3GPP TSG SA WG3 Security — S3#32 **S3-040097**
9-13 February 2004
Edinburgh, UK

Source: **Siemens**

Title: **Using GBA_U within MBMS**

Document for: **Discussion and decision**

Agenda Item: **6.20: MBMS**

1 Introduction

As highlighted within SA3#31, SA3 should require only one MBMS delivery mechanism/protocol towards the UE. This is advantageous while at the network side then the investment is lower but also the OPEX[1] can then be minimized. A companion document to this meeting has developed a concept for bootstrapping a secret key to the UICC. It is based on the concept of GBA_ME which was already a candidate[2] to be used for MBMS services to the ME. By the introduction of GBA_U in addition to GBA_ME it allows an harmonization on the bootstrapping phase and gives the MBMS_ME a single protocol view on the network, independent from the type of MBMS service. While for MBMS with low- or medium-value content it may be quite appropriate to use GBA_ME, for MBMS with high-value content GBA_U shall be used.

After this GBA-step the same MBMS key delivery protocol (Ua-interface) should be used for both types of Ks_NAF if SA3 envisages using the same protocols for ME and UICC-based MBMS solutions. The selection on the Ua-interface protocol is however not part of this contribution.

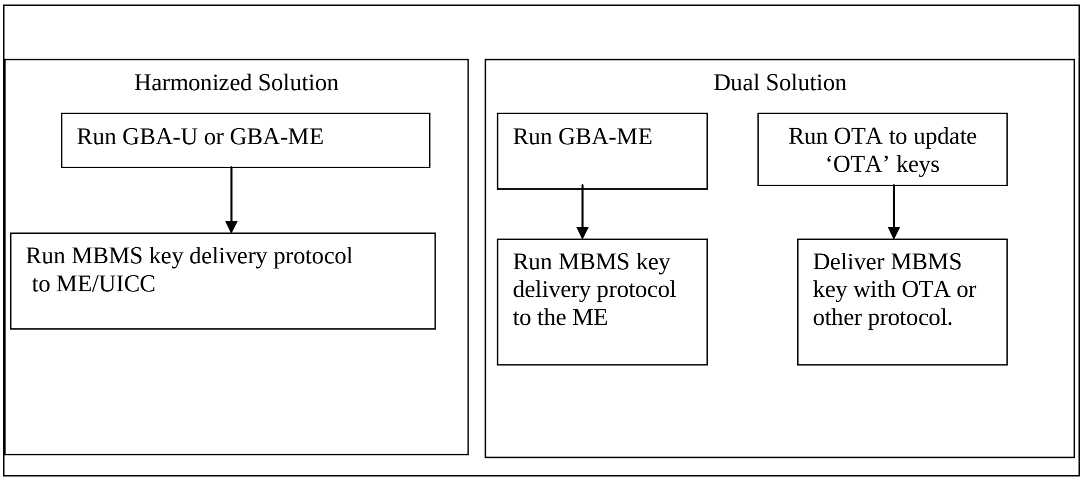
Harmonized Solution
Run GBA-U or GBA-ME
Run MBMS key delivery protocol to ME/UICC
Dual Solution
Run GBA-ME
Run MBMS key delivery protocol to the ME
Run OTA to update 'OTA' keys
Deliver MBMS key with OTA or other protocol.

Figure 1: Harmonized and Dual solution

This contribution shows further details how MBMS can make use of the GBA_U solution from the viewpoint of the ME.

[1] Operational costs

[2] See MBMS working assumption reached at SA3#31

2 How the Ua-protocol can use GBA_U generated keys on the UICC.

This section describes how an MBMS-application on the ME can use the GBA_U generated key Ks_int_NAF in order to show how the Ua-interface protocol (e.g. MBMS key management) does interact with the UICC.

There is a two-step approach taken place to deliver an MBMS-key to the UICC (similar as with GBA_ME)

Step-1: Run http-digest-AKA to create the GBA_U key (Ks and Ks_int_NAF).

Step-2: Run Ua-interface protocol (MBMS key delivery protocols) secured by Ks_int_NAF from step-1.

Possibly a re-run of step-2 can be performed (without requiring step-1) using the still valid Ks_int_NAF existing on the UICC. The NAF shall be able to control the lifetime of the keys Ks_int_NAF, Ks_ext_NAF. How this can be done optimally is for further study.

During step-1 the ME obtains a temporary Transaction Identifier (TID) from the http-digest-AKA run. The TID is needed within step-2 to relate the MBMS-key management message with the GBA-keys stored within the BSF. The ME obtains the TID only after successful authentication check at the BSF. So the TID is not available at the UICC to reference the right Ks_int_NAF. The RAND however is a suitable parameter that can be used at the ME to uniquely identify the key. So the ME has to store the TID for network GBA_U referencing purposes within protocol Ua and the RAND for referencing the right Ks on the UICC. This requires that a GBA_U UICC stores the RAND and the Ks internally when bootstrapping.

From one BM-SC (NAF) several MBMS-services could be delivered. All these services could make use of the same UE-individual bootstrapping run. The same NAF delivering low- and high value content could use the Ks_int_NAF from the GBA_U run.

When intending to use the bootstrapping keys towards a NAF the ME calls a procedure to derive the sub-keys and to assign the NAF_ID_n to the derived keys. This is done by calling a function with input parameters (IMSI, RAND, NAF_ID_n)[3] , the output is Ks_ext_NAF.

Within step-2 the ME initiates the Ua-protocol and uses the TID as a GBA reference. The NAF (BM-SC) retrieves Ks_ext_NAF and Ks_int_NAF from the BSF and uses these keys as appropriate to protect the point to point MBMS Master Key delivery.

[3] The same parameters as will be used for GBA-ME, inputs are dependent on the exact KDF realization).

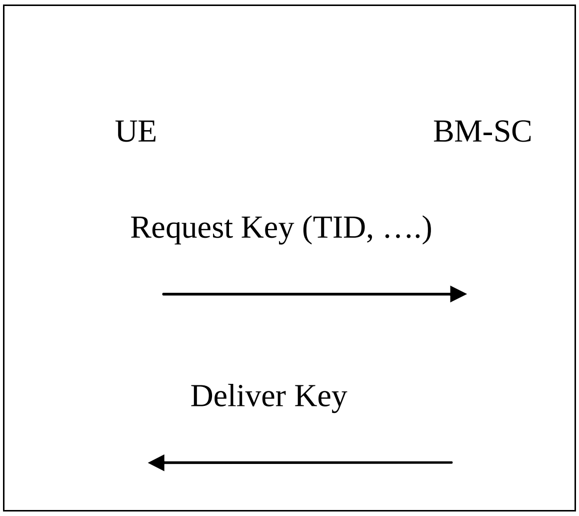

Figure 2: High level MBMS key delivery (See TS 33.246) on Ua-interface

With repetition of step-2 a GBA-less step is meant. Now no TID is given to the BM-SC within the MBMS key request. The ME may query the availability of a bootstrapping key shared with the BM-SC by calling a procedure to check if a valid Ks_ext_NAF or Ks_int_NAF exists for NAF_ID_n and use it as appropriate.

3 Conclusion

This contribution showed how the ME can handle the GBA_U secrets for MBMS and explained the advantages in using GBA for both ME and UICC based MBMS services. Siemens therefore proposes to adopt the working assumption that the point to point MBMS key delivery protocol shall use a GBA bootstrapped secret Ks_xxx_NAF to protect the MBMS service specific key delivery. This Ks_xxx_NAF was either bootstrapped to the ME using GBA_ME or bootstrapped to the UICC using GBA_U.