3GPP TSG SA WG3 Security — S3#34 **S3-040xxx**
6 - 9 July 2004
Acapulco, Mexico

Source: **Axalto, Gemplus, OCS**

Title: **GBA_U Scenarios and Rel 6 MEs capabilities**

Document for: **Discussion and decision**

Agenda Item: **GBA**

1 Introduction

At SA3 #33 the concept of GBA_U was introduced in TS 33.220 [TD S3-040413]. An important point, which needs further clarification, is the migration path from GBA_ME to GBA_U solutions. Related to this new functionality the definition of the following GBA entities could be the following.

GBA_U aware NAF: Able to use Ks_int_NAF for specific security purposes. From the NAF perspective the usage of Ks_ext_NAF and Ks_NAF is equivalent. No matter of which GBA type has been run (either GBA_U or GBA_ME) between the BSF and UE.

GBA_U aware UICC: Able to perform h1 derivation (Ks_ext & Ks_int) and h2 derivation for Ks_int_NAF and for ks_ext_NAF (if that alternative is approved by SA3). This capability in the UICC will be likely indicated in the corresponding Service Table in the UICC (e.g. GBA security context activated in the USIM Service Table file).

A GBA_U aware ME: Able to ask the UICC to perform h1 derivation (Ks_ext & Ks_int) and h2 derivation (for Ks_int_NAF and Ks_ext_NAF*)

*if h2 derivation in the UICC for Ks_ext_NAF is approved by SA3.

GBA_U aware BSF Able to obtain/send GBA_U specific AV and perform h1and h2 derivations.

The following different scenarios could be available.

	UICC	ME	BSF	NAF	***Comments***
1)	-	√ or -	√ or -	√ or -	Only GBA_ME is supported.
2)	√	√	√	√	GBA_U is run. NAF may use Ks_int_NAF (e.g. MBMS)
3)	√	√	√	-	GBA_U is supported. NAF uses Ks_ext_NAF
4)	√	-	√	√	Only GBA_ME is supported. NAF cannot use Ks_int_NAF
5)	√	√	-	√	Only GBA_ME is supported. NAF cannot use Ks_int_NAF
6)	√	√	-	-	Only GBA_ME is supported. NAF uses Ks_NAF
7)	√	-	√	-	Only GBA_ME is supported. NAF uses Ks_NAF
8)	√	-	-	√	Only GBA_ME is supported. NAF cannot use Ks_int_NAF
9)	√	-	-	-	Only GBA_ME is supported. NAF uses Ks_NAF

2 SCENARIO Analysis

In order to limit interoperability and deployment issues, it seems highly suitable to analyze non-desirable scenarios and avoid them by standardizing the needed sets of Rel-6 features:

-Scenarios 5 and 8 are not relevant and can be discarded, since the introduction of a NAF using GBA_U is not possible without the relevant actions in BSF.

5)	√	√	-	√	Only GBA_ME is supported. NAF cannot use Ks_int_NAF
8)	√	-	-	√	Only GBA_ME is supported. NAF cannot use Ks_int_NAF

-Scenario 3 states the fact that from a non-GBA_U aware NAF the usage of GBA_U or GBA_ME is equivalent. Non-GBA_U NAFs can be used with GBA_U bootstrapping.

3)	√	√	√	-	GBA_U is supported. NAF uses Ks_ext_NAF

-Scenario 6 depicts the case of an Operator deploying GBA_U aware UICC and not upgrading the BSF yet.

6)	√	√	-	-	Only GBA_ME is supported. NAF uses Ks_NAF

Some considerations on this Scenario are shown in contribution [] (“Alternative to Special Random or AMF indication for GBA_U: MAC indication”)

-Scenario 4, 7 and 9 shows the case when the ME is not GBA_U capable and a GBA_U capable UICC is inserted.

4)	√	-	√	√	Only GBA_ME is supported. NAF cannot use Ks_int_NAF
7)	√	-	√	-	Only GBA_ME is supported. NAF uses Ks_NAF
9)	√	-	-	-	Only GBA_ME is supported. NAF uses Ks_NAF

These 3 scenarios may cause significant deployment and interoperability problems, which may likely be avoided. The following two considerations summarize them:

GBA_U and BSF: The decision on choosing either GBA_ME or GBA_U shall be taken by the BSF based on subscription information. In other words, a GBA_U capable BSF is not able to know ME capabilities and thus, will typically provide GBA_U Authentication Vectors if a GBA_U aware UICC is involved.

As discussed in [TD S3-040346], the indication in the Ub interface (GBA_U request flag) of the ME capabilities is not appropriate, neither from the security nor from the usage perspective.

So, when a GBA_U capable UICC is involved, the BSF will provide a GBA_U Authentication Vector.

GBA_U requires specific key derivation process in the UICC. This includes detection of GBA_U AV, h1 derivation and h2 derivation for ks_int_NAF (and for ks_ext_NAF if that alternative is approved by SA3). Both derivations do require modifications in the existing ME-UICC interface (i.e. likely implemented as a specific GBA_U security context in the Authenticate command) and specific GBA_U parameter storage in the UICC (e.g. TID). The bootstrapping mechanisms will then fail unless the ME supports this GBA_U UICC interface.

The main consequence is that, because of ME features, an operator is not able to choose GBA_U security even if UICC, NAF and BSF have already been updated.

Applications using GBA_U:

Some applications in the UE may apply security functions using Ks_int_NAF even if the ME is not able to address services provided by a GBA_U capable NAF.

An archetype example is a SIM Toolkit application using ks_int_NAF to offer some security services (e.g. banking application…). In this case, even if the ME does not support any service/application using GBA_U internal keys (as e.g. MBMS), GBA_U bootstrapping is however required.

Another example is a downloaded ME application (e.g. a middlet) being able to offer services based in a combination of Ks_ext_NAF and Ks_int_NAF (e.g. using appropriate interfaces with a javacard applet).

If a Rel 6 GBA capable ME doesn't support GBA_U, a large number of applications using GBA security architecture will not be possible.

2.1 PROPOSAL

As shown in the la previous analysis these 3 scenarios (4, 7 and 9) may offer significant deployment and interoperability problems, which may likely be avoided by mandating GBA_U support in GBA capables MEs.

Two other considerations could be taken into account:

a) From the ME perspective, running GBA_ME or GBA_U bootstrapping functions is quite similar. Differences are limited to small modifications in the ME-UICC interface and eventually slight modifications in internal APIs to access/produce ks_ext_NAF derivation. Furthermore GBA is a Rel6 feature, so these modifications can be taken into account in new products without any backward compatibility problem.

b) Existing requirements in MBMS already mandate support of both GBA_ME and GBA_U in MBMS capable terminals.

This contribution propose the addition of the following requirement to GBA:

Rel 6 ME supporting GBA shall support both GBA_ME and GBA_U. This implies that GBA capable MEs shall support GBA_U specific ME-UICC interface. GBA_U functions in the ME-UICC interface will be used when a GBA_U capable UICC is inserted.

From this requirement, no specific inferences on the usage of GBA_U shall be implied. In other words, GBA_U bootstrapping procedures shall be supported by a GBA capable ME regardless if the specific ME applications may or may not use GBA_U specific keys with a GBA_U capable NAF.

A CR [S3-040478] implements this proposal.