3GPP TSG SA WG3 Security **S3-040343**

10th – 14th May, 2004

Beijing, China

Agenda Item:

Source: Ericsson

Title: Multiple key derivation in GBA

Document for: Discussion/Decision

1. Introduction

This document discusses multiple key derivation in GBA. Basically, there is potential interoperability problem when GBA multiple key derivation is implemented in UICC, or when multiple software vendors provide applications to same UE. It is proposed that multiple key derivation is made optional in UE side. BSF should be able to maintain multiple keys (Ks) for one UE.

2. Problem statement

SA3 has currently the following working assumptions related to key derivation in GBA:

- The basic format of keying material, Ks, is concatenated session keys (i.e. Ks = IK||CK).
- Multiple key derivation is optional. If multiple key derivation is used (indicated by a flag), Ks_NAF = KDF (Ks, key derivation parameters), where KDF is a suitable key derivation function, and the key derivation parameters include the user's IMSI, the NAF_Id and RAND.
- The same Ks may optionally be used with several NAFs. This procedure is called as “multiple key derivation”, and it results different keying material for different NAF.

If several applications share the same key material, the following practical interoperability problems should be addressed:

- If the key derivation is implemented in UICC (as it may be done in GBA_U), there must be a standard interface for applications to give relevant key derivation parameter, such as NAF_Id, to UICC, and for UICC to return potential keying material back to applications. UICC is not able to derivate the keys for different NAFs if it does not know related NAF identities.
- If different software vendors develop applications to same UE, there must be similar interface between applications and software entity that implements multiple key derivation.

3. Alternative solutions

The following alternative solutions have been identified:

1. In ME based GBA, the interface between applications and the entity implementing multiple key derivation is proprietary. The use of multiple key derivation is not supported in UICC based GBA.
2. Open interface should be developed in UE between applications and the entity implementing multiple key derivation. This interface could be used in both UICC based GBA implementations, and in the case when multiple software vendors provide applications to same UE.
3. BSF should be able to tolerate an exceptional situation where some applications use multiple key derivation while others do not. The indication from BSF to UE about the use of multiple key derivation should be seen as strong recommendation. However, BSF should still be able to store multiple keys (Ks) for one UE. BSF should

not use the generation of new Ks as an indication that existing Ks can be removed before its lifetime has expired.

4. Conclusions

Ericsson proposes that SA3 makes multiple key derivation optional to use in UE (solution 3 in section 3). This solution would give more time to develop the interface between applications and the entity implementing multiple key derivation, e.g. UICC.

3GPP TSG SA WG3 Security — S3#33 **S3-040xxx**

10th – 14th May, 2004, Beijing, China

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 [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	GBA multiple key generation
Source:	⌘	Ericsson
Work item code:	⌘	SEC1-SC
Date:	⌘	29 April 2004
Category:	⌘	
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:	⌘	There is potential interoperability problem when GBA multiple key derivation is implemented in UICC, or when multiple software vendors provide applications to same UE.
Summary of change:	⌘	Multiple key derivation is made optional to use in UE. BSF shall be able to store multiple keys (Ks) for one UE.
Consequences if not approved:	⌘	SA3 should develop open interface in UE between application software and the entity that implements multiple key derivation.

Clauses affected:	⌘	4.5.2

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

Other comments: ⌘

***** Begin of Change ****

4.5.2 Bootstrapping procedures

When a UE wants to interact with a NAF, and it knows that the bootstrapping procedure is needed, it shall first perform a bootstrapping authentication (see figure 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).

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 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.

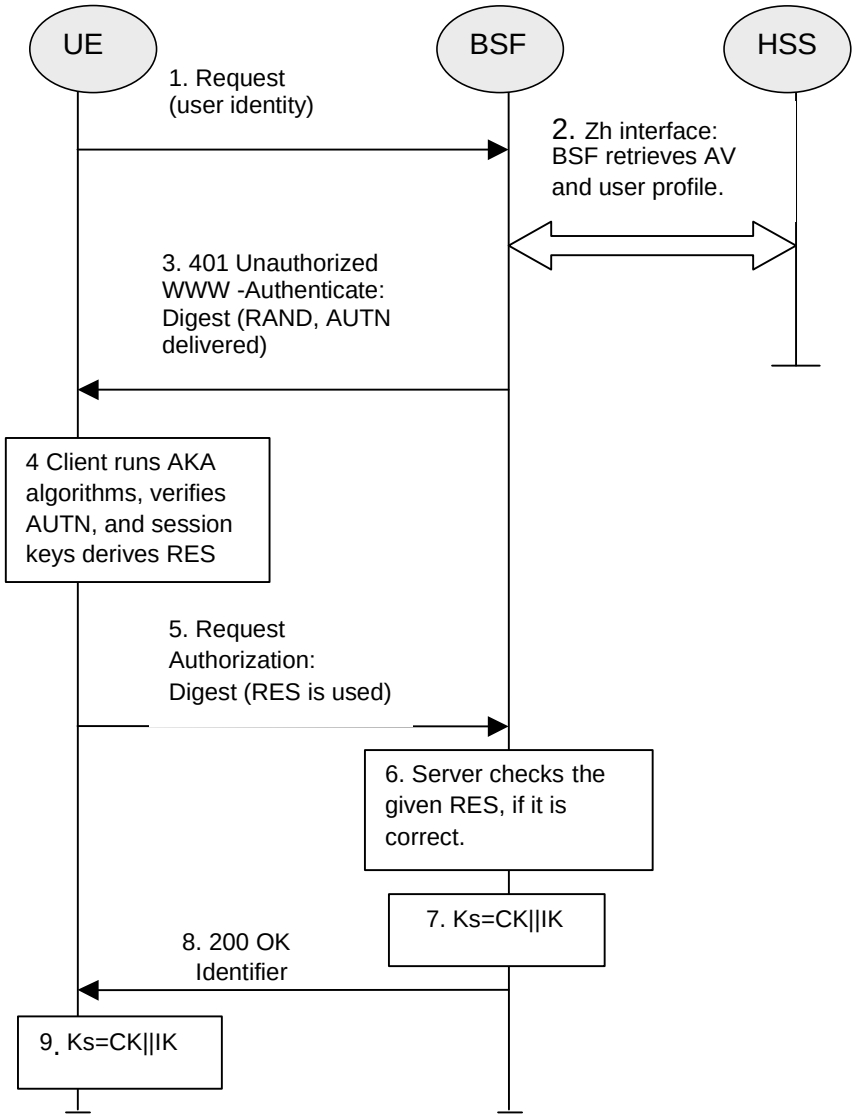

Figure 3: The bootstrapping procedure

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 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~~ should 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 used for securing the 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 the user's IMSI, the NAF_Id and RAND. The NAF_Id consists of the full DNS name of the NAF. KDF shall be implemented in the ME.

Editor's note: The definition of the KDF 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 UE and the BSF 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~~. It is recommended that all applications in UE use the same Ks for multiple key derivation, however, BSF shall also be able to store multiple Ks values for one UE for the case when all applications in UE are not able to reach the same Ks value. 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.