3GPP TSG SA WG3 Security — SA3#33 **S3-040217**
10-14 May 2004, Beijing, China

CR-Form-v7

CHANGE REQUEST

⌘ **33.220** **CR** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*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:	⌘	Introduction of a UICC-based Generic Bootstrapping Architecture
Source:	⌘	Siemens
Work item code:	⌘	SSC-GBA
Date:	⌘	06/04/2004
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:	⌘	Addition of a UICC-based Generic Bootstrapping mechanism
Summary of change:	⌘	
Consequences if not approved:	⌘	The feature cannot be used.

Clauses affected:	⌘	- Introduction of new section 5 of TS 33.220 v6.0.0 - Addition of new definitions and abbreviation to section 3

	Y	N			
Other specs affected: ⌘		X	Other core specifications	⌘	TS 24.109, 29.229, 31.102
		X	Test specifications		
		X	O&M Specifications		

Other comments:	⌘	-

BEGIN OF CHANGE

---

1 Scope

The present document describes the security features and a mechanism to bootstrap authentication and key agreement for application security from the 3GPP AKA mechanism. Candidate applications to use this bootstrapping mechanism include but are not restricted to subscriber certificate distribution TS 33.221 [5]. Subscriber certificates support services whose provision mobile operator assists, as well as services that mobile operator provides.

The scope of this specification includes a generic AKA bootstrapping function, an architecture overview and the detailed procedure how to bootstrap the credential.

Section 4 of this specification describes a mechanism, called GBA_ME, to bootstrap authentication and key agreement, which does not require any changes to the UICC. Section 5 of this specification describes a mechanism, called GBA_U, to bootstrap authentication and key agreement, which does require changes to the UICC, but provides enhanced security by storing certain derived keys on the UICC.

NOTE: The specification objects are scheduled currently in phases. For this specification release, only the case is considered where bootstrapping server functionality and network application function are located in the same network as the HSS. In further specification release, other configurations may be considered.

END OF CHANGE

BEGIN OF CHANGE

3 Definitions and abbreviations

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.

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

Editor's note: Definition to be completed.

Transaction Identifier:

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, cf. section 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, cf. section 5 of this specification.

3.2 Abbreviations

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

AK	Anonymity Key
AKA	Authentication and Key Agreement
BSF	Bootstrapping Server Function
CA	Certificate Authority
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	Derived key in GBA_U which remains on UICC
Ks_ext	Derived key in GBA_U, which is transferred from UICC to ME
MNO	Mobile Network Operator
NAF	Network Application Function
PKI	Public Key Infrastructure

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 section are capable of handling the GBA_U specific enhancements. For issues of migration from UICC, BSF, and HSS, which are not GBA_U - aware, see Annex Dsection 5.4 of this specificationdocument. The procedures specified in this section also apply if ME and NAF are not GBA_U aware, but, of course, in that case there are no benefits of the GBA_U specific enhancements.

5.1 Requirements and principles for bootstrapping with UICC-based enhancements

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

5.1.1 Requirements on UE

The 3G AKA keys CK and IK resulting from a run of the protocol over the Ub interface 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. One of these derived keys shall be transferred to the ME, the other key shall be stored on the UICC. 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.

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

The text from section 4.4 of this specification document applies also here, with the addition that the interface between the ME and the USIM, as specified in TS 31.102 [1], needs to be enhanced with GBA_U specific commands. The requirements on these commands can be found in section 5.1.1, details on the procedures in section 5.3.

5.3 Procedures for bootstrapping with UICC-based enhancements

5.3.1 Initiation of bootstrapping

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

5.3.2 Bootstrapping procedure

The procedure specified in this section differs from the procedure specified section 4.5.2 in the generation of the random challenge in the HSS and the local handling of keys in the UE and the BSF. The messages exchanged over the Ub interface 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 key update indication from the NAF, or when the lifetime of the key in UE has expired (cf. subclause 5.3.3).

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

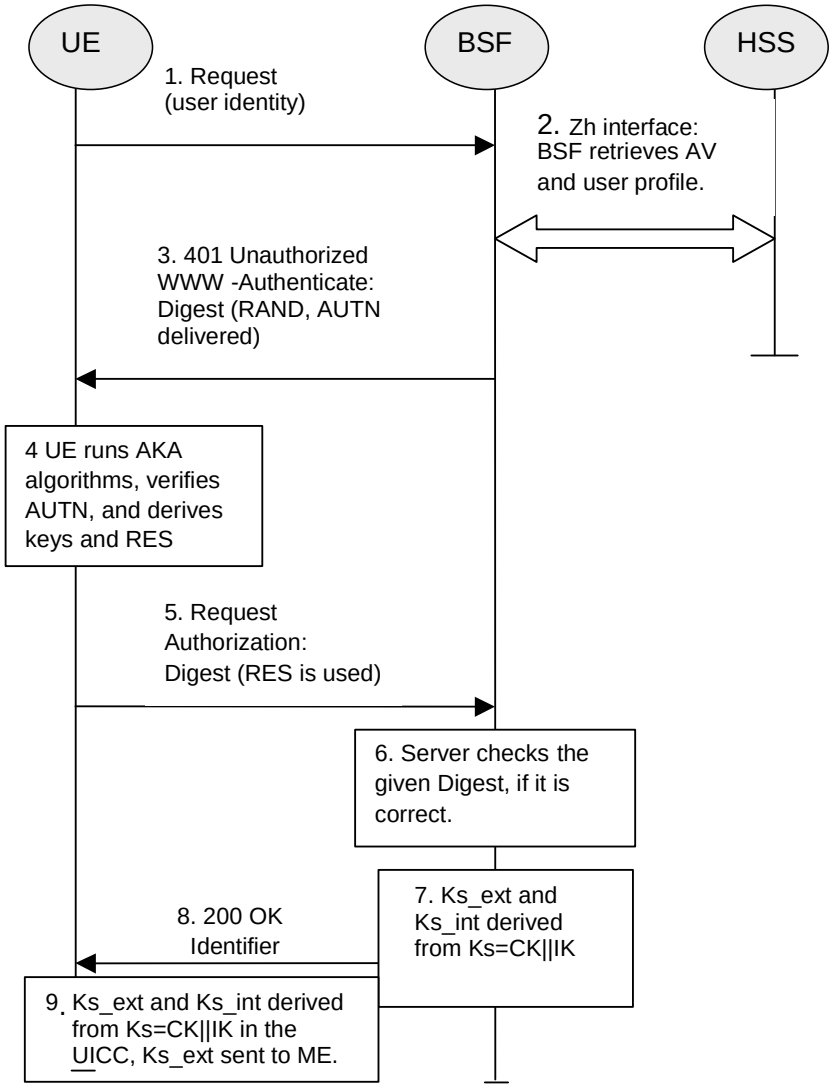

Figure 5.1: The bootstrapping procedure with UICC-based enhancements

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

2. The BSF retrieves the user profile and one or a whole batch of Authentication Vectors (AV, AV = RAND||AUTN||XRES||CK||IK) over the Zh interface from the HSS. The HSS recognises that the UICC is GBA_U aware and that the request for AVs came from a GBA_U aware BSF, and generates a challenge with a special RAND. The special RAND is described within Annex C of this specification. If the BSF received AVs with special-RANDs then it stores the XRES after flipping the least significant bit.

3. Then BSF forwards the RAND and AUTN to the UE in the 401 message (without the CK, IK and XRES). This is to demand the UE to authenticate itself.

4. The ME sends RAND and AUTN to the UICC. The UICC checks AUTN to verify that the challenge is from an authorised network; the UICC also calculates CK, IK and RES. This will result in session keys CK and IK in both BSF and UICC.

5. The UICC checks if the RAND was a special RAND as specified in step 2 of this clause. If this is not the case, the UICC transfers RES, CK and IK to the ME, and the ME proceeds according to the procedures

specified in section 4 of this document, without involving the UICC any further. If a special RAND was received, the UICC then applies a suitable key derivation function h1 to Ks, which is the concatenation of CK and IK, and possibly further h1-key derivation parameters to obtain two keys, Ks_ext and Ks_int, each of length 128 bit, i.e. h1(Ks, h1 key derivation parameters) = Ks_ext || Ks_int (cf. 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 on the UICC.

Editor's note: The definition of the key derivation function h1 and the possible inclusion of further h1-key derivation parameters (e.g. IMSI and RAND) are left to ETSI SAGE and to be included in the Annex B of the specification.

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 key material Ks by concatenating CK and IK. The BSF checks if the RAND was a special RAND as specified in step 2 of this clause. If this is not the case, the BSF applies the procedures specified in section 4 of this document. If the special RAND was recognized then the BSF applies the key derivation function h1 to Ks and possibly further h1-key derivation parameters to obtain two keys, Ks_ext and Ks_int, in the same way as the UICC did in step 5. The Transaction Identifier value shall be also generated in format of NAI by taking the RAND value from step 3, and the BSF server name, i.e. RAND@BSF_servers_domain_name.

9. The BSF shall send a 200 OK message, including the Transaction Identifier, to the UE to indicate the success of the authentication. The BSF also supplies a flag DER_FLAG to the UE, which indicates whether NAF-specific key derivation shall be applied to Ks_ext and Ks_int or not. If NAF-specific key derivation is performed it is to be applied uniformly to all keys shared between any UE and any NAF. In addition, in the 200 OK message, the BSF shall supply the lifetime of the keys Ks_ext and Ks_int, and an indication whether multiple key derivation shall be used. 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 interface. Ks_ext_NAF and Ks_int_NAF are used for securing the Ua interface. The ME 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 ME 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 IMSI, the NAF_Id and RAND. The NAF_Id consists of the full DNS name of the NAF.

Editor's note: *The definition of the h2 and the possible inclusion of further key derivation parameters are left to ETSI SAGE and to be included in the Annex B of the present specification.*

If multiple key derivation is used then the ME, the UICC and the BSF store the keys Ks_ext and Ks_int together with the associated Transaction Identifier 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. Otherwise, the keys Ks_ext and Ks_int and the Transaction Identifier may be deleted in the UE and in the BSF after the Ks_ext_NAF and Ks_int_NAF have been derived.

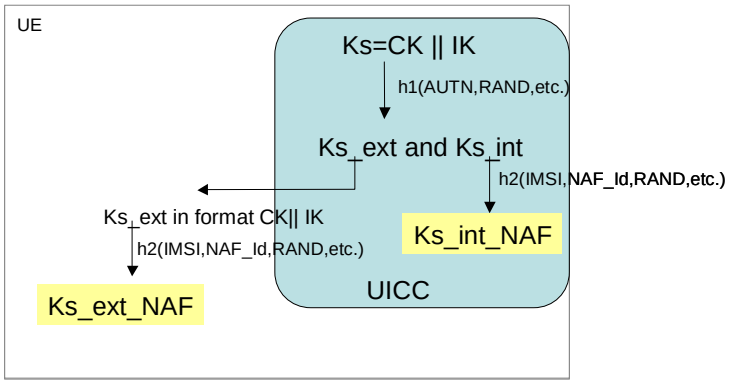

Figure 5.2: Key derivation for GBA-aware UICC when GBA-run was triggered

5.3.3 Procedures using bootstrapped Security Association

After UE and the BSF have been mutually authenticated, each time the UE wants to interact with an NAF the following steps are executed as depicted in Figure 5.3.

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 section 5.3.1. If the UE requests the NAF to use the GBA the UE includes a transaction identifier as specified in section 5.3.2, step 9, with a BSF server name as realm. If the NAF agrees, NAF and UE continue as specified in this clause. If the NAF does not agree it responds with a message which is specific to the application and is not specified in this specification.

Next, the UE and the NAF have to agree, which type of keys to use, Ks_ext_NAF or Ks_int_NAF, or both. The default is the use of Ks_ext_NAF only. This use is also supported by MEs and NAFs, which are GBA_U unaware. If Ks_int_NAF, or both, are to be used, this use has to be agreed between UE and NAF prior to the execution of the procedure described in the remainder of this clause 5.3.3. How this agreement is reached is application-specific and is not within the scope of this document.

Note: *such an agreement could e.g. be reached by manual configuration, or by an application-specific protocol step.*

In general, UE and NAF will not yet share the key(s) required to protect the Ua interface. If they do not, the UE proceeds as follows:

- if Ks_ext_NAF is required and a key Ks_ext is available in the ME, the ME derives the key Ks_ext_NAF from Ks_ext, as specified in clause 5.3.2;
- if Ks_int_NAF is required and a key Ks_int is available in the UICC, the ME requests the UICC to derive the key Ks_int_NAF from Ks_int, as specified in clause 5.3.2;
- if Ks_ext and Ks_int are not available in the UE, the UE first agrees on new keys Ks_ext and Ks_int with the BSF over the Ub interface, 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 key update request to the UE and terminate the protocol used over Ua interface. The form of this indication may depend on the particular protocol used over Ua interface;

UE and NAF can now start the communication over Ua interface using the keys Ks_ext_NAF or Ks_int_NAF, or both, as required. They proceed as follows:

- the UE supplies the Transaction Identifier to the NAF, as specified in clause 5.3.2, to allow the NAF to retrieve the corresponding keys from the BSF;

NOTE: *The UE may adapt the keys Ks_ext_NAF or Ks_int_NAF to the specific needs of the Ua interface. This adaptation is outside the scope of this specification.*

- when the UE is powered down, or when the UICC is removed, any keys Ks_ext and Ks_ext_NAF 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: *after each run of the protocol over the Ub interface, new keys Ks_ext and Ks_int, associated with a new transaction identifier, are derived in the UE according to clause 5.3.2, so that it can never happen, that keys Ks_ext and Ks_int with different transaction identifiers simultaneously exist in the UE.*

- when new keys Ks_ext and Ks_int are agreed over the Ub interface 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: *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 interface with BSF.

- the NAF requests from the BSF the keys corresponding to the Transaction Identifier, which was supplied by the UE to the NAF over the Ua interface. If the NAF is GBA_U aware it indicates this by including a corresponding flag in the request;

- The BSF derives the keys Ks_ext_NAF, and Ks_int_NAF (if additionally required), as specified in clause 5.3.2. If the NAF indicated in its request that it is GBA_U aware, the BSF supplies to NAF both keys, Ks_ext_NAF, and Ks_int_NAF, otherwise the BSF supplies only Ks_ext_NAF. In addition, the BSF supplies the lifetime time of these keys. If the key identified by the Transaction Identifier 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 key update request to the UE.

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

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

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

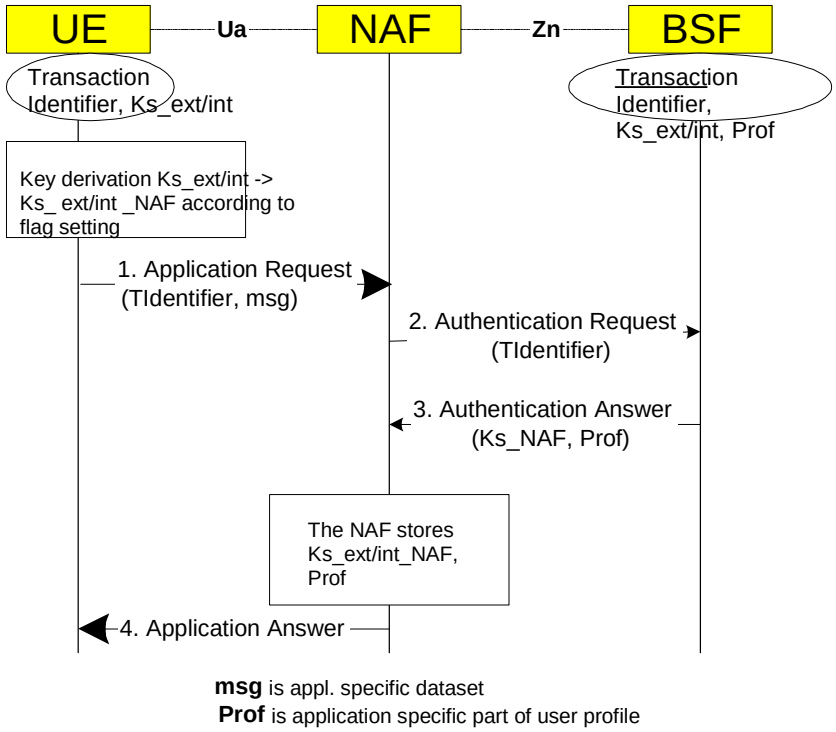

Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

5.3.4 Procedure related to service discovery

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

END OF CHANGE

Annex D (informative): Issues regarding migration from GBA_ME to GBA_U

This annex contains a few rules which should be heeded when upgrading from GBA_ME to GBA_U in order to avoid incompatibilities.

The HSS (AuC) shall be upgraded first before NAFs are introduced in the network that uses the GBA_U services and the GBA-aware UICC has to be administrated within the HSS so that the HSS (AuC) can generate the special RAND challenge.

The HSS (AuC) does NOT need to be upgraded in case GBA-aware UICC's are introduced within the network, but no NAFs make use of it.

The upgrade of the BSF to support GBA_U shall occur no later than that of the HSS.

BEGIN OF CHANGE

Annex EC (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2004-03	SP-23	SP-040175	-	-	Presented for approval at TSG SA #23	1.2.1	2.0.0
2004-03	SP-23	-	-	-	Approved and placed under Change Control (Rel-6)	2.0.0	6.0.0

END OF CHANGE