3GPP TSG-SA WG3 Meeting #33
10th – 14th May 2004, Beijing, China

Tdoc ⌘S3-040434

CR-Form-v7

CHANGE REQUEST

⌘ **33.220** **CR** **CRNum** ⌘ **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⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☒

Title:	⌘	Multiple key derivation mandatory
Source:	⌘	3GPP SA3
Work item code:	⌘	SSC-GBA
Date:	⌘	14/05/2004
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:	⌘	To standardise multiple key derivation for keys used over Ua as mandatory, for security reasons and in order to reduce the number of options and flags defined.
Summary of change:	⌘	In the current version of the spec, both, NAF-specific key derivation and multiple key derivation are optional. This CR proposes to make both mandatory for implementation. A note is added to explain how the use of multiple key derivation can be avoided by an appropriated policy in the NAF. Additionally clarifications were made for the case, when the NAF is accessible under different FQDNs.
Consequences if not approved:	⌘	If key derivation is left optional, this may result in insecure solutions. If multiple key derivation is left optional, this may result in solutions, which are inefficient in operation. Furthermore, the complexity of implementations is increased when multiple key derivation is left optional, as different versions of key usage (use of plain Ks, use of single derived Ks_NAF, and use of multiple derived Ks_NAF) must be signalled and supported.

Clauses affected:	⌘	2, 3.2, 4.2.1, 4.5.2, 4.5.3			
Other specs affected:	⌘	**Y**	**N**		
		X		Other core specifications	⌘ TS 24.109
			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".

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

********** begin change **********

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
FQDN	Fully Qualified Domain Name
GAA	Generic Authentication Architecture
GBA	Generic Bootstrapping Architecture
HSS	Home Subscriber System
IK	Integrity Key
KDF	Key Derivation Function
MNO	Mobile Network Operator
NAF	Network Application Function
PKI	Public Key Infrastructure

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

********** begin 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 ~~can~~ shall restrict the applicability of the key material to a ~~defined set of~~specific NAF~~s~~ 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 section 4.5.2.

~~Editor's note: Key generation for NAF is ffs. Potential solutions may include:~~
~~— Separate run of HTTP Digest AKA over Ub interface for each request of key material from a NAF~~
~~— Issues with key lifetime are ffs.~~

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

********** 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 3). 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 4.5.3).

NOTE1: 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.

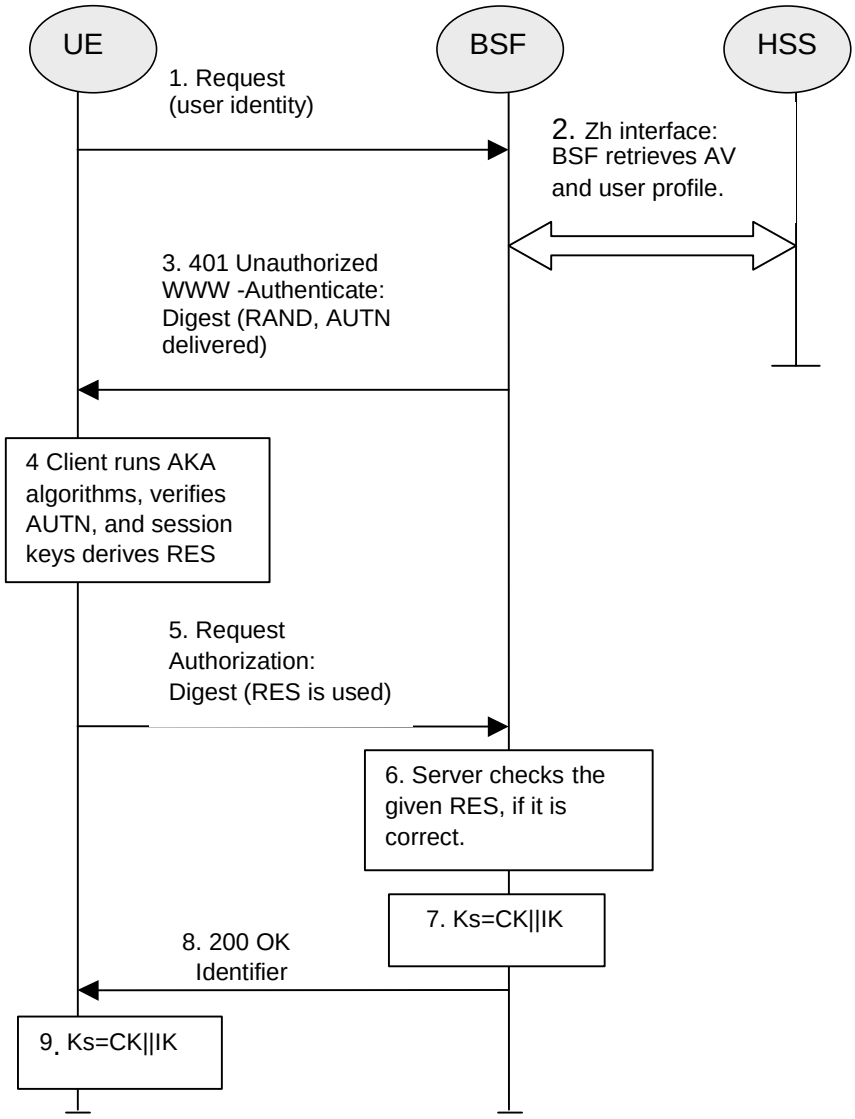

Figure 3: The bootstrapping procedure

1. The UE sends an HTTP request towards the BSF.
2. BSF retrieves the user profile and one or a whole batch of Authentication Vectors (AV, AV = RAND||AUTN||XRES||CK||IK) over the reference point Zh ~~interface~~ from the HSS.
3. Then BSF forwards the RAND and AUTN to the UE in the 401 message (without the CK, IK and XRES). This is to demand the UE to authenticate itself.
4. The UE checks AUTN to verify that the challenge is from an authorised network; the UE also calculates CK, IK and RES. This will result in session keys IK and CK in both BSF and UE.
5. The UE sends another HTTP request, containing the Digest AKA response (calculated using RES), to the BSF.
6. The BSF authenticates the UE by verifying the Digest AKA response.
7. The BSF generates key material Ks by concatenating CK and IK. The Transaction Identifier 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.
8. The BSF shall send a 200 OK message, including a 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 key derivation shall be applied to Ks or not. If 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 key Ks~~, and an indication whether multiple key derivation shall be used~~. 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~~, if applicable~~. Ks_NAF ~~is~~ shall be used for securing the reference point Ua~~ interface~~.

   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 ~~include~~ consist of the user's IMP~~S~~I, the NAF_Id and RAND. The NAF_Id consists of the full DNS name of the NAF. KDF shall be implemented in the ME.

   NOTE2: 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 "TLS V1.0 Extensions" [9] or other protocol means with similar purpose.

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

~~If multiple key derivation is used then t~~The UE and the BSF shall store the key Ks with the associated Transaction Identifier for further use, until the lifetime of Ks has expired, or until the key Ks is updated~~. Otherwise, the key Ks and the Transaction Identifier may be deleted in the UE and in the BSF after the key Ks_NAF has been derived~~.

4.5.3 Procedures using bootstrapped Security Association

After UE is authenticated with the BSF, every time the UE wants to interact with an NAF the following steps are executed as depicted in figure 5.

UE starts communication over reference point Ua~~ interface~~ with the NAF:

- in general, UE and NAF will not yet share the key(s) required to protect reference point Ua~~ interface~~. If they already do (i.e. if a key Ks_NAF for the corresponding key derivation parameter NAF_Id_n is already available),, the UE and the NAF can start to securely communicate right away. If the UE and the NAF do not yet share a key, the UE proceeds as follows:
  - if a key Ks is available in the UE, the UE derives the key Ks_NAF from Ks, as specified in clause 4.5.2;
- ~~—~~if no key Ks is available in the UE, the UE first agrees on a new key Ks with the BSF over the reference point Ub~~ interface~~, and then proceeds to derive Ks_NAF;

NOTE: if it is not desired by the UE to use the same Ks to derive more than one Ks_NAF then the UE should agree on a new key Ks with the BSF over the reference point Ub, and then proceed to derive Ks_NAF:.

- ~~—~~if the NAF shares a key with the UE, but an update of that key is needed, e.g. because the key's lifetime has expired, it shall send a suitable key update request to the UE and terminates the protocol used over reference point Ua~~ interface~~. The form of this indication may depend on the particular protocol used over reference point Ua~~ interface~~ (cf. 4.5.1);

NOTE: if it is not desired by the NAF to use the same Ks to derive more than one Ks_NAF then the NAF should always reply to the first request sent by a UE by sending a key update request to the UE.

- the UE supplies Transaction Identifier to the NAF, in the form of a Transaction Identifier, to allow the NAF to retrieve specific key material from BSF;

NOTE: To allow for consistent key derivation in BSF and UE, both have to use the same FQDN for derivation (cf. NOTE2 in section 4.5.2). For each protocol used over Ua it shall be specified if only cases (1) and (2) of NOTE2 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.

- the UE derives the keys required to protect the protocol used over reference point Ua ~~interface~~ from the key material, as specified in clause 4.3.2;

NOTE: The UE shall adapt the key material Ks_NAF to the specific needs of the reference point 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 and Ks_NAF shall be deleted from storage;

- when a new Ks is agreed over the reference point Ub ~~interface~~ 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 ~~interface~~ with BSF

- The NAF requests key material corresponding to Transaction Identifier supplied by the UE to the NAF used over reference point Ua ~~interface~~. 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 (cf. note above on key derivation in this section);

- The BSF derives the keys required to protect the protocol used over reference point Ua ~~interface~~ from the key material Ks and the key derivation parameters, as specified in clause 4.5.2, and supplies to NAF the requested key material Ks_NAF, as well as the lifetime time of that key material. 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 shall adapt the key material Ks_NAF to the specific needs of the reference point Ua ~~interface~~ in the same way as the UE did. This adaptation is outside the scope of this specification.

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

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

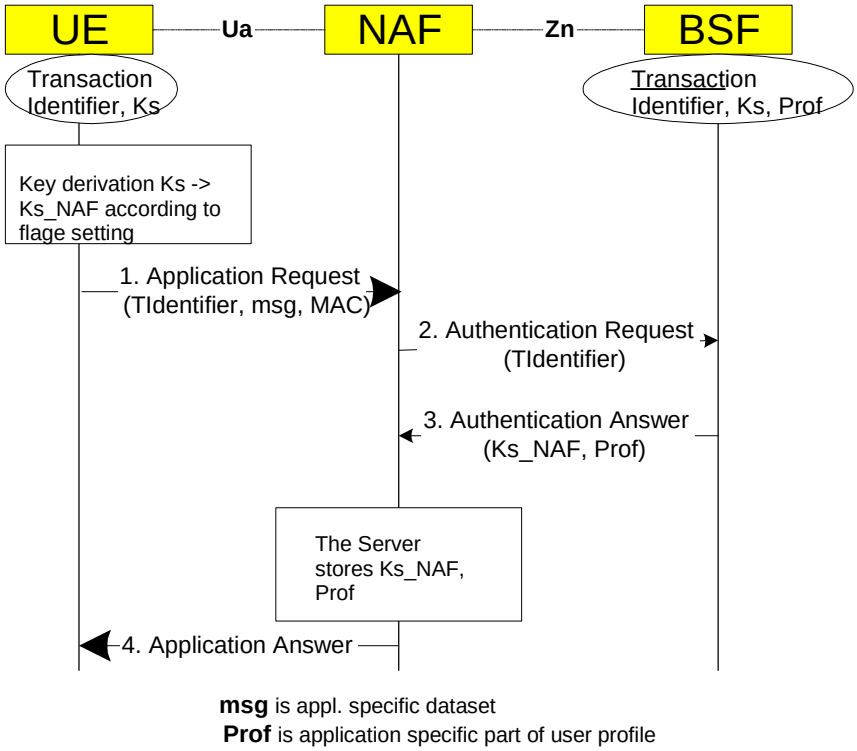

Figure 5: The bootstrapping usage procedure

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