Technical Specification Group Services and System Aspects
Meeting #27, 14 - 17 March 2005,
Tokyo, Japan

TSGS#27(05)0139

Source: **SA WG3**

Title: **Three CRs to TS 33.220 (Rel-6)**

Document for: **Approval**

Agenda Item: **7.3.3**

The following CRs were 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-050139	33.220	045	1	Rel-6	Key derivation function: character encoding	C	6.3.0	S3-050168	SEC1-SC
SP-050139	33.220	047	1	Rel-6	Bootstrapping timestamp	C	6.3.0	S3-050143	SEC1-SC
SP-050139	33.220	048	-	Rel-6	Storage of B-TID in GBA_U NAF Derivation procedure	F	6.3.0	S3-050086	SEC1-SC

3GPP TSG-SA WG3 Meeting #37 **Tdoc ⌘S3-050168**
Sophia Antipolis, France, February 21-25, 2005

CR-Form-v7.1

CHANGE REQUEST

⌘	**33.220**	**CR**	**045**	⌘**rev**	**1**	⌘	Current version:	**6.3.0** ⌘

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:	⌘	Key derivation function: character encoding			
Source:	⌘	SA WG3			
Work item code:	⌘	SEC1-SC	***Date:***	⌘	23/02/2005
Category:	⌘	**C** *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.	***Release:***	⌘	Rel-6 *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:	⌘	How to encode input parameters for the key derivation function is unclear, e.g., how an input parameter which is a character string is encoded to an octet string. UTF-8 encoding shall be used in the encoding. To avoid confusion, the KDF input parameters are now separated by commas "," instead of concatenation marks "||".
Summary of change:	⌘	Input parameter encoding is clarified, i.e., UTF-8 encoding shall be used. The KDF input parameters are separated by commas "," instead of concatenation marks "||".
Consequences if not approved:	⌘	Input parameter encoding is unclear.

Clauses affected:	⌘	2, 4.5.2, 5.3.2, B.2.1 (new), B.3				
		Y	N			
Other specs affected:	⌘		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: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G security; Access security for IP-based services".

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

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

[19] 3GPP TS 33.310: "3rd Generation Partnership Project; Technical Specification Group Service and System Aspects; Network Domain Security (NDS); Authentication Framework (AF)".

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

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

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

[23] ISO/IEC 10118-3:2004: "Information Technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions".

[24] IETF RFC 3629 (2003): "UTF-8, a transformation format of ISO 10646".

===== 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.

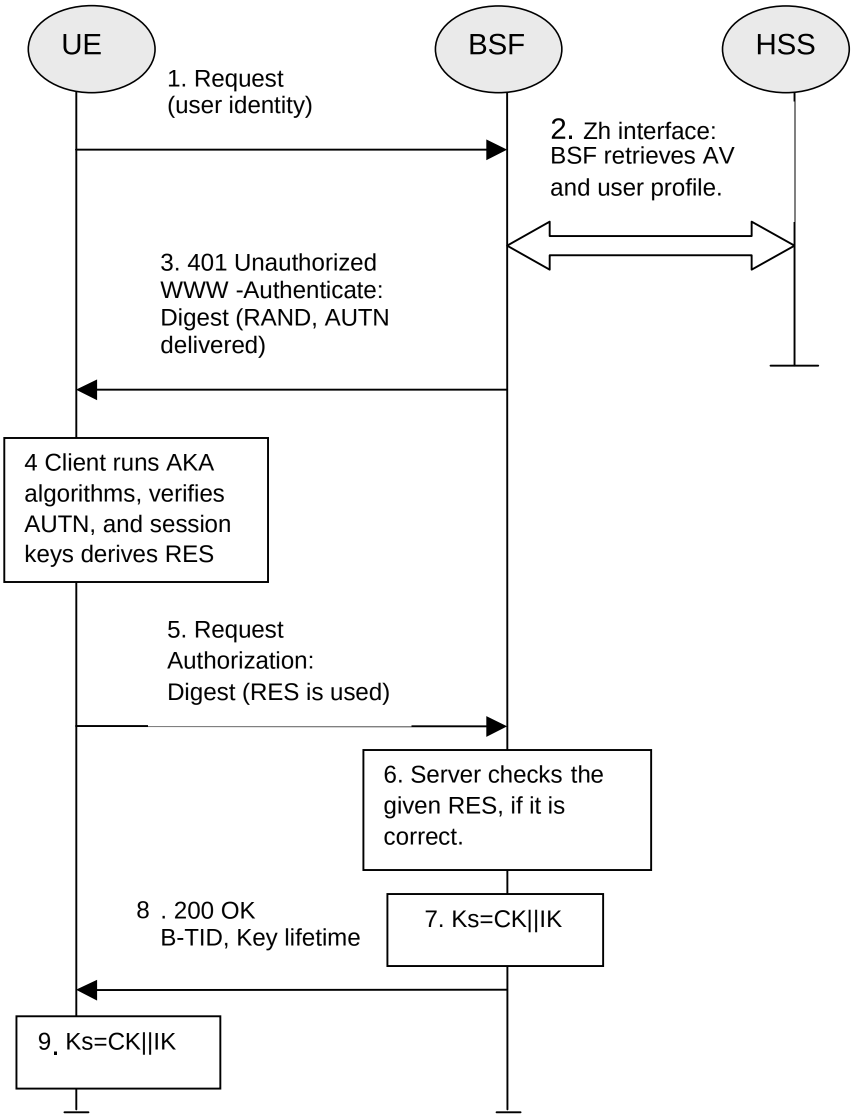


Figure 4.3: The bootstrapping procedure

1. The UE sends an HTTP request towards the BSF.
2. BSF retrieves the complete set of GBA user security settings and one Authentication Vector (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), where KDF is 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.

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.

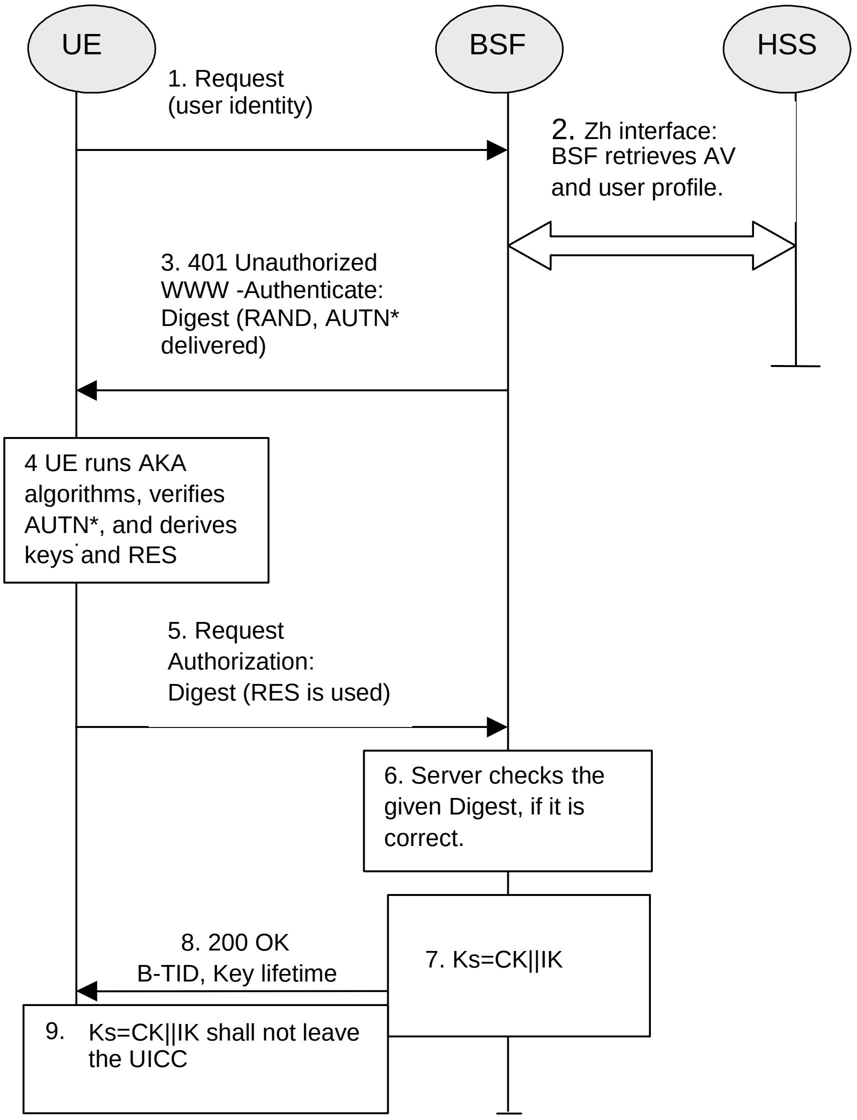

UE
BSF
HSS
1. Request (user identity)
2. Zh interface: BSF retrieves AV and user profile.
3. 401 Unauthorized WWW -Authenticate: Digest (RAND, AUTN* delivered)
4 UE runs AKA algorithms, verifies AUTN*, and derives keys and RES
5. Request Authorization: Digest (RES is used)
6. Server checks the given Digest, if it is correct.
7. Ks=CK||IK
8. 200 OK B-TID, Key lifetime
9. Ks=CK||IK shall not leave 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 Authentication Vector (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))

NOTE: Trunc denotes that from the 160 bit output of SHA-1 [21], the 64 bits numbered as [0] to [63] are used within the * operation to MAC.

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* ⊕ Trunc(SHA-1(IK))). 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 ME 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 the key 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 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 Ks.

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. Ks_ext_NAF and Ks_int_NAF are used for securing the Ua reference point.

   Ks_ext_NAF is computed in the UICC as Ks_ext_NAF = KDF(Ks, ~~h1 key derivation parameters~~"gba-me", RAND, IMPI, NAF_Id), and Ks_int_NAF is computed in the UICC as Ks_int_NAF = KDF(Ks, ~~h1 key derivation parameters~~"gba-u", RAND, IMPI, NAF_Id), where KDF is the key derivation function as specified in Annex B, and the 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.

NOTE: The NOTE 2 of clause 4.5.2 also applies here.

   The UICC and the BSF 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

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 k~~l~~ in the range [0,65535].~~0 □ l □ 65535~~.

   b) Li is then a two-octet representation of the number k~~l~~, with the most significant bit of the first octet of Li equal to the most significant bit of k~~l~~, and the least significant bit of the second octet of Li equal to the least significant bit of k~~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 [22] and [23]) computed on the string S using the key Key:

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

B.2.1 Input parameter encoding

A character string shall be encoded to an octet string according to UTF-8 encoding rules as specified in IETF RFC 3629 [24].

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 encoded to an octet string using UTF-8 encoding (see clause B.2.1),
- L2 = length of IMPI is variable (not greater that 65535),
- P3 = NAF_ID encoded to an octet string using UTF-8 encoding (see clause B.2.1), 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 Meeting #37
Sophia Antipolis, France, February 21-25, 2005

Tdoc ⌘S3-050143

CR-Form-v7.1

CHANGE REQUEST

⌘	**33.220**	**CR**	**047**	⌘**rev**	**1** ⌘	Current version: **6.3.0** ⌘

*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:	⌘	Bootstrapping timestamp		
Source:	⌘	SA WG3		
Work item code:	⌘	SEC1-SC	***Date:*** ⌘	23/02/2005
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:	⌘	Currently, the NAF can only implictly discover the actual bootstrapping time. The NAF may need the actual bootstrapping time to discover the freshness of the original GBA session key Ks. Upon discovering the bootstrapping time the NAF can determine whether the original bootstrapping procedure is too old according to its policies and whether it requires the UE re-run the bootstrapping procedure. Also, the operator may have subscriber specific bootstrapping lifetimes (e.g., for prepaid subscribers). This can be set in subscriber's GBA User Security Settings (cf. clause 4.2.3 of TS 33.220). In this case the implicit discovery of the actual bootstrapping time is not possible as the bootstrapping lifetime time may vary per subscriber.
Summary of change:	⌘	The bootstrapping time is sent from the BSF to the NAF (in addition to the bootstrapping lifetime time).
Consequences if not approved:	⌘	The NAF cannot reliably determine the actual bootstrapping time.

Clauses affected:	⌘	4.4.6, 4.5.3, 5.3.2				
		Y	**N**			
Other specs affected:	⌘	X		Other core specifications	⌘	TS 29.109
			X	Test specifications		
			X	O&M Specifications		
Other comments:	⌘					

===== BEGIN 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];

NOTE 1: Annex E 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 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 USSs for;

NOTE 2: If some application needs only a subset of an application-specific USS, e.g. only one IMPU, the NAF selects this subset from the complete set of USS 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 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 bootstrapping time and 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 3: This does not preclude a NAF to refresh the key before the expiry time according to the NAF's local policy.

NOTE 4: 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).

===== 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.

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 key's lifetime has expired or will expire soon, or the key can not meet the NAF local validity condition, it shall send a suitable bootstrapping renegotiation request to the UE, see figure 4.5. If the key's 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.

   - 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;

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

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

- With the key material request, the NAF shall supply NAF's public hostname that UE has used to access NAF to BSF, and BSF shall be able 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 bootstrapping time and 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 6: 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.

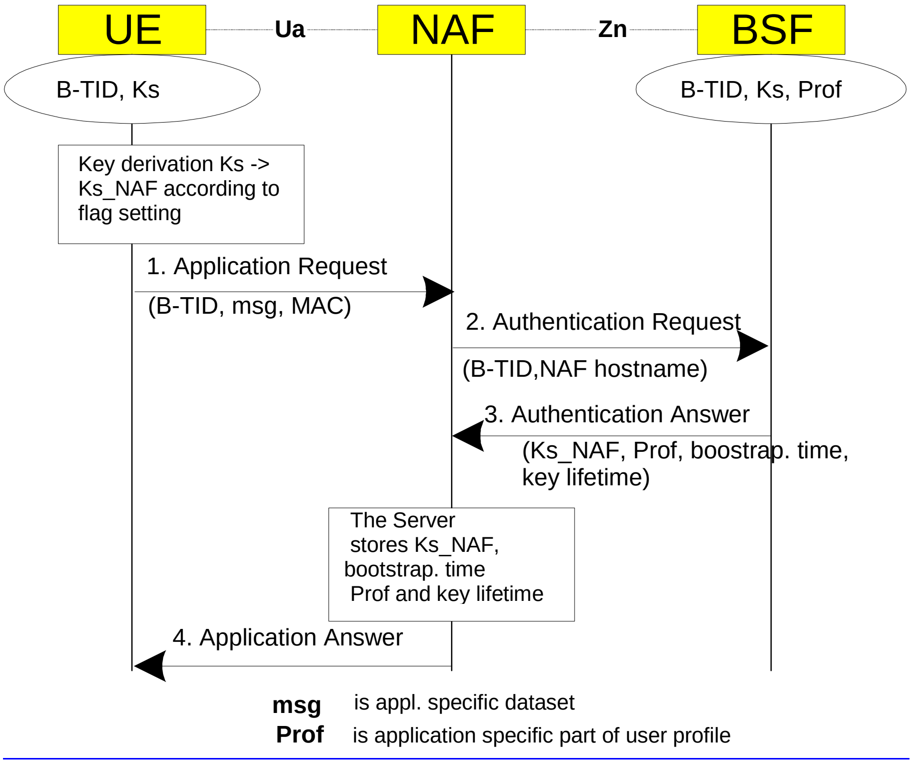
NOTE 7: The NAF shall adapt the key material Ks_NAF to the specific needs of the reference point Ua in the same way as the UE did. This adaptation is outside the scope of this specification.

- 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 (see clause 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 USSs 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.


UE
Ua
NAF
Zn
BSF
B-TID, Ks
B-TID, Ks, Prof
Key derivation Ks -> Ks_NAF according to flag setting
1. Application Request
(B-TID, msg, MAC)
2. Authentication Request
(B-TID,NAF hostname)
3. Authentication Answer
(Ks_NAF, Prof, boostrap. time, key lifetime)
The Server stores Ks_NAF, bootstrap. time Prof and key lifetime
4. Application Answer
msg is appl. specific dataset
Prof is application specific part of user profile

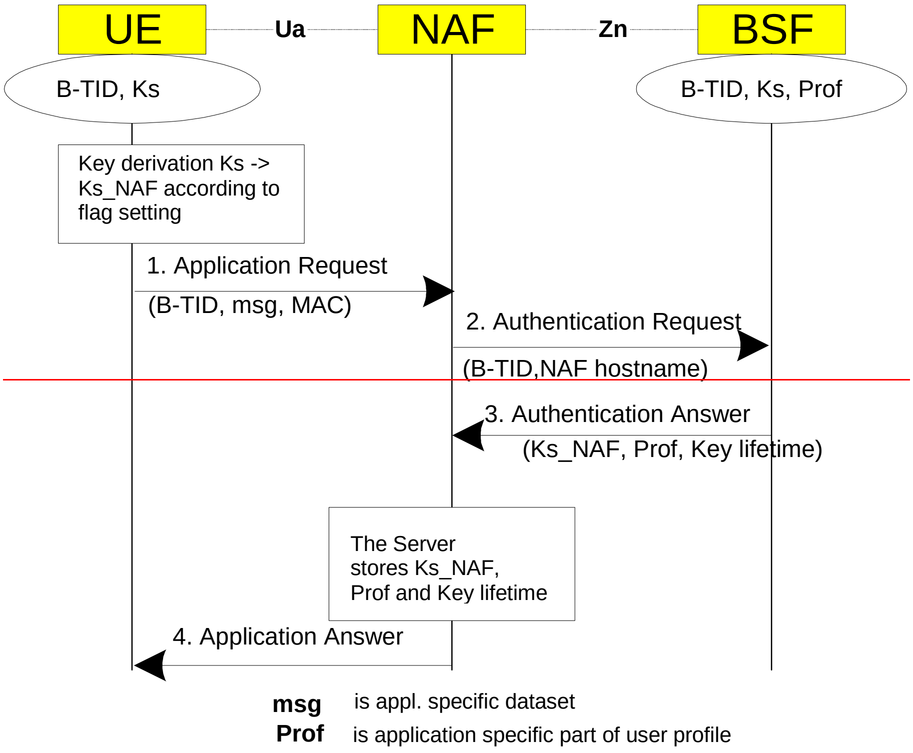

UE
Ua
NAF
Zn
BSF
B-TID, Ks
B-TID, Ks, Prof
Key derivation Ks -> Ks_NAF according to flag setting
1. Application Request
(B-TID, msg, MAC)
2. Authentication Request
(B-TID,NAF hostname)
3. Authentication Answer
(Ks_NAF, Prof, Key lifetime)
The Server stores Ks_NAF, Prof and Key lifetime
4. Application Answer
msg is appl. specific dataset
Prof is application specific part of user profile

Figure 4.4: The bootstrapping usage procedure

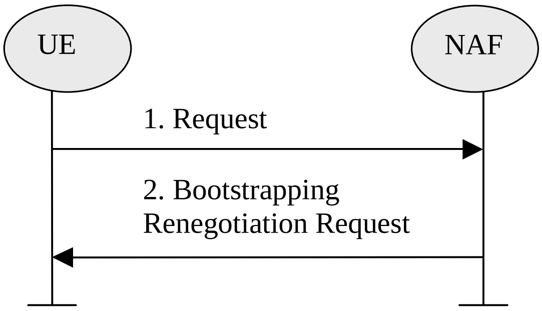


Figure 4.5: Bootstrapping renegotiation request

===== 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_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.

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 for the selected UICC application is available in the UICC, the ME requests the UICC to derive the key Ks_ext_NAF from Ks, as specified in clause 5.3.2;

- if Ks_int_NAF is required and a key Ks for the selected UICC application is available in the UICC, the ME requests the UICC to derive the key Ks_int_NAF from Ks, as specified in clause 5.3.2;

NOTE 2: If it is not desired by the UE to use the same Ks for the selected UICC application to derive more than one Ks_ext/int_NAF then the UE should first agree on new key Ks 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 for the selected UICC application is not available in the UE, the UE first agrees on a new key Ks 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. If the key's 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.

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.

- 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, a new key Ks, associated with a new B-TID, are derived in the UE according to clause 5.3.2, so that it can never happen, that key Ks with different B-TIDs simultaneously exist in the UE.

- When new key Ks 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: According to the procedures defined in clauses 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: 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 one or more application-specific 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 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.

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 bootstrapping time and 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 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 (See figure 4.5) to the UE;

NOTE 10: 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 11: 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 (see clause 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 USSs to NAF according to the BSF's 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.

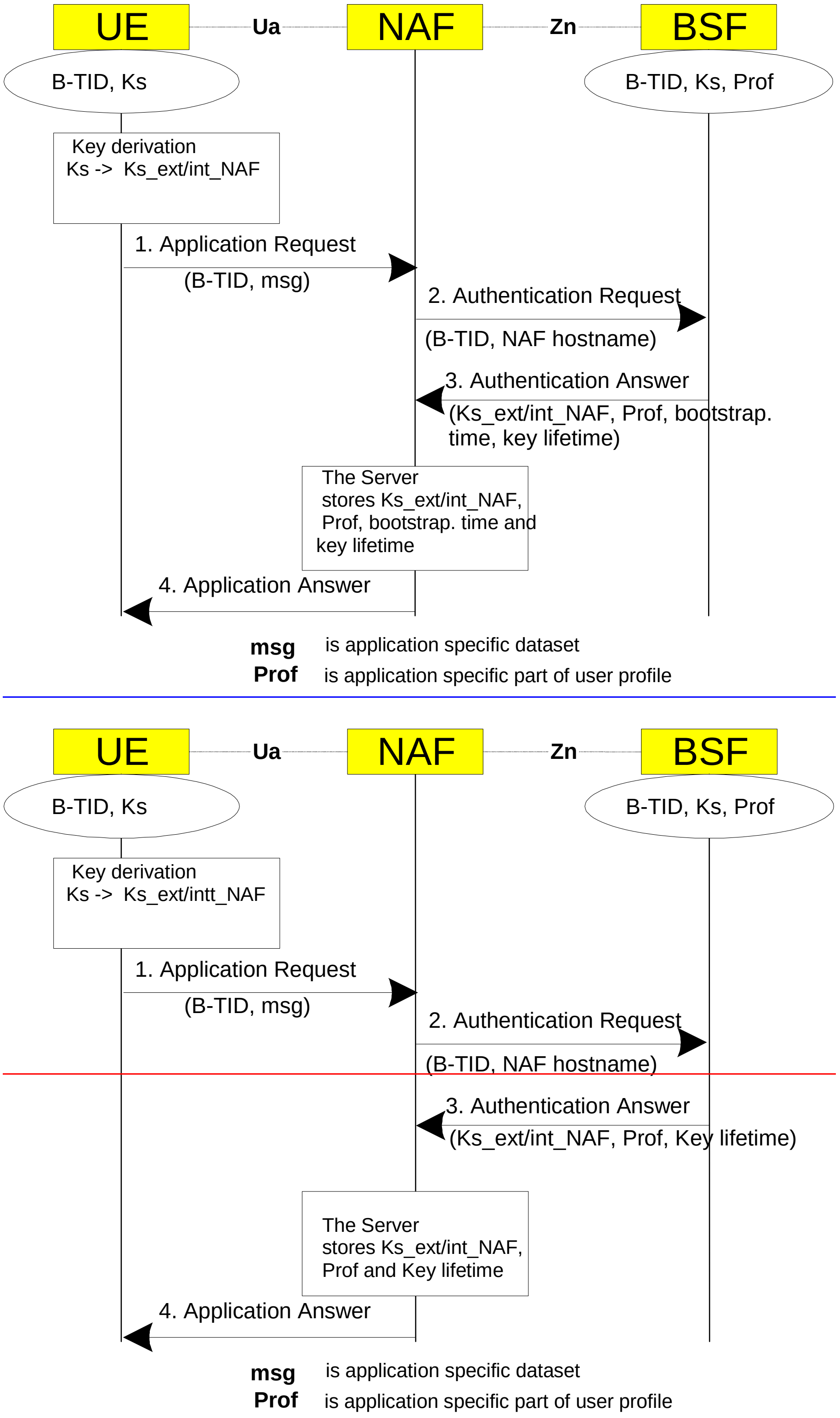

UE
Ua
NAF
Zn
BSF
B-TID, Ks
B-TID, Ks, Prof
Key derivation
Ks -> Ks_ext/int_NAF
1. Application Request
(B-TID, msg)
2. Authentication Request
(B-TID, NAF hostname)
3. Authentication Answer
(Ks_ext/int_NAF, Prof, bootstrap. time, key lifetime)
The Server stores Ks_ext/int_NAF, Prof, bootstrap. time and key lifetime
4. Application Answer
msg is application specific dataset
Prof is application specific part of user profile
UE
Ua
NAF
Zn
BSF
B-TID, Ks
B-TID, Ks, Prof
Key derivation
Ks -> Ks_ext/intt_NAF
1. Application Request
(B-TID, msg)
2. Authentication Request
(B-TID, NAF hostname)
3. Authentication Answer
(Ks_ext/int_NAF, Prof, Key lifetime)
The Server stores Ks_ext/int_NAF, Prof and Key lifetime
4. Application Answer
msg is application specific dataset
Prof is application specific part of user profile

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

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

3GPP TSG-SA WG3 Meeting #37
Sophia Antipolis, France, 21-25 February 2005

Tdoc ⌘S3-050086

CR-Form-v7.1

CHANGE REQUEST

⌘	**33.220**	**CR**	**048**	⌘**rev**	**-**	⌘	Current version:	**6.3.0** ⌘

*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:	⌘ Storage of B-TID in GBA_U NAF Derivation procedure		
Source:	⌘ SA WG3		
Work item code:	⌘ SEC1-SC	***Date:***	⌘ 10/02/2005
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:	⌘ In some cases the NAF-ID is not enough to identify the Ks_int_NAF/Ks_ext_NAF unambigiously. For some example a new NAF key generation, from which the http session was not able to complete towards the corresponding NAF, results in different Ks_ext_NAF/Ks_int_NAF key pairs (on the UE and onother in the NAF) identified with the same NAF_ID. Therefore, text shall be added in TS 33.220 to indicate that the UICC shall store B-TID together with Ks_int_NAF and NAF_ID in order to identify unambiguously the Ks_int_NAF key.
Summary of change:	⌘ Storage of B-TID together with Ks_int_NAF and NAF_ID in GBA_U NAF Derivation procedure.
Consequences if not approved:	⌘ Incomplete description of GBA_U NAF Derivation procedure.

Clauses affected:	⌘ Annex G.2				
		Y	**N**		
Other specs affected:	⌘		**X**	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

G.2 GBA_U NAF Derivation procedure

This procedure is part of the Procedures using bootstrapped Security Association as described in clause 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 clause 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 and associated B-TID 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.

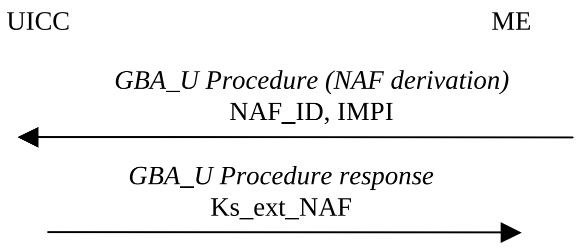

UICC
ME
GBA_U Procedure (NAF derivation)
NAF_ID, IMPI
GBA_U Procedure response
Ks_ext_NAF

Figure G.2: GBA_U NAF derivation procedure