3GPP TSG SA WG3 Security — S3#36 **S3-040950**
23 – 26 November 2004
Shenzhen

Source: **Gemplus, Axalto, Oberthur**

Title: **GBA_U: GBA_U derivations**

Document for: **Discussion and decision**

Agenda Item: **6.9.2**

1. Introduction

The location of the storage of Ks_ext is missing in the current version of TS 33.220 [1]. At SA3#35 Malta meeting, the following agreement was reached concerning GBA_U: *"If the UICC supports GBA_U, Ks_ext shall not leave the UICC and as a direct consequence, a terminal will behave differently if a GBA-capable UICC is inserted, to when a legacy UICC without GBA support is inserted"* cf SA3#35 meeting report [2]**.**

This contribution proposes alternative CRs to complete TS 33.220 taking into account the agreement reached at SA3#35 meeting.

2. Ks_int and Ks_ext stored in the UICC

The CR "GBA_U: Storage of Ks_ext in the UICC" [3] proposes the storage of Ks_ext in the UICC, it corresponds to the agreement reached at SA3#35 meeting concerning the location of the storage of Ks_ext for GBA_U: "*if the UICC supports GBA_U, Ks_ext shall not leave the UICC"*.

The present document also introduces an alternative solution, which takes into account the proposal made at SA3#34 meeting to optimise the GBA_U bootstrapping procedure by removing at least one key derivation procedure. Ks_int and Ks_ext could be replaced with a single key Ks. This alternative proposal, which is in line with SA3#35 decision on the storage of Ks_ext, is described in the following section.

3. A single bootstrapping key Ks

In GBA_U Ks_ext and Ks_int are the bootstrapping keys since they are computed in the UICC during the GBA_U bootstrapping procedure and they are used to derive NAF-specific keys (Ks_ext_NAF and Ks_int_NAF). The agreement reached at SA3#35 meeting "*if the UICC supports GBA_U, Ks_ext shall not leave the UICC"* means that the "**GBA_U bootstrapping keys**" (Ks_ext and Ks_int) shall not leave the UICC.

The optimisation proposed at SA3#34 meeting consists in replacing the bootstrapping keys Ks_ext and Ks_int with a single key Ks. This proposal fulfils the agreement reached at SA3#35 concerning the storage of the GBA_U bootstrapping keys in the UICC since Ks would be the GBA_U bootstrapping key and it would not leave the UICC.

3.1. Current methods to derive NAF-specific keys

The current TS 33.220 [1] proposes the following key derivations for GBA-U:

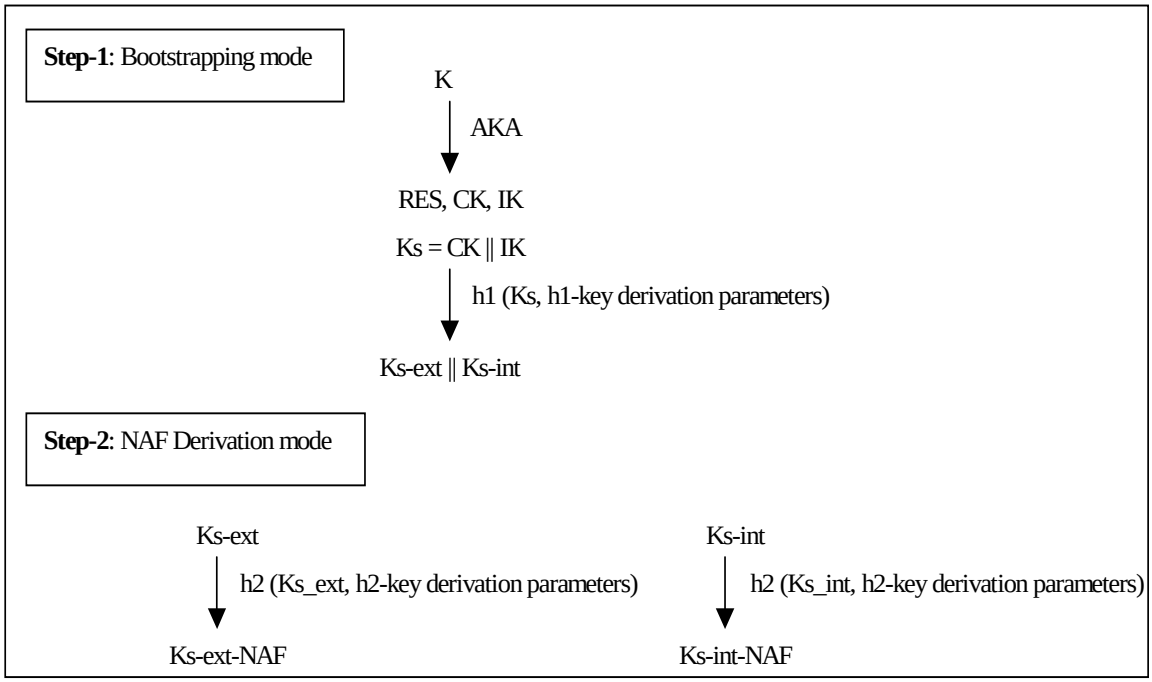

Figure 1: Current derivation methods to compute Ks_ext_NAF and Ks_int_NAF

Ks_ext and Ks_int are 128-bit keys
The definition of the key derivation function is left to ETSI SAGE.

3.2. Methods to derive NAF-specific keys from Ks

In case of Ks_ext and Ks_int replaced with a single key Ks, different schemes are possible to derive the NAF-specific keys from Ks:

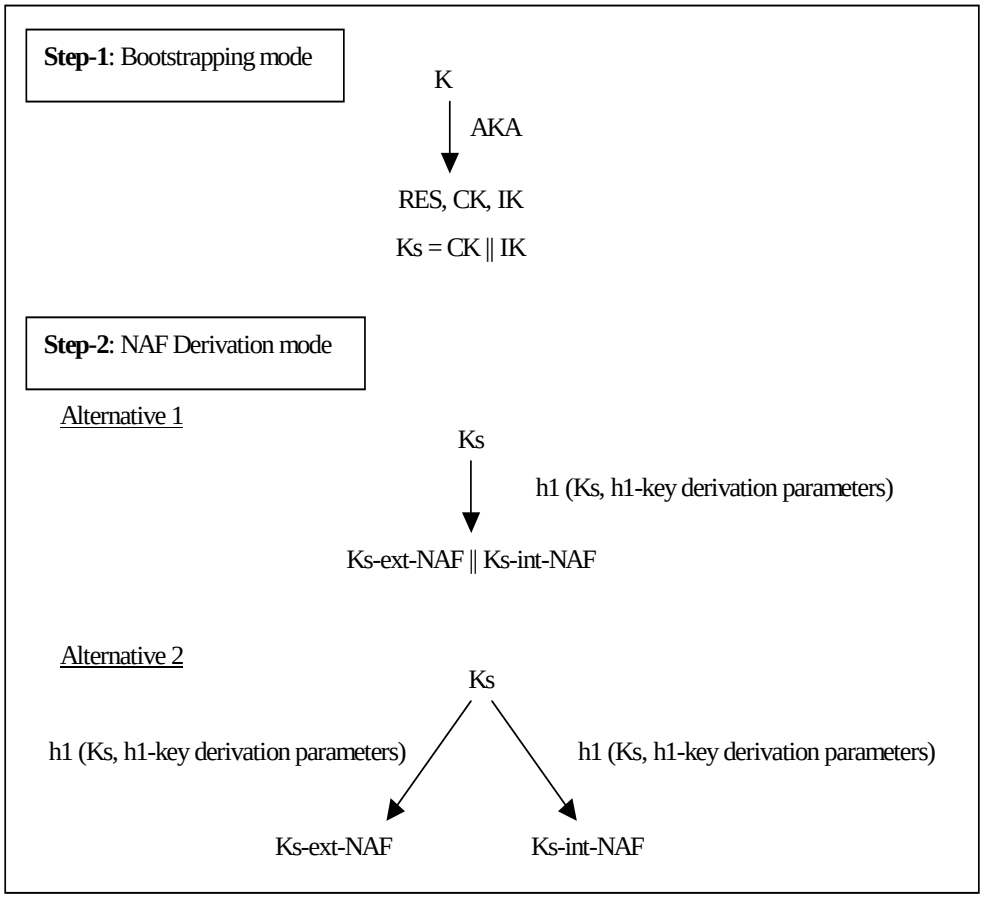

Figure 2: Alternatives for Ks_xx_NAF derivation from Ks

- **Step-1**
  There is no longer h1 key derivation function in the bootstrapping mode. The single key Ks is the concatenation of CK and IK.

- **Step-2**
  - Alternative-1:
    The output size of h1 key derivation function is at least two times the length of Ks_xx_NAF key.

  - Alternative-2:
    The computation of Ks_ext_NAF and Ks_int_NAF requires two executions of the key derivation function h1, one derivation for the external key Ks_ext_NAF and one derivation for the internal key Ks_int_NAF.
    The h1 key derivation function may either be the same function, in which case the differentiation between external and internal keys will happen via different parameters, or be a set of two different functions, h1_ext and h1_int.
    The output size of the h1 key derivation function is at least the length of Ks_xx_NAF key.

3.3. Choice of the key derivation function

Requirement to fulfill:
The solution should fulfill the following requirement: the key size of Ks_xx_NAF with GBA_U shall be the same than the key size of Ks_NAF with GBA_ME .

The following table shows the features of the h1 key derivation function (KDF) according to the alternative:

	GBA_ME Step 2: NAF derivation mode KDF h1	**GBA_U Step-2**: NAF Derivation mode	
		Alternative 1: KDF h1	**Alternative 2:** KDF h1
NAF-specific key: **256-bit key**	Output size: at least 256 bits (e.g. HMAC-SHA-256) The KDF is executed once	Output size: at least 512 bits The KDF is executed once	Output size: at least 256 bits (e.g. HMAC-SHA-256) The KDF is executed twice

Figure 1: Description of the key derivation function according to the alternative

The choice of the alternative depends on the complexity to implement the key derivation function and the time processing to execute the command.

3.4. Benefits of the use of single key Ks

The use of a single key Ks reduces the bootstrapping time since in the Bootstrapping mode the UICC would have to perform only the concatenation of CK and IK whatever the alternative selected for the NAF Derivation mode. It also reduces the implementation complexity in the BSF, as the GBA_U procedure will be similar to the GBA_ME procedure at least for the Bootstrapping mode (Step-1).

So, it will lead to better performance in the UICC and the BSF.

3.5. CR proposal

Therefore, we recommend the use of a single key Ks. The choice of the alternative and the key derivation function for the NAF Derivation mode is left to SA3 and ETSI SAGE.

The CR "Optimization of the GBA_U key derivation procedure" [4] implements the alternative 2 (KDF is executed twice). This CR could be updated during SA3#36 meeting to reflect SA3's decision.

4. Conclusion

A CR on the location of the storage of the bootstrapping keys has to be approved at SA3#36 Shenzhen meeting in order to complete TS 33.220. This contribution provides two alternative CRs:

- CR "GBA_U: Storage of Ks_ext in the UICC" [3] proposes the storage of Ks_ext in the UICC
- CR "Optimization of the GBA_U key derivation procedure" [4] proposes the use of a single bootstrapping key Ks which does not leave the UICC.

Both CRs propose that the bootstrapping keys do not leave the UICC, we recommend the approval of the CR "Optimization of the GBA_U key derivation procedure" since the use of a single GBA_U bootstrapping key Ks optimizes GBA_U procedure.

5. References

[1] TS 33.220, v6.2.0 “Generic Authentication Architecture, Generic bootstrapping architecture” Rel-6

[2] “SA3_35_Draft_Rep_v004.doc”; Draft Report of SA3#35 version 0.0.4

[3] TD S3-040xxx, GBA_U: Storage of Ks_ext in the UICC”, Gemplus, Axalto, Oberthur, SA3#36

[4] TD S3-040xxx, “CR: Optimization of the GBA_U key derivation procedure”, Gemplus, Axalto, Oberthur, SA3#36