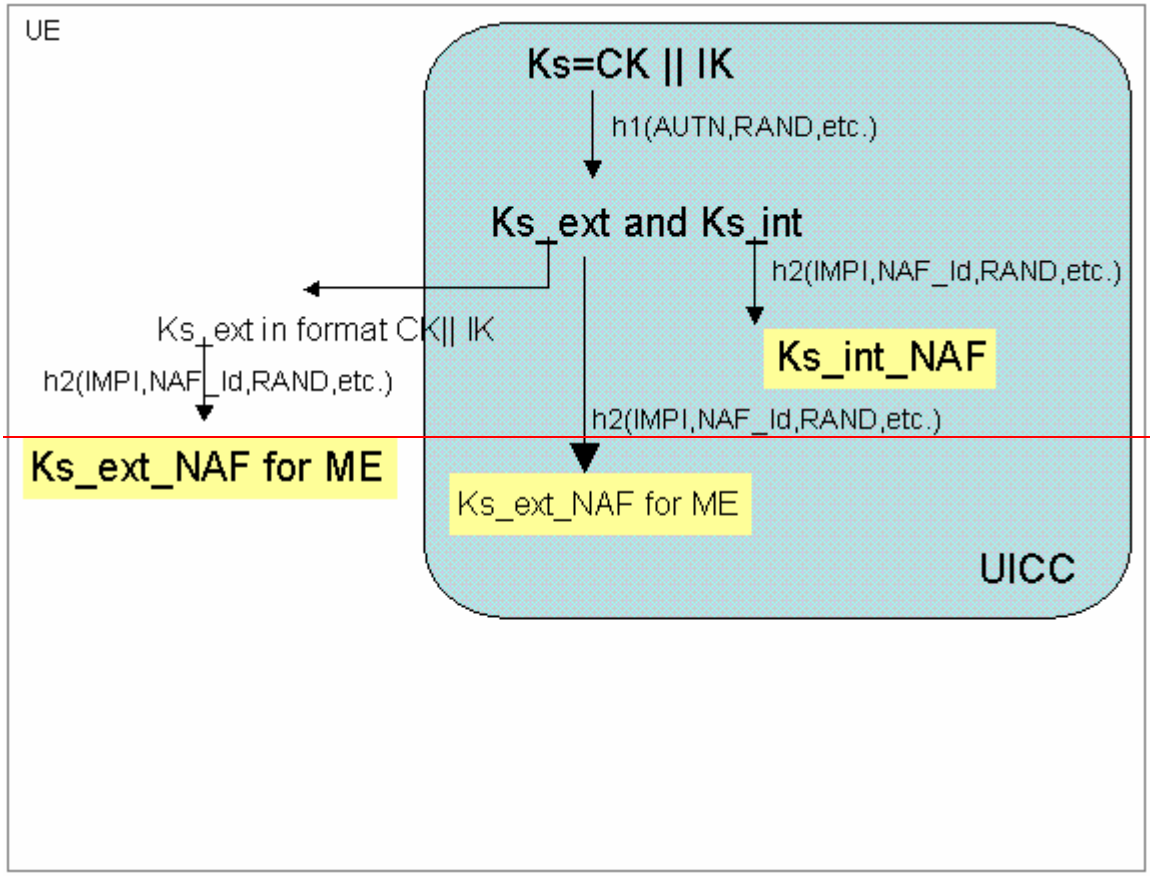
~~10~~9. Both the UICC and the BSF shall use the Ks to derive NAF-specific keys Ks_ext_NAF and Ks_int_NAF during the procedures as specified in clause 5.3.3, if applicable. ~~The BSF shall use the keys Ks_ext and Ks_int to derive the NAF specific keys Ks_ext_NAF and Ks_int_NAF, if requested by a NAF 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 NAF's public hostname that UE has used to access NAF to BSF, and BSF shall be able to verify that NAF is authorized to use that hostname.

- The BSF derives the keys Ks_ext_NAF, and Ks_int_NAF (if additionally required), as specified in clause 5.3.2. If the NAF indicated in its request that it is GBA_U aware, the BSF supplies to NAF both keys, Ks_ext_NAF, and Ks_int_NAF, otherwise the BSF supplies only Ks_ext_NAF. In addition, the BSF supplies the lifetime time of these keys. If the key identified by the B-TID supplied by the NAF is not available at the BSF, the BSF shall indicate this in the reply to the NAF. The NAF then indicates a bootstrapping renegotiation request (See figure 4.5) to the UE;

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

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

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

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

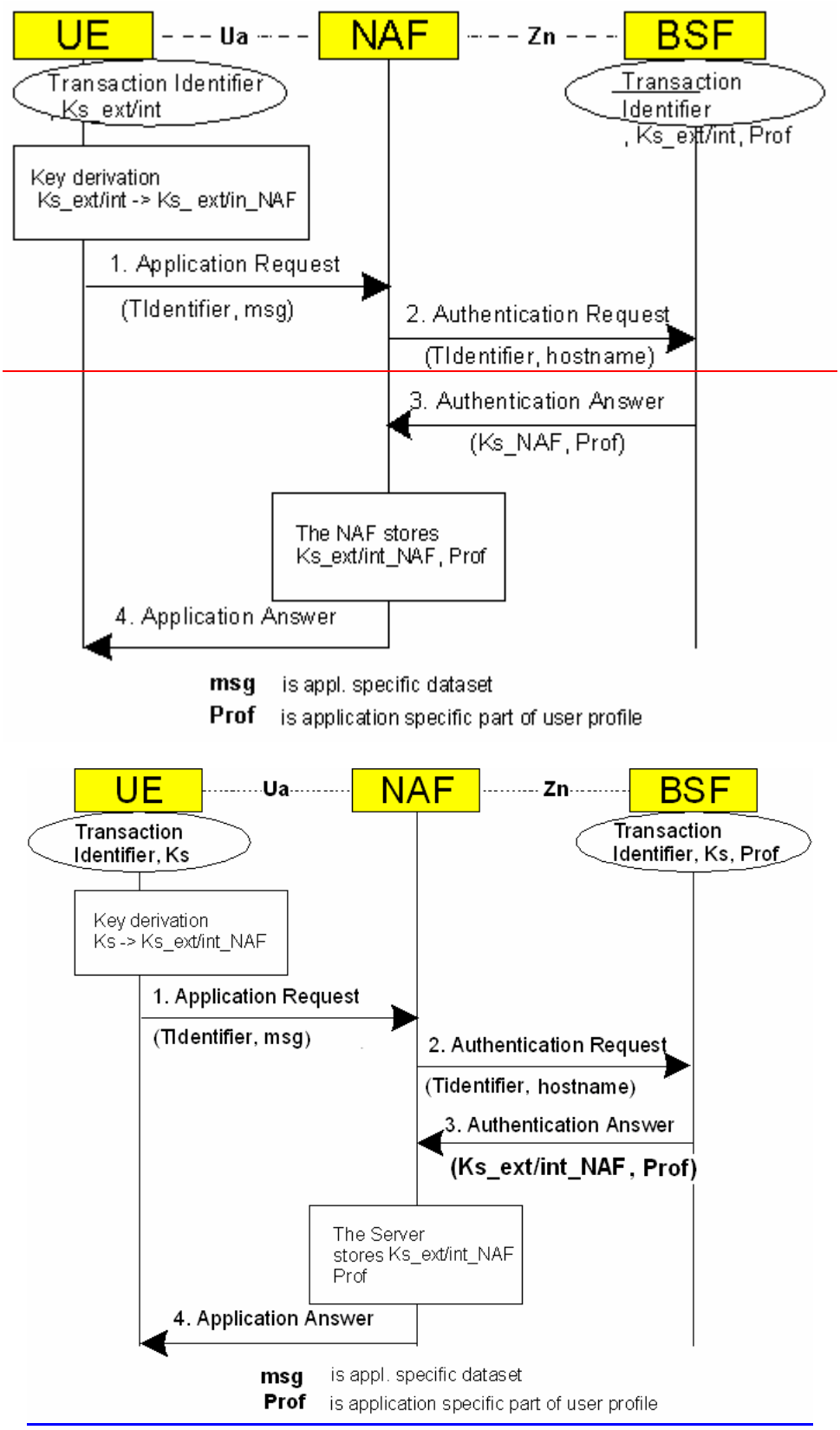


Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

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.

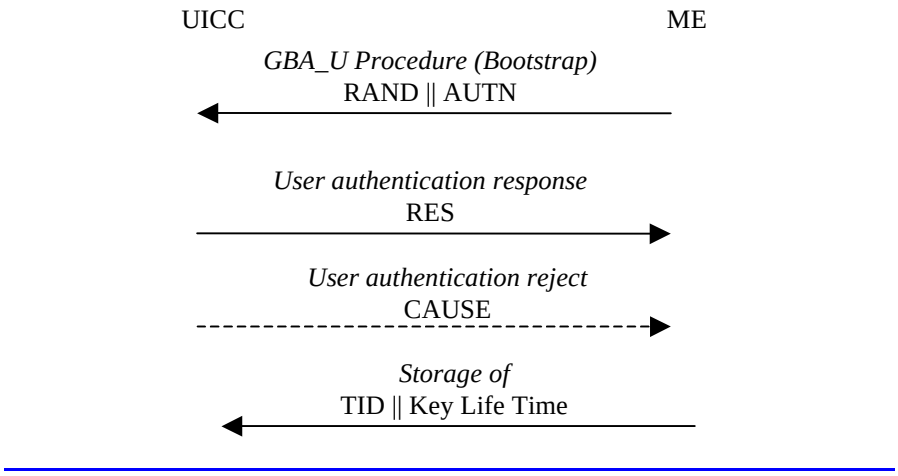


Figure x: GBA_U Bootstrap Procedure

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.

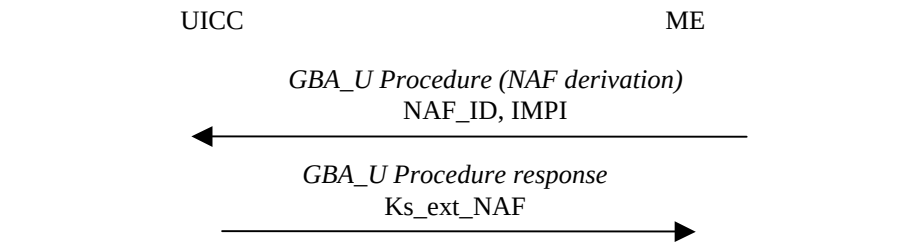


Figure x: GBA_U NAF derivation procedure