3GPP TSG-SA WG3 Meeting S3#33
Beijing, China, May 10-14, 2004

Tdoc ⌘S3-040334

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:	⌘	Removal of key derivation options
Source:	⌘	Nokia
Work item code:	⌘	SSC-GBA
Date:	⌘	02/05/2004
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:	⌘	It has been agreed the multiple key derivation shall always be done. This CR implements the required changed, i.e., removes the two key derivation flags from this specification.
Summary of change:	⌘	Key derivation flag are removed.
Consequences if not approved:	⌘	The unnecessary key derivation flags are left to the specification.

Clauses affected:	⌘	4.2.1, 4.5.2

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

⌘

Other comments: ⌘

===== 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. 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.~~

===== 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 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.

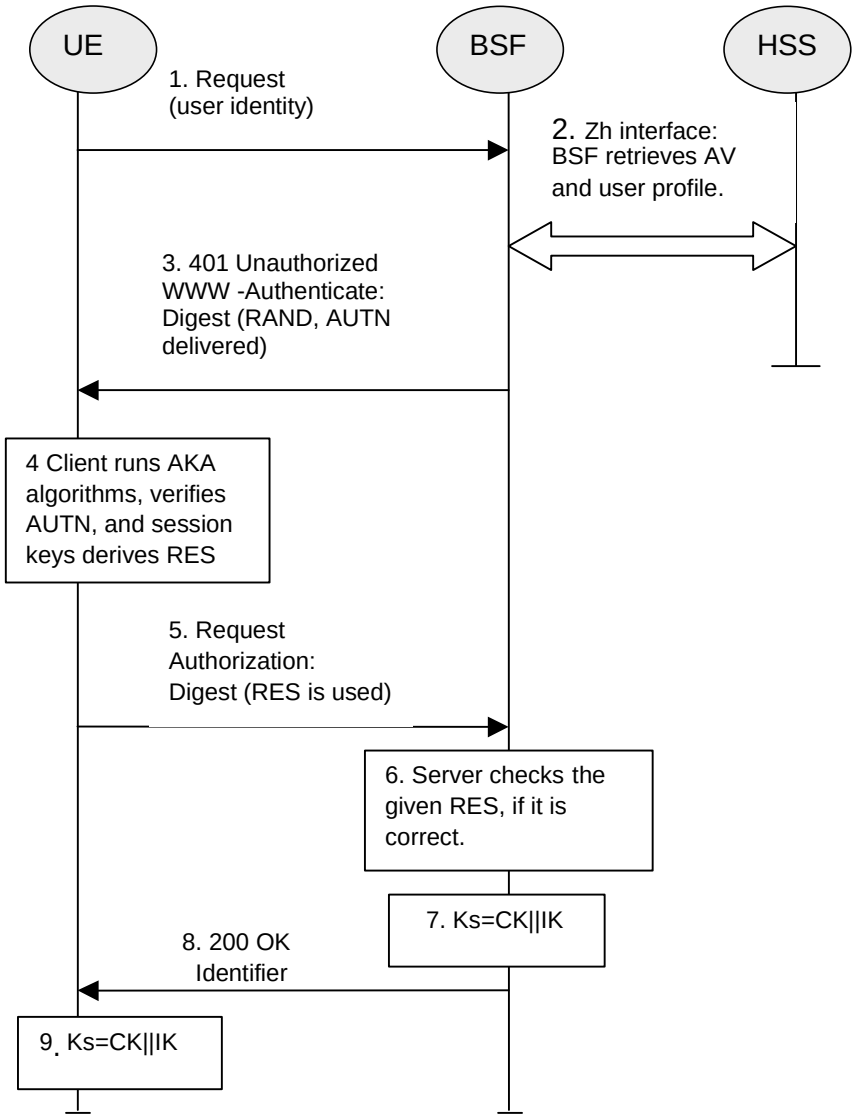

Figure 3: The bootstrapping procedure

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 k~~Key derivation shall always ~~is~~be performed and 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 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 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.~~

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

3GPP TSG SA WG3 Security — SA3#33 **S3-040xxx**
10-14 May 2004
Beijing, China

Source: **Nokia**

Title: **Removal of key derivation flags**

Document for: **Discussion and decision**

Agenda Item: **GBA**

1 Introduction

In the current GBA specification [TS 33.220], there are two flags indicating different behaviour related key derivation:

- a flag indicating whether key derivation is done or not, and
- a flag indicating whether multiple keys (Ks_NAFs) can derived from the key material (Ks).

This contribution discusses the necessity of the two key derivation flags, and concludes that the flags are not needed. Chapter 2 discusses whether key derivation should be done or not, and chapter 3 discusses the need for *dedicated bootstrapping session* (i.e., whether *multiple* key derivations should always be done, and should there be a possibility to limit the usage of one bootstrapping results to single NAF).

2 Key derivation

2.1 Bootstrapping states identified by key derivation flags

When the two flags are used it allows four different options in key derivation:

Description	Key derivation flag	Multiple key derivations flag
Option 1: Key derivation is not done, and naturally also multiple key derivations are not allowed. Ks itself is used as the secret key in Ua interface, and with only one NAF.	off	off
Option 2: Key derivation is done, but multiple key derivations are not allowed. Ks is used to derive only one Ks_NAF.	on	off
Option 3: Key derivation is not done, but multiple key use is allowed. Ks itself would be used as the secret key in Ua interface with multiple NAFs. *This option should not be used.*	off	on
Option 4: Key derivation is done and multiple key derivations are allowed. Ks would be used derive multiple Ks_NAF keys which would be used the corresponding NAFs as the secret key in Ua interface.	on	on

Table 1. Key derivation table.

Option 3 should not be used, since the same key (Ks) is used with multiple NAFs. If same key is used with multiple NAFs then a NAF may identify as UE to another NAF. Option 4 is a valid state, where key derivation is used with

multiple NAFs. Options 1 and 2 are similar since Ks is used only once. The difference is that in option 2, Ks_NAF is derived from Ks to be used with NAF where option 1 uses directly Ks as the key.

Hence, option 4 is a valid use case and option 3 should be dropped. Options 1 and 2 are discussed below.

2.2 Key derivation with dedicated bootstrapping

This chapter discusses the need for key derivation when the bootstrapping session is used with only one NAF (options 1 and 2 in Table 1). The case where the bootstrapping session is used with only one NAF is called “dedicated bootstrapping”. The need of dedicated bootstrapping is discussed in chapter 3.

Key derivation must be done when results of same bootstrapping session are used with multiple NAFs. However, when the bootstrapping session is used with only single NAF it is questionable whether key derivation is needed.

One advantage of key derivation is that NAF itself is authenticated, not just the network. However, the correct NAF_ID must be used both in BSF and UE to derive the correct key (Ks_NAF). This requires that the NAF_ID is known by the NAF when it requests the Ks_NAF from BSF over Zn interface. The address is typically known to a normal NAF who is identified by a single NAF_ID, but a specialized NAF that is doing for example virtual name based hosting (such as authentication proxy), may not know the NAF_ID at the time when it should fetch Ks_NAF from BSF over Zn interface. If key derivation is not used, this problem would not be present. However, both during TLS connection establishment and HTTP Digest authentication it is possible for the UE to indicate to the NAF (i.e., authentication proxy), what is the NAF_ID of the server that UE intends to contact. With the usage of TLS extensions [TLSext], UE can indicate the NAF_ID in the ClientHello message, and with the usage HTTP Digest authentication NAF_ID may be given in the Server header field of the HTTP request.

Thus, in order for the UE to be able to authenticate the NAF as part of the GAA it is recommended that option 2 should be considered as a valid state, and option 1 should not be used, i.e., key derivation should always be done.

2.3 Conclusion

The conclusion is that in order of the UE to be able to authenticate the NAF as part of the GAA the key derivation should always be done, and that only options 2 and 4 in Table 1 are be considered as valid states. Section 3 discusses the need of dedicated bootstrapping (i.e., option 2).

SA3 is asked to endorse that key derivation should always be done, i.e., only NAF specific key Ks_NAF is used in Ua interface.

3 Dedicated bootstrapping

A *dedicated bootstrapping* means that the use of bootstrapped key material Ks is restricted to a single NAF, i.e., UE cannot derive keys for other NAFs than the selected NAF. Currently, it is assumed that the decision on the usage of dedicated bootstrapping session (i.e., bootstrapping session is used for only one NAF) is done by BSF.

3.1 Need of dedicated bootstrapping

The decision whether to use dedicated bootstrapping session is made by BSF during Ub interface run with UE. The decision logic (discussed in Annex B) unnecessarily complicates the GAA procedures. Also, the requirements on why dedicated bootstrapping session is needed should be clarified. One of the requirements may be that NAF requires that bootstrapping session is “fresh”, but this already now possible since NAF can knows the lifetime of the key and is able send a key update request to UE (see section 3.2). If the lifetime is long enough (i.e., UE is using an old bootstrapping session), NAF may require UE to bootstrap again.

Also, it should be noted that if key derivation function (KDF) is secure, then multiple key derivations should be harmless.

Thus, it is recommend that multiple key derivations (named “shared bootstrapping” session, i.e., bootstrapping session is used with multiple NAFs) are always allowed. The key material (Ks) is used the derive NAF specific keys (Ks_NAFs) as long as the key lifetime given by BSF is valid.

3.2 Fresh bootstrapping session

If NAF requires a dedicated bootstrapping because it wants a fresh bootstrapping session, i.e., bootstrapping between UE and BSF has been done fairly recently (e.g., in the order of minutes), then this functionality is already possible with TS 24.109, subclause 5.2.4 “Bootstrapping renegotiation indication”, where it is stated that NAF can indicate to the UE that the bootstrapping session has expired, and that UE should bootstrap again.

The only difference between this and the dedicate bootstrapping is that dedicated bootstrapping session is restricted to single NAF, i.e., UE cannot use the key material Ks to derive keys for other NAFs.

Note that the parallel usage of multiple NAF specific keys (in UE) that are derived from different bootstrapping sessions is discussed in Annex C.

3.3 Conclusion

The need for dedicated bootstrapping session is questioned in this contribution. If no requirements for the need of dedicated bootstrapping are identified we suggest to use only “shared bootstrapping”. In general, it should be noted that if the key derivation function is secure, then the use of multiple key derivations should be harmless.

SA3 is asked to endorse that multiple key derivation should always be allowed, i.e., Ks can be used multiple times to derive NAF specific keys (Ks_NAFs) and that are used with multiple NAFs as long as key lifetime parameter is valid.

4 Proposal

SA3 is asked to endorse the following:

- key derivation should always be done, i.e., only NAF specific key Ks_NAF is used in Ua interface.
- multiple key derivation should always be allowed, i.e., Ks can be used multiple times to derive NAF specific keys (Ks_NAFs) and that are used with multiple NAFs as long as key lifetime parameter is valid.

The attached CR implements the required changes, i.e., both *key derivation* flag and *multiple key derivation* flag are removed.

5 References

[TS 33.220] 3GPP TS 33.220: “Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture”.

[TLSext] IETF RFC 3546: “Transport Layer Security (TLS) Extensions”, June 2003.

Annex A: Implications of key derivation to authentication proxy

The usage of key derivation has implications to authentication proxy (AP). Since AP functions as a reverse proxy, UE does not know that AP is handling the authentication for multiple application servers. If key derivation is used, that causes UE to use multiple keys (one per application server identified by FQDN) with AP, i.e., AP must be able handle and store multiple keys (Ks_NAFs) with same B-TID value.

The need for multiple keys can be avoided if the application servers behind the AP are addressed by a single hostname, and the distinction between different services is made in the URL. For example, all the operator services are under single hostname: operator.com, and the services are identified in the URL like this:

- Presence admin service: http://operator.com/presence; and
- Conferencing service: http://operator.com/conference.

This would also enable the use of single TLS tunnel from the UE to the AP without any modifications to the TLS implementations, since the AP is addressed by using a single hostname.

Annex B: Decision logic for dedicated bootstrapping

This section discusses about how the decision to do dedicated bootstrapping could be done in BSF, if dedicated bootstrapping is possible within GAA.

Preferred by operator: An operator may decide to run a separate bootstrapping session for each NAF that wants to authenticate the UE using GBA. In this case, for every NAF that requests a GBA-based authentication, UE is required to run bootstrapping procedure, and not to use any previous bootstrapping session that has already been used with other NAFs. This can be achieved by having for example, a configuration parameter in BSF to set the multiple key derivations option on or off.

Required by NAF: Also a NAF may require due to its security policy that the bootstrapping has been done only for it, i.e., no other NAF has rights to use the transaction identifier, and Ks is used to derive only one Ks_NAF.

If NAF is able to disallow multiple key derivations, i.e., bootstrapping session would be dedicated to one NAF, BSF needs know that the multiple key derivations flag on Ub interface should be set accordingly. This can be done two ways:

- **static case:** during bootstrapping session UE may communicate the identity of NAF that required GBA-based authentication to BSF. BSF would have a list of NAFs that require a dedicated bootstrapping session. This list would be populated with NAFs that have made an “a dedicated bootstrapping session” agreement with operator.
- **dynamic case:** during bootstrapping session UE itself may indicate to BSF that a dedicated bootstrapping session is required by NAF (or by UE). In this case, NAF needs to be able to indicate to the UE that a dedicated session is required.

In the static case, NAF can trust the operator that BSF bootstrapping session is dedicated.

In the dynamic case, NAF needs to be assured that the bootstrapping session is dedicated. This can be done in several ways, for example:

- the transaction identifier may have specific flag indicating that this is a dedicated bootstrapping session.
- an indication of a dedicated bootstrapping session may be given over Zn interface.

In addition to current parameters being sent over Ub interface, the dynamic case described above imposes new additional requirements in both Ua and Zn interfaces:

- It shall be possible for NAF to indicate to the UE that a dedicated bootstrapping session is required.
- It shall be possible for NAF to discover that the bootstrapping session is dedicated to the NAF.

Annex C: Parallel bootstrapping sessions in UE

There are scenarios where UE would use several NAF specific keys (Ks_NAFs) that have been derived using different bootstrapping sessions. By way of example, UE may have a single bootstrapping session in use with one or more NAFs identified by B-TID-1. Suppose that UE contacts a NAF that requires that the bootstrapping session is fresh (cf. chapter 3.2). In this case, UE initiate a bootstrapping procedure and establish a new bootstrapping session that is identified by B-TID-2. Now UE has several possibilities:

- UE will start to use the new bootstrapping session identified by B-TID-2 with all the NAFs it is currently communicating with, or

- UE will continue using the old NAF specific keys derived from older bootstrapping session identified by B-TID-1 and use B-TID-2 only with the new NAF.

If the first option is selected then this causes UE to re-authenticate itself towards all the NAFs that it currently communicating with. Also, this causes all these NAFs to fetch the correct key from BSF using the B-TID-2. If the second option is selected then this causes that UE is using several bootstrapping sessions in parallel (i.e., B-TID-1 and B-TID-2). However, if NAFs don't have problems with using the older bootstrapping session then there is not need to force the other NAFs to use the latest bootstrapping session. Thus, the parallel usage of several bootstrapping session in UE is the preferred.

Let's now consider the case where one of the NAFs that is using the older bootstrapping session (identified by B-TID-1) requests UE to bootstrap again because it considers the bootstrapping session to be too old. Note that in this phase, the other NAFs don't have a problem with using the older bootstrapping. Now UE has several possibilities to function again:

- UE will establish a new bootstrapping session identified by B-TID-3, and use that with the NAF, or

- UE will offer NAF to use the existing and newer bootstrapping session identified by B-TID-2.

The latter option is preferable here since it enables the reuse of existing bootstrapping session. If NAF finds this bootstrapping session unacceptable it can again indicate to the UE a refresher bootstrapping session is required (cf. chapter 3.2).

Let's now consider the case where the oldest bootstrapping session expires based on the key lifetime parameter. In this case, we also have the same two options than in the previous case, i.e., UE may establish a new bootstrapping session and use that with the NAFs, or it may offer NAFs to use the existing but newer bootstrapping session. The reasoning would apply here, i.e., some of the NAFs may accept to use the existing bootstrapping session, and some of the NAFs may require UE to bootstrap again. In the latter case, UE should take care that only one bootstrapping session is established although several NAFs have requested bootstrapping.