3GPP TSG SA WG3 Security — SA3#33 **S3-040346**
10-14 May 2004 **Reply-comments: S3-040218**
Beijing, China

Source: **Siemens, Ericsson (Commented by Gemplus 23/04/04) reply by Siemens 03/05/04**

Title: **GBA_U: Bootstrapping secrets to the UICC**

Document for: **Discussion and decision**

Agenda Item: **GBA**

1 Introduction

Contributions S3-040089/95/97 at SA3#32 provided a concept to bootstrap secrets to the UICC based on GBA, which could be used for MBMS. The draft SA3#32-report mentions:

"*It was agreed that whether a new UICC would work in an older GBA-ME terminal needed to be studied. It was also mentioned that the GBA-request flag on the Ub-interface could be superfluous and simplify the handling. It was clarified that MBMS as user of GBA_U would still need to realize own security procedures towards the UICC. Further input on these issues was requested. Siemens was asked to further develop the mechanism ands provide the contributions (for those parts that are relevant for MBMS) with the same contributions deadline as agreed for MBMS contributions to SA WG3 meeting #33.*

It was agreed as a working principle that the GBA_U is added as a generic mechanism, it is for further study to decide if it could be used for MBMS."

This contribution provides a further elaboration of the GBA_U concept (the generic part), taking into account the agreed working assumptions and remarks given at the meeting.

2 GBA_U concept description

2.1 Summary of the problem statements

- For GBA_ME as currently described within TS 33.220 v6.0.0 the ME receives CK and IK from the UICC, and concatenates these keys to form Ks. This and all further key derivation functions are implemented within the ME (e.g. Ks_NAF derivation from Ks).

- When running GBA_U the key Ks shall never leave the UICC. For some applications (ME security services) a key Ks_ext_NAF is needed within the ME (e.g. within http digest authentication). For other applications (UICC security services) a key Ks_int_NAF shall not be made available to the ME (e.g. for MBMS the key Ks_int_NAF may be used for securing the transfer of the MBMS Service Key to the UICC). A GBA_U aware UICC shall be able to supply keys to both types of services.

Conclusion: Two problems have to be solved.

a. The ME shall NOT be able to obtain a key used within UICC security services, but the ME shall be able to obtain a key used for ME security services. The solutions for this is described within section 2.2
b. The UICC has to be told that GBA_U shall be run. The solution for this is described within section 2.3

It is assumed that the specification of GBA_U will have no effect on the specifications for GBA (i.e. GBA_ME in our terminology), as currently specified in TS 33.220 v6.0.0.
For the purpose of this contribution we define a GBA-aware UICC as: A UICC capable of deriving both Ks_int and Ks_ext, whereby Ks_ext is given to the ME in the form of CK' and IK' and whereby the key Ks_int is kept secret within the UICC.

- When the ME wants to use Ks_ext for ME security service then it performs a key derivation step that is implemented on the ME to obtain Ks_ext_NAF.
  <G+>
  Why does GBA-aware UICC give Ks_ext to the ME? In S3-040095 contribution (SA3#32), the Ks_ext_NAF computation takes place on the UICC.
  <Sie>
  A GBA-aware UICC needs to give a Ks_ext to the ME to handle the situation where an ME (with only GBA_ME functions implemented) does not know that a GBA-aware UICC has been inserted. This type of ME will always perform GBA_ME derivation to obtain Ks_NAF from Ks. In case this full derivation would be done on the UICC this type of ME would again do a derivation leading to a key mismatch between network and terminal.

  Ks_ext is used to derive the Ks_ext_NAF keys, the retrieval of Ks_ext on the ME allows to retrieve all the NAF-specific keys (more specific the external keys, but NOT the internal keys). The storage of Ks_ext on the UICC avoids the retrieval of the derived Ks_ext_NAF. (Why should it be avoided for the external keys?) The level of security is higher. Moreover, the lifetime of Ks_ext could be longer if Ks_ext were stored on the UICC.
  <Sie>
  I think your suggested optimization to take advantage on the extended lifetime when stored on the UICC, can be realized but at the same time it shall be allowed that GBA_ME only ME be able to function with a GBA aware UICC. Together with the implementation of the function call to derive Ks_int_NAF from Ks_int on the UICC, the UICC could also derive Ks_ext_NAF from Ks_ext (which is stored internally on the UICC) and give it back to the ME which supports GBA_U.
  So, we recommend the computation of Ks_ext_NAF on the UICC as result of a second call which includes the key derivation parameters.

- When the UICC wants to use Ks_int for UICC security service then it performs a key derivation step to obtain Ks_int_NAF. This key derivation is implemented on the UICC. The derivation of keys Ks_int and Ks_ext can done by the UICC as a result of one call to the UICC (very similar to Rel99 authentication), and the derivation of Ks_int_NAF from Ks_int is then done as a result of a second call to the UICC which includes the key derivation parameters (NAF id etc).

The solution has to take care of different combinations of features with different terminal and UICC releases:

(0) It shall be guaranteed that the handling for the non-GBA authentication domains is unchanged.
This is achieved by administrating a GBA-aware UICC within a Rel-6 HSS and signaling explicitly towards the UICC and HSS when a GBA-run is performed. This explicit signalling towards the UICC would use RAND bits (see section 2.3) which allows the UICC to distinguish between GBA and non-GBA runs. The HSS will know that a GBA-run is involved (to consider setting the RAND bits) when the AV-request comes from a BSF. The BSF needs to be told by the HSS that GBA_U needs to be run, so that the BSF can perform the correct key derivations. **The BSF shall recognize this on the basis of the RAND bit-settings or alternatively on the basis of an explicit parameter on the Zh-interface**.
Note that a Pre-Rel6 ME supports neither GBA_ME nor GBA_U and therefore the ME will not be able to trigger bootstrapping functionality.

From (0) it is guaranteed in particular that:
(1) It shall be possible to use a Pre-Rel6 ME with a GBA-aware UICC.

(2) It shall be possible to use a Rel-6 ME with a GBA-unaware UICC.
In case the Rel-6 ME supports GBA_ME then the Rel-6 ME shall be able to obtain/derive a Ks_ext_NAF based on CK||IK received from the UICC. The needed functions for GBA_ME support can already be found within TS 33.220 v6.0.0.

(3) It shall be possible to use a Rel-6 ME with a GBA-aware UICC.
A Rel-6 ME may be able to support GBA_ME and in addition may be able to support GBA_U functions. The GBA aware UICC is administrated within a Rel-6 HSS and a GBA run for the UICC is explicitly signaled with a Special-RAND flag. In case the Rel-6 ME supports GBA_ME then the Rel-6 ME shall be able to obtain/derive a Ks_ext_NAF from a received Ks_ext. The Ks_ext shall have the same format as CK,IK. The key derivation functions for the ME to execute shall be the same as for case (2).
<G+>
Because of Ks_ext storage on the ME the level of security is also the same as for case (2).
<Sie>
This is not true, Ks_int_NAF is not available to the ME. You probably mean that the Bootstrapping frequency is in both cases still the same and no opportunity is taken to extend the lifetime of GBA_U keys

If the Rel-6 ME supports GBA_U (i.e. by supporting a UICC-feature like MBMS key storage), then it shall be able to supply NAF-dependent parameters to the UICC for deriving the key Ks_int_NAF from Ks_int. This key Ks_int_NAF shall be kept within the UICC.

<G+>
The derivation of Ks_ext_NAF shall take place on the UICC. Confer previous comments of this section.
<Sie>
See also before. One particular case still requires that Ks_ext_NAF can be computed by the ME, although I agree that the derivation can also be implemented on the UICC.

2.2 Delivering keys to two types of applications

Problem statement: 'The ME shall NOT be able to obtain a key used within UICC security services, but the ME shall be able to obtain a key used for ME security services'.
At the same time it shall be possible for the NAF to use both derived keys Ks_int_NAF and Ks_ext_NAF towards the UE. The security service that builds on top of GBA_U shall be able to make the distinction between these two keys. As an example the BM-SC may choose whether it wants the MBMS keys stored on the UICC or the ME secure storage. The way this is achieved is outside the scope of GBA.

Following figures describe the proposed key derivation steps. **Error! Reference source not found.** describes the proposed two-step key derivation for a GBA-aware UICC when GBA is run. **Error! Reference source not found.** describes the one-step key derivation which is used for a GBA-unaware UICC in case the ME wants to use GBA_ME.

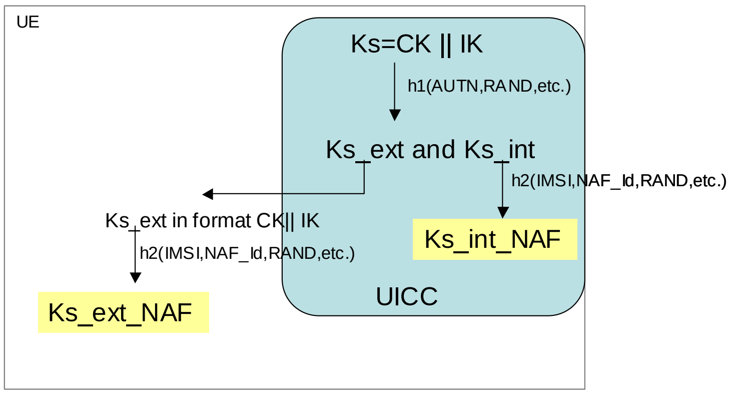

Figure 1: Key derivation for GBA-aware UICC when GBA-run was triggered

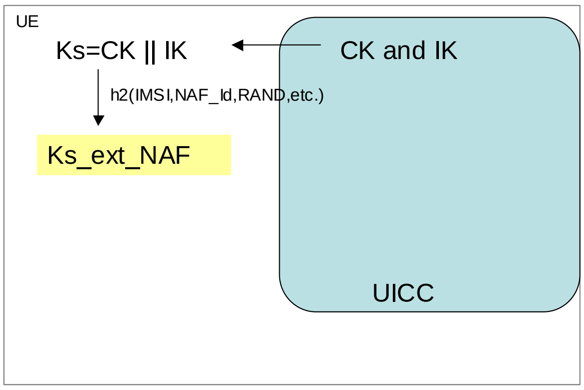

Figure 2: GBA_ME Key derivation

According to **Error! Reference source not found.**, two keys are derived whereby the first key Ks_int_NAF will remain internal to the UICC and the second derived key Ks_ext will be delivered from the UICC to the ME. The keys Ks_int_NAF and Ks_ext_NAF are derived from Ks = CK||IK using a two-step key derivation.

The input parameters to the first key derivation function h1 have to be restricted to the information that is available within the AV. AUTN and RAND could be used for this.

The second key derivation step is performed on the ME for deriving Ks_ext_NAF from Ks_ext using the function[1] h2 which is also used for a GBA-unaware UICC. The UICC itself derives an internal key Ks_int_NAF from Ks_int using the same function h2 (with the same input parameter), but securely implemented on the UICC. The second key derivation function h2 may include parameters like IMSI, NAF_Id and RAND. The parameter RAND is the random challenge in the AKA authentication vector. NAF_id is the DNS name of the NAF as is defined within TS 33.220. The ME shall not able to derive the Ks_int_NAF from Ks_ext. This is achieved by using function h1.

From the network point of view the procedure is as follows for a GBA-aware UICC.

- When recognising from the response of the HSS that GBA_U is to be run, the BSF derives both Ks_int_NAF and Ks_ext_NAF from Ks and the key derivation parameters in the same way as the UICC (h1/h2) and ME (h2 for Ks_ext), and sends both Ks_int_NAF and Ks_ext_NAF to the NAF.
- The NAF makes the choice between Ks_int_NAF and Ks_ext_NAF depending on the requirements. It shall be possible for the NAF to use both types of keys.

It could be considered to perform function h1 within the HSS instead of the BSF. Ks_ext and optionally Ks_int (for a GBA_U aware UICC) will then need to be carried over the Zh-interface. The disadvantage of this option is increased HSS load and impacted Zh-interface for transporting Ks_int. This paper therefore assumes that the h1-implementation is done within the BSF.

From the network point of view the procedure is as follows for GBA-unaware UICC.

- The BSF derives both only Ks_ext_NAF from Ks and the key derivation parameters in the same way as the ME, and sends only Ks_ext_NAF to the NAF.

NOTE: The key derivation procedures above are only given by way of example. Further study and input by SAGE is needed regarding the key derivation procedure in general and the input parameters required by it.

[1] The second key derivation function on the UICC may be different from the second key derivation function on the ME.

2.3 Enforcing a GBA_U run at the UICC

Within contribution S3-040095 two solutions were analyzed from which solution 1 (**Set one bit within the RAND to indicate to the UICC that GBA_U has to be run)** was taken as a working assumption. This section focuses on detailing the RAND solution and adapts it (with respect to S3-040095) were necessary.

The proposal from S3-040095 was to use **one** bit that has to be set differently by a Rel-6 HSS (AuC) to distinguish between a GBA-call involving a GBA_U aware UICC (bit x set to 1) and a GBA unaware UICC (bit x set to 0). The RAND bit will also NOT be set (bit x set to 0) in case the authentication is not for GBA-usage i.e. for the PS, CS, IMS and WLAN domain authentication, the CK and IK need still present at the ME and the authentication result CK and IK will still be the same as within Pre-Rel 6 networks.

A variant of solution 1 is to use a 32-bit Special-RAND flag (which may be different from A5/GEA proposal, but could also include a flag and context format as was proposed in SA3#32) in order to achieve the same goal that a GBA_U aware UICC recognizes a GBA_U run. The Special-RAND flag has to be set differently by a Rel-6 HSS (AuC) to distinguish between a GBA-call involving a GBA_U aware UICC and a GBA_U unaware UICC. The Special-RAND flag will also NOT be set in case the authentication vector is not for GBA-usage i.e. for the PS, CS, IMS and WLAN domain authentication as these authentication domains have no h1 function implemented within the core network. If the special-RAND flag is not set the original CK and IK are sent from the UICC to the ME and the authentication output will still be the same as within Pre-Rel-6 networks. **The additional advantage that comes with using a Special-RAND flag for GBA_U is that an operator does not need to upgrade the HSS to Rel-6 when starting to introduce GBA-aware UICC in the network, but no NAFs still use GBA_U.** This approach also ensures an independent and gradual introduction (and upgrade) of GBA-unaware UICC to GBA-aware UICC.

A GBA-unaware UICC will ignore the special RAND-bit pattern if it was to occur accidentally. A GBA aware UICC introduced in a Rel-6 Network will then be able to detect the Special-RAND bit-pattern and enforce the GBA_U handling. Such a UICC will pre-process the Ks:=CK||IK with h1 in order to make it impossible for the ME to derive Ks_int_NAF from Ks_ext. For a GBA aware UICC together with Pre-Rel-6 HSS (AuC) there is one slight disadvantage in that there will be a statistical increase 1exp-32 failed CK and IK mismatches between the Network and the UE. By flipping one bit of the RES when executing GBA_U on the UICC this mismatch can detected early during authentication already. The number of failed authentications due to the mismatches is considered negligible compared to other reasons for failed authentications (such as transmission bit errors).

It is proposed to adopt the Special-RAND flag proposal for GBA_U as it allows the operator to introduce GBA-aware UICC independently from upgrading the HSS.

2.4 The necessity of a GBA_U request flag?

As was described within section 2.1, an ME could include different feature combinations. It is possible that a GBA-aware UICC is inserted into an ME that does not support the GBA_U procedure to derive Ks_int_NAF from Ks_int on the UICC. If in that case a GBA_U type of key derivation was run then the NAF will receive both Ks_int_NAF and Ks_ext_NAF but will not be able to use Ks_int_NAF. This could lead to errors on the Ua-interface if the NAF falsely assumes that the ME supports GBA_U UICC interface procedures. So the NAF has to be made aware if the ME supports GBA_U UICC interface procedures.

For a first solution the NAF could rely on the fact that the NAF-application has knowledge of the UE feature support of that application i.e. GBA_U is only useful for the NAF, if in addition to the GBA_U functions on the card, also some security application is defined on the UICC which makes use of Ks_int_NAF (e.g. decryption of MBMS service key). The ME needs to support the required interface procedures for that UICC application. It could be assumed that the NAF obtains UE required feature knowledge via other means than the Zn-interface. How this is done is outside the scope of GBA_U and would then have to be addressed in the respective application, e.g. MBMS.

<G+>
Which MBMS SA3#33 contribution does indicate how the NAF obtains UE required feature knowledge?
<Sie>
Did the contribution refer to another contribution ? Note that SA4 still discusses MBMS user capabilities and that when knowing that the UE supports MBMS UICC storage is enough to derive that GBA_U is supported

It is important to know the element which sends information on the GBA_U capability of the ME.
The operator does not store information related to the UE features of a user since the UICC is a portable device.
Information coming from the ME cannot be trusted since fraudulent ME exists. E.g in MBMS context, the user is a potential attacker interested in hacking his ME.
<Sie>

A hack in the context of GBA cannot be used advantageously and this is different from the MBMS ME 'key distribution hack' for MTK. The BM-SC decides whether the MBMS service keys shall be stored at the UICC (with prerequisite GBA_U key) or the ME (with GBA ME run), which cannot be influenced by the malicious ME.

Suppose that the BM-SC want to deliver MSK to UICC storage of the UE.

Case 1: UE supports UICC MBMS storage (and also GBA_U) and downgrades his own settings towards the NAF to try to tell it that it only supports GBA_ME (or ME based MBMS services). If the BM-SC will have no ks_int_NAF available then it will reject the joining. But as this is a GBA-aware UICC, always GBA_U will be run and therefore the BM-SC will sent the key MSK encapsulated with Ks_int_NAF to the UICC, so a malicious ME cannot decrypt.

Case 2: UE does not support UICC MBMS storage. → Here an upgrade of the settings make no sense, anyhow the ME is never able to decrypt a key Ks_int_NAF[MSK].

A second solution relies on the use of a GBA_U request flag on the Ub-interface with following interpretation: '*An ME supporting GBA_U UICC interface procedures will set the flag on Ub*'. An ME not supporting GBA_U UICC interface procedures will not use the flag. If the UICC is GBA aware then the two step key derivation is done by the BSF, otherwise a one step key derivation. In case of a GBA_U aware UICC, the BSF will forward only Ks_ext_NAF to the NAF if GBA_U was not requested by the ME, otherwise both keys will be forwarded to the NAF. If the NAF receives only Ks_ext_NAF then it knows that the UE does not support GBA_U (although UICC might be GBA-aware already). But as many applications may rely on GBA_U, the NAF cannot conclude from the availability of the key Ks_int_NAF that the UE (ME or UICC) supports the application specific use of Ks_int (See previous paragraph). Therefore the GBA_U request flag cannot serve in telling the NAF that e.g. UICC-based MBMS key management is possible. A GBA_U request flag used in that way could only serve in holding a key Ks_int_NAF back from the NAF, which the NAF would not be able to use. This is not considered a sufficient advantage for introducing a GBA-request flag.

Conclusion: There are no advantages seen in using a GBA_U request flag on the Ub-interface.

<G+>
The use of GBA_U shall not depend on the ME. An attack consists in having a ME which claims that it does not support GBA_U, then the NAF has not longer choice between GBA_U and GBA_ME, it will continue with low level of security or will not provide the required service. In MBMS context, the user is a potential attacker so the attack on the ME is feasible, it would oblige the delivery of low value content only and there would be roaming problem in case of Visited Network mandating UICC-based only solution to protect their MBMS Service Keys.
<Sie>

See before, a hack does not work. The user hacking his own terminal in order not to be able to consume high valued MBMS ? The network (BM-SC) controls the key management and the appropriate level of key storage at the ME.

But, what is the reason to have Rel-6 ME supporting GBA_ME only? GBA is a Rel-6 feature, there is no legacy issue.
<Sie>

Modularity in terminal feature implementation could be a reason to allow Rel-6 terminals no to implement GBA_U interfaces.
GBA-capable ME shall support both GBA_U and GBA_ME to avoid the security issues identified in the previous comments of this section.
<Sie>

It is a decision that needs to be taken in SA3 whether we want that modularity or not.

2.5 Migration and impacts to introduce GBA_U.

Impacts on TS 33.220:

- Zh-interface: No impacts.
- Ub-interface protocols: No impacts.
- Zn-Interface: If GBA_U has been run the BSF forwards both Ks_ext_NAF and Ks_int_NAF and if GBA_ME has been run the BSF forwards only Ks_NAF to the NAF. The BSF knows this on the basis of a Special-RAND flag.
- A GBA-aware UICC shall implement GBA_U key derivation to generate Ks_ext and Ks_int from Ks.
- An ME supporting GBA_U type of services needs to be able to initiate the new GBA_U procedures on the UICC (i.e. UICC to derive Ks_int_NAF from Ks_int with function h2).
- NAF: If the BSF forwards both Ks_int_NAF and Ks_ext_NAF to the NAF, the NAF knows that the UICC supports GBA_U type of services. The NAF still has to obtain the knowledge about the ME capabilities specific to the NAF-application in order to decide whether it can (and wants) to use Ks_int_NAF. (The way the NAF obtains this information is outside of GBA_U feature scope and specific to each NAF application).
  <G+>
  Confer previous comments.

  <Sie>

  See also before

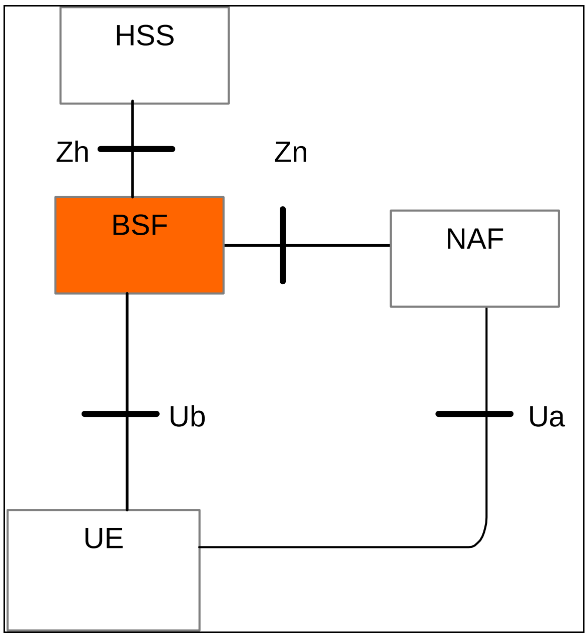

Figure 3: GBA architecture and references points from TS 33.220

Impacts on other specifications:

- TS 31.102 For GBA_U handling on UICC (including h1- key derivation) and Key derivation h2 on UICC (T3)
- TS 29.229 (CN4) to include the Zn-interface impact and for Zh-interface describe the handling at BSF and HSS for GBA_U.
- TS 24.109 to include GBA_U handling at BSF and UE for Ub-interface (CN1)

Migration issues:

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. 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 BSF needs to be upgraded as well; however the upgrade is considered small. The upgrade of the BSF to support GBA_U needs to occur no later than that of the HSS.

A NAF supporting GBA_U signals this on the Zn-interface. A BSF supporting GBA_U will only return Ks_ext_NAF to GBA unaware NAFs when a GBA_U run has been performed. .

Notes on migration:
The above analysis has been done with the assumption that the BSF will remain within the Home Network. This assumption is inline with the analysis that can be found within paper S3-040224 section 2 that concludes that there are several security disadvantages in placing the BSF within the Visited Network.
If the BSF would be placed within the Visited Network then the migration also becomes more complex as the migration path requires all BSFs to be updated when the HSS is updated. A problem then occurs if the UICC and HSS are GBA_U aware and the BSF (in the Visited Network) is not. Then the BSF will not perform the correct translation of the keys, if the HSS sends a GBA_U with Special-RAND. In this case the UE and NAF will always end up with different keys. A possible solution to avoid this is to mandate that a GBA_U aware BSF requests GBA_U with Special-RAND and a GBA_U aware HSS only sends GBA_U AVs when requested. That the HSS recognizes that the AV-request comes from a BSF that supports GBA_U, can be done either by explicit parameter or by administration within the HSS.

3 Conclusion

The proposed concept has been designed carefully to avoid any impacts on the ME based Generic Bootstrapping Architecture (TS 33.220 section 4). The Change-requests to TS 33.220 (which are present within contributions S3-040216/217 to this meeting), show that this goal has been achieved. Furthermore, the issues that were raised at last meeting have been solved and the migration issues are well understood such that the concept has reached stability at stage 2 level. The contributing companies therefore propose to adopt the proposed GBA_U concept as described within this document.

SAGE needs to be involved in the specification of the key derivation functions h1 and h2. The function h1 is needed for GBA_U in addition to a key derivation function h2 as needed for GBA_ME.

4 Change tracking of main changes with respect to SA3#32-concept.

- Removed GBA_U request flag as no advantages are seen (see section 2.4).

- Introduces a Special-RAND flag for GBA_U in order to avoid upgrading the HSS if at least one GBA aware UICC is used.

- A GBA aware UICC shall be able to provide GBA-service to both ME security services and UICC security services i.e. shall generate Ks_int(_NAF) and Ks_ext (A possible example is http digest authentication for MBMS key request and the subsequent key delivery).
  <G+>
  This contribution proposes the storage of Ks_ext on the ME while S3-030095 proposes the storage of this key on the UICC. Cf previous comments.
  <Sie>
  See my suggestion and explenation before.

- A two step key derivation is performed in stead of the one-step key derivation function.

- A GBA_U run generates a different RES then a non GBA_U run.