3GPP TSG SA WG3 Security — SA3#35 **S3-040821**
October 5-8, 2004
St Paul's Bay, Malta

3GPP TSG SA WG3 Security — S3#35 **S3-040774**
5 – 8 October 2004
Malta

Source: **~~Gemplus, Axalto, Oberthur~~ Nokia, Siemens, Ericsson**

Title: **GBA: Support of GBA_U capabilities for Rel-6 MEE~~e~~s commented by Nokia, Siemens, Ericsson**

Document for: **Discussion and decision**

Agenda Item:

1. Introduction

In SA3#34, several papers on the support of GBA-U capabilities for Rel-6 MEs were presented and discussed during the evening session and some of them provided incomplete or misleading information. This paper provides some clarifications and corrections.

2. Clarifications concerning S3-040491

S3-040491 [1] analysed the proposal “GBA-aware ME support both GBA_U and GBA_ME”, but some of the arguments are misleading. This section provides some clarifications concerning the following items:

2.1. GBA and low-end MEs in Rel-6

Several SA3#34 contributions changed the scope of the requirement for Rel-6 GBA_aware MEs to support both GBA_U and GBA_ME:

S3-040491 [1]

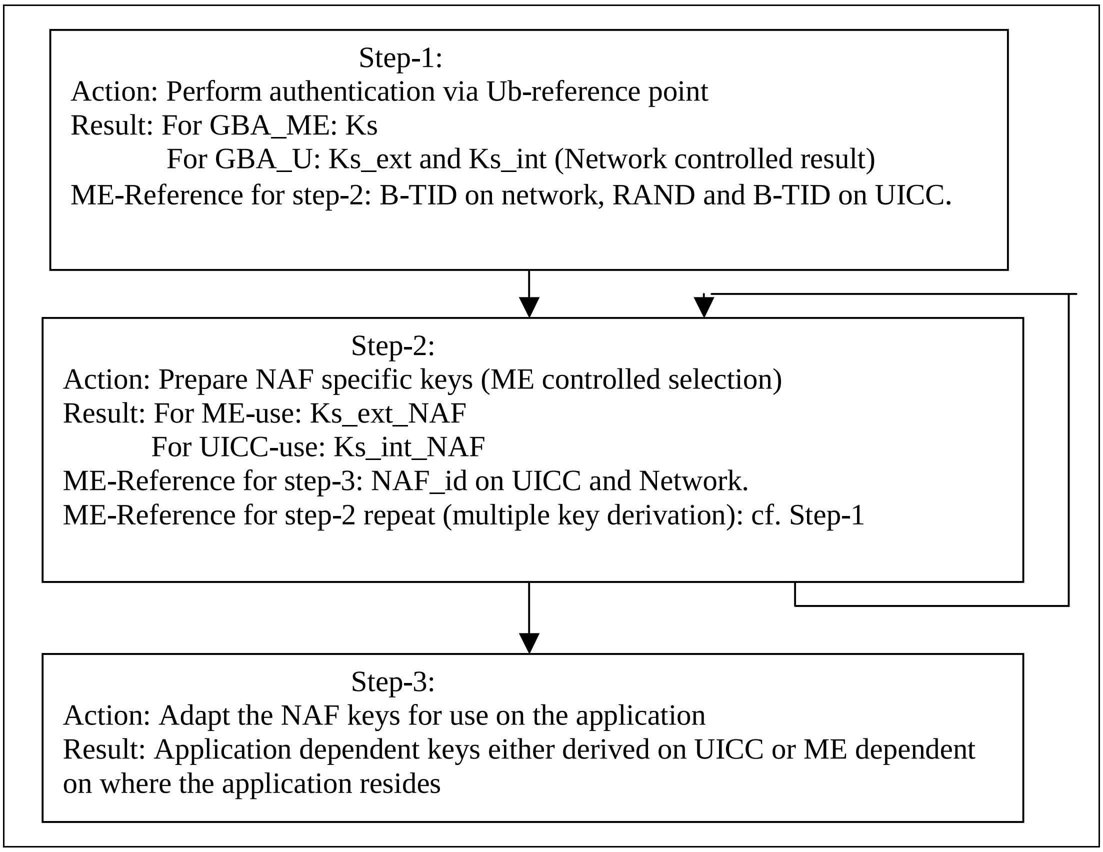
- *“It should be possible to bring lower-cost mobiles on the market that have dedicated limited functionality e.g. a Rel-6 ME that is manufactured for VGCS (ciphering) or GSM-only ME shall not be obliged to implement GBA.”*

S3-040655 (GBA_U Evening session report):

- *“Nokia, Siemens, and Ericsson stated that GBA_U should not be made mandatory, especially as "low-end" terminals in Release-6 would probably not use GBA_U.”*

Clarification:
All Gemplus/Axalto/OCS contributions state that “all Rel-6 **GBA-aware** MEs shall support both GBA_U and GBA_ME mechanisms”. This requirement to implement both GBA-ME and GBA_U concerns only MEs supporting GBA, it does not oblige low-end terminals (e.g ME for VGCS, or GSM-only ME) to implement GBA_U.

2.2. Generation and usage of Ks_xx_NAF



Step-1:
Action: Perform authentication via Ub-reference point
Result: For GBA_ME: Ks
For GBA_U: Ks_ext and Ks_int (Network controlled result)
ME-Reference for step-2: B-TID on network, RAND and B-TID on UICC.

Step-2:
Action: Prepare NAF specific keys (ME controlled selection)
Result: For ME-use: Ks_ext_NAF
For UICC-use: Ks_int_NAF
ME-Reference for step-3: NAF_id on UICC and Network.
ME-Reference for step-2 repeat (multiple key derivation): cf. Step-1

Step-3:
Action: Adapt the NAF keys for use on the application
Result: Application dependent keys either derived on UICC or ME dependent on where the application resides

Fig 1: Steps for setting up and using GBA_U key material

Step-1 includes the AUTHENTICATE call to the UICC.
Step-2 would be needed for:

- Ks_ext_NAF derivation and storage (derivation either on the ME or on the UICC).
- Ks_int_NAF derivation and storage on the UICC.

Step-3 is application dependent.

Step-2 and step-3 mix
S3-040491 gives the impression that Step 2 and step 3 can be combined.

- *"An ME that supports GBA_U shall support both step-1and step-2 procedures. But steps 2 and 3 may be executed by /combined with calling one or more applications"*
- *"From this there are several possibilities for the realization of the step 2 and 3:"*

Clarification:
GBA_U is proposed as a generic bootstrapping mechanism to provide shared key material between the UE and the NAFs (Ks_ext_NAF and Ks_int_NAF). GBA_U consists only of step-1 and step-2; step-3 is application dependent.

Comparing bootstrapping and key derivation procedures
Figure 1a depicts how GBA_U would be used as described in CRs S3-040783 and S3-040784. The difference compared to figure 1 is that old step 1 from figure 1 is divided to new steps 1 and 2 in figure 1a. The old steps 2 and 3 are the new steps 3 and 4 respectively.

If the ME is GBA_U-unaware, only step 1 is executed by the ME. If the ME is GBA_U-aware then steps 1 and 2 are both executed during the bootstrapping phase. Step 3 and 4 are executed by GBA_U-aware ME if it is required by a GBA_U-aware service, i.e, MBMS.

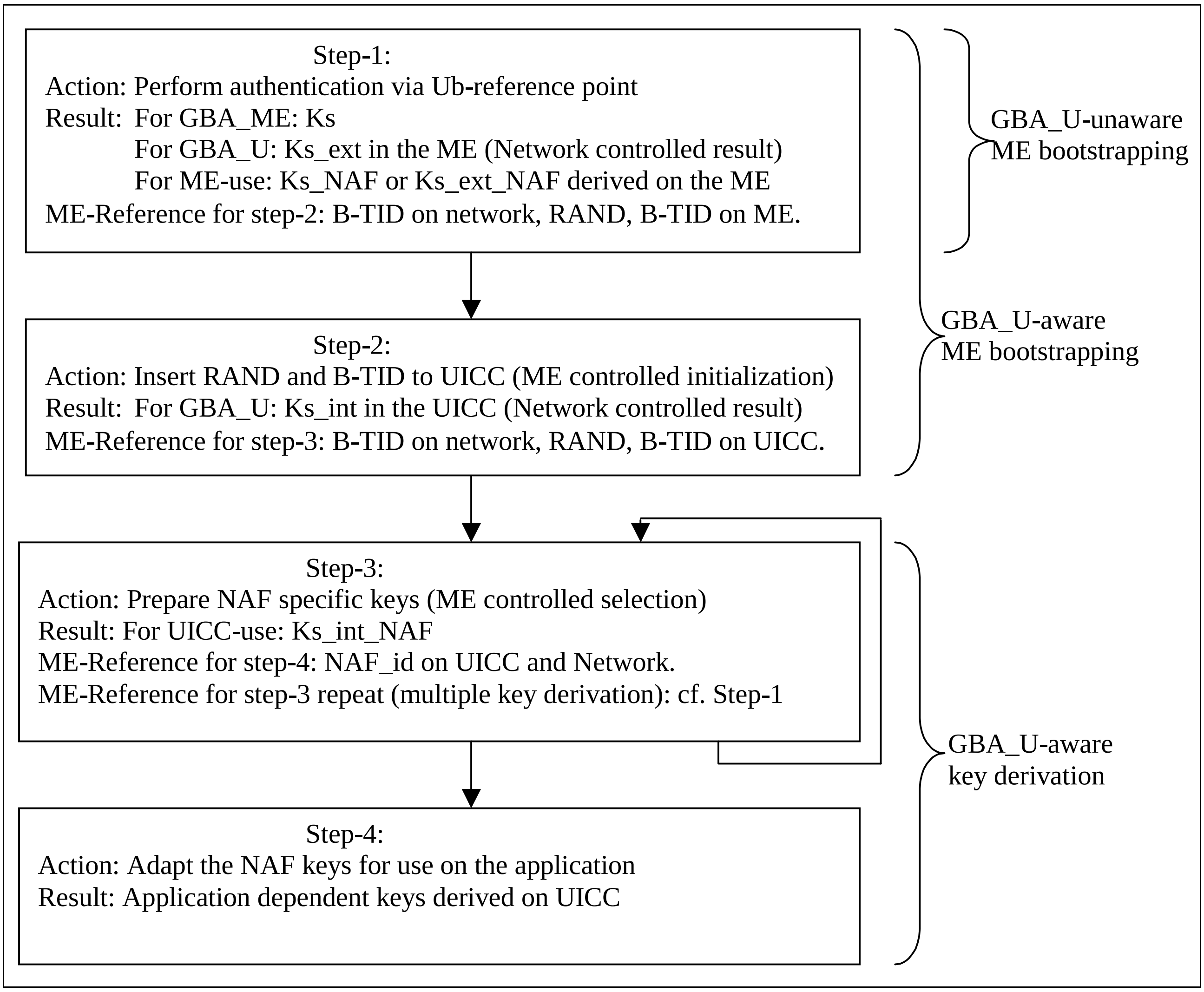

Step-1:
Action: Perform authentication via Ub-reference point
Result: For GBA_ME: Ks
For GBA_U: Ks_ext in the ME (Network controlled result)
For ME-use: Ks_NAF or Ks_ext_NAF derived on the ME
ME-Reference for step-2: B-TID on network, RAND, B-TID on ME.
Step-2:
Action: Insert RAND and B-TID to UICC (ME controlled initialization)
Result: For GBA_U: Ks_int in the UICC (Network controlled result)
ME-Reference for step-3: B-TID on network, RAND, B-TID on UICC.
Step-3:
Action: Prepare NAF specific keys (ME controlled selection)
Result: For UICC-use: Ks_int_NAF
ME-Reference for step-4: NAF_id on UICC and Network.
ME-Reference for step-3 repeat (multiple key derivation): cf. Step-1
Step-4:
Action: Adapt the NAF keys for use on the application
Result: Application dependent keys derived on UICC
GBA_U-unaware
ME bootstrapping
GBA_U-aware
ME bootstrapping
GBA_U-aware
key derivation

Fig 1a: Steps for setting up and using GBA_U key material based on S3-040783 and S3-040784.

2.3. Processing delay

- *The execution of a step-2 call to the UICC does have the disadvantage of adding additional processing delay (calling a UICC function) for Ua-interface."*

Clarification:
Currently, a call to the UICC represents only few tens of ms.

Remark:
If all Rel-6 ME support GBA_U, then it could be possible to modify the scheme to derive Ks_xx_NAF since SA3 decided at SA3#34 meeting to study the possibility to replace Ks_ext and Ks_int with a single Ks key. In case of the use of Ks instead of Ks_int and Ks_ext, the processing time for step-1 would decrease, so the global processing time for GBA_U would decrease, cf. [1]

2.4. Ks_int_NAF use

S3-040491 states that MBMS is the only use of GBA_U and Ks_int_NAF.

- *"Even if we do mandate that a Rel-6 ME supports step-2 interfaces procedures separately, GBA_U support on the ME will not be useful until there is an UICC application that can make use of it and the ME supports these application interface functions."*

- Conclusion section*:" In order for the UE to take advantage of the GBA_U key Ks_int_NAF, the UE needs to have an application that uses the Ks_int_NAF. For the mentioned Rel-6 applications in section 3 this may mean the availability of some generic cryptographic functions on the UICC that can make use of the Ks_int_NAF. These UICC functions are not yet available for Rel-6 and it is probably too late to start standardization on this. In the absence of such UICC-applications the support of ME-UICC interfaces procedures (step-2) at the ME for these functions has no added value as Ks_ext_NAF has to be used anyhow".*

Clarification:
For Rel-6, Ks_int_NAF could be used by other applications than MBMS key management. GBA_U is a generic mechanism to provide shared key material between the UE and the NAF, the use of Ks_int_NAF does not always require the definition of a new ME-UICC interface since some existing UE applications (i.e. specified in release 6) may use those keys without involving the ME-UICC interface.
For instance, (U)SIM Toolkit Application [3] could use Ks_int_NAF and Ks_ext_NAF to secure communication over a BIP channel (Bearer Independent protocol). Besides, a Java Middlet in a JSR177-based ME could access cryptographic functions provided by the (U)SIM application using Ks_int_NAF and Ks_ext_NAF (JSR177 is a standardized API allowing communication between a UICC and a J2ME ME).

These mechanisms allow the use of GBA_U shared keys to establish secure associations with operator or third parties servers, many applications could be proposed, e.g. banking applications, service provider's applications.

3. Implementation cost

In order to specify GBA_U, T3 agreed at T3#32 meeting the creation of a GBA Security Context in the AUTHENTICATE command with two specific modes: Bootstrapping mode and NAF Derivation mode, cf [2] and [3]. So, the support of GBA_U for Rel-6 GBA-aware MEs does not require the implementation of a new command, it only implies the implementation of the GBA Security Context for the AUTHENTICATE command.

Moreover, at SA3#34 meeting, SA3 proposed an alternative to derive Ks_xx_NAF in case of Ks_ext stored on the UICC, Ks_int and Ks_ext could be replaced with a single Ks key. This proposal is studied in an SA3#35 contribution [1]. This alternative decreases the number of key derivations and the complexity on UE and BSF sides.

The cost of the GBA_U implementation in a GBA-aware ME is not significant.

4. Inter-operability and security

Despite the negligible cost of the GBA_U implementation in a GBA-aware ME, an operator implementing GBA_U in their network (this will be at least the case for MBMS) will not be able to take full advantage of GBA_U security benefits [4] unless the GBA_U is mandated in the ME. In fact, when both the operator's BSF and the user's UICC are GBA_U aware, which will be likely the case on the long run, the BSF will perform a GBA_U bootstrapping procedure. In such a case, if the GBA-aware ME does not support GBA_U, the whole procedure will fail. This may lead the BSF to fall back systematically to GBA_ME when the bootstrapping procedure fails even though the reason for failure may be quite different from the one mentioned above.

5. Reminders

In addition to security improvement and the possible use of the Ks_int_NAF key to secure applications without a systematic need to define a new UICC-ME interface, the following reasons have also been identified to require that all Rel-6 GBA-aware MEs shall support both GBA_U and GBA_ME (Cf S3-040477 [5] presented at SA3#34 meeting):

- The support of GBA_U by all Rel-6 GBA-aware MEs decreases deployment and interoperability problems.
- GBA is a Rel-6 feature so these modifications can be taken into account in Rel6-MEs without any backward compatibility issue.

6. Conclusion

The cost for all GBA-aware MEs to support GBA_U consists of implementing the "GBA security context" of the AUTHENTICATE command. This cost is not significant compared to the security benefit provided by the storage of Ks_ext on the UICC. Moreover, failing to support GBA_U on all Rel-6 GBA-aware MEs would prevent deployment and would result in interoperability problems.

So, we kindly ask SA3 to require that all Rel-6 GBA-aware ME shall support both GBA_U and GBA_ME. A CR implements this proposal [6].

We kindly ask SA3 to require that the support for GBA_U in Rel-6 GBA-aware MEs shall be optional. CRs S3-040783 and S3-040784 implement the changes (depicted in figure 1a).

7. References

[1] TD S3-040xxx, "Alternatives for GBA_U derivation", Gemplus, Axalto, Oberthur, SA3#35

[2] TD T3-040450, "GBA_U ME-USIM interface", T3#32

[3] TD T3-040456, "GBA _U ME-ISIM interface", T3#32

[4] TD S3-040xxx, "Finalisation of GBA_U procedures", Gemplus, Axalto, Oberthur, SA3#35

[5] TD S3-040xxx, "GBA_U scenarios and Rel-6 MEs capabilities", Axalto, Gemplus, Oberthur, SA3#34

[6] TD S3-040xxx, "CR: Support of GBA-U for all Rel-6 GBA-aware MEs", Gemplus, Axalto, Oberthur, SA3#35