3GPP TSG SA WG3 Security — S3#32 **S3-040095**
9-13 February 2004
Edinburgh, UK

Source: **Siemens**

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

Document for: **Discussion and decision**

Agenda Item: **6.9.2: GBA**

1 Introduction

This contribution works out a solution to bootstrap secrets to the UICC which has been called GBA_U within this contribution. The architecture that has been developed for bootstrapping secrets to the ME (GBA_ME) shall be re-used as far as possible. The BM-SC could be considered as the a NAF-type that can make use of GBA_U, but also current NAF-types like the presence admin server could make use of the GBA_U characteristics. One important characteristic is the long term secure storage at the UICC. Therefore a longer lifetime may be assigned to keys generated by GBA_U than for GBA_ME leading to less bootstrapping runs. Another advantage is that keys stored on the UICC allow plastic-roaming. A companion document to SA3#32 describes how the changes could be accommodated within TS 33.220 optimally. When SA3 agrees with the proposed concept, Siemens volunteers to finalize the stage 2 concept for GBA_U till next SA3-meeting by writing the needed CRs.

2 GBA_U Concept

2.1 Requirements and assumptions

Sections 2.3 and 2.4 will explore the possible solutions to bootstrap a secret key Ks to the UICC using GBA_U. But before doing this, we make the following assumptions for the proposed alternative solutions.

1. As few network nodes as possible shall be able to obtain or derive the bootstrapped GBA_U secret.
2. The ME shall not be able to obtain or derive the bootstrapped GBA_U secret.
3. The effects on AKA should be minimized.
4. Impacts on Rel-6 UICC are allowed, but should have no effect on the authentication of Pre-Rel-6 cards. *It is assumed that all functions of Pre-Rel6 card are upgradeable using manual point-of-sale provisioning, but preferably using OTA-mechanism*.

We make following assumptions with respect to the effects on the GBA-architecture currently used for GBA_ME.

5. GBA-functional changes are still allowed for Rel-6 but should be minimized. One important reason for changing some GBA_ME details may be to plan a smooth migration path for NAFs upgrading from GBA_ME to GBA_U.
6. The GBA_U architecture shall be NAF-type independent.

2.2 Problem description

- For GBA_ME 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 = CK||IK shall never leave the UICC. For some applications (ME security services) the key Ks_NAF is needed within the ME (e.g. within http digest authentication). For other applications (UICC security services) the key Ks and its derived keys shall not be made available to the ME (e.g. for MBMS the key Ks_NAF may be used for transferring the MBMS Master Key to the UICC).

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. Solutions are discussed within section 2.3
b. The UICC has to be told that GBA_U shall be run. Solutions are discussed within section 2.4

It is further assumed that GBA_U is run whenever the UE is capable of GBA_U. (Otherwise there may be confusion between keys Ks_NAF derived by means of GBA_U and those derived by means of GBA_ME.)

2.3 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'.

There are two possible ways to solve this
a) Introduce a second set of derived keys, where one remains on the UICC and the other is transferred to the ME.
b) Realise ME-service specific security functions on the UICC.

Within solution approach b, as an example, the digest authentication function will have to be realized on the UICC. (This is similar with the EAP-SIM termination on the card). The same would hold true for all other security functions in a Ua protocol which make use of the derived key. This would necessitate new implementations on the UICC for every new Ua protocol requiring security services.

Therefore solution (a) is the preferred one.

Following solution approach (a), two keys are derived whereby the first key Ks_int_NAF would remain internal to the UICC and the second derived key Ks_ext_NAF would be delivered from the UICC to the ME. The key Ks_int_NAF and Ks_ext_NAF are derived from Ks = CK||IK and further key derivation parameters including IMSI, NAF_Id_n and RAND. NAF_Id_n is sent by the ME to the UICC when the ME calls the UICC for key derivation. The parameter RAND is the random challenge in the AKA authentication vector containing CK and IK. The ME shall not able to derive the Ks_int_NAF from Ks_ext_NAF.

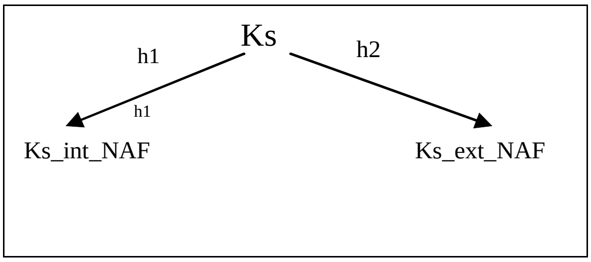

Figure 1: Key derivation

From the network point of view the procedure is as follows.

- The BSF derives Ks_int_NAF and Ks_ext_NAF from Ks and the key derivation parameters in the same way as the UICC, 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.

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 required input parameters in particular. But it is envisaged that the two derived keys Ks_int_NAF and Ks_ext_NAF can be derived from Ks and the key derivation parameters "in one go", i.e. using one function call.

Examples: For MBMS Ks_int_NAF would be used as the point to point key to transfer a MKK (MBMS Master key) to the UICC, whereas Ks_ext_NAF would be used as the point to point key to transfer a MKK to the ME. For Ut-reference point authentication with http-digest using GBA_U, the ME would ask Ks_ext_NAF from the UICC.

2.4 Enforcing a GBA_U run at the UICC

Problem statement: The UICC has to be told that GBA_U shall be run.

Within this section two solutions are discussed:

1. Set one bit within the RAND to indicate to the UICC that GBA_U has to be run.
2. Pre-process the AKA inputs (RAND and K) before feeding them into the authentication functions.

Solution 1: Set one bit within the RAND to indicate to the UICC that GBA_U has to be run.

The authentication generation function is currently authentication domain agnostic (i.e. PS, CS, IMS and WLAN domain use the same procedures , a sequence number management array mechanism may be applied per domain but the result CK, IK and RES value are the same for the same RAND and are always available for the ME).

The proposal is to use one bit that has to be set differently by the AuC to distinguish between GBA_U and GBA_ME users. The PS, CS, IMS and WLAN domain can be considered as GBA_ME type of users from the UICC point of view (although these authentication domains do not use GBA_ME mechanisms for authentication) as they currently require the CK and IK to be present at the ME.

Assumption: **A network with NAFs/UICC's that wants to use GBA_U shall first introduce a Rel-6 AuC that generates GBA_U RANDs in response to a GBA_U request).**

When a GBA_U capable UICC would be used without an AuC capable of GBA_U RAND generation, then the authentication in the other domains may fail (PS, CS, IMS, WLAN). All further text for this solution assumes a Rel-6 AuC.

A GBA_U capable UICC introduced in a Rel-6 Network will then be able to enforce GBA_U handling. Such a UICC shall hold back CK and IK and not give it to the ME if the GBA_U RAND bit has been set. Non GBA_U capable UICC will ignore the RAND setting.

Only these users that have a GBA_U capable UICC in their possession shall be able to request GBA_U handling, otherwise a malicious user could fake GBA_U handling and as such bypass the security settings of the UICC. So within the HSS a setting per user is needed which specifies that GBA_U functions are available or not on the UICC.

The ME shall indicate within the initial GET request on the Ub-interface that GBA_U has to be run. The BSF will return an error if the users profile settings do not indicate GBA_U capability. If GBA_U has been run the BSF forwards both Ks_ext_NAF and Ks_int_NAF, if GBA_ME has been run the BSF forwards only Ks_ext_NAF to the NAF. The RAND implicitly indicates GBA_U during the sequence number re-synchronization procedure.

The ME shall know what type of GBA-run shall be invoked (e.g. depending of the type of service it is intended for, or according to stored settings).

A minor disadvantage is that it would sacrifice a bit of the RAND, also for those existing authentication domains where it is not intended for (CS, PS, ..). It should be noted here, though, that the mechanism proposed here does not interfere with the special RAND mechanism proposed to counter the attack against A5/2 for the following reason: GBA_U itself does not need to apply encryption of A5/x or GEAx, therefore AV's destined for GBA_U need not to apply the special-RAND encoding to counter the attack against A5/2.

<u>Solution 2</u>: Pre-process the AKA inputs (RAND and K) before feeding them into authentication functions.

A simple and fast solution would be to compute a 128-bit keyed hash using K and RAND and using the result as key K. Another solution is to use a SHA-1 hash on K and using the result as key K for the authentication process.

The concept therefore relies on realizing two different authentication procedures which can be called by the ME and which produces different outputs. One authentication procedure, which is called by the current authentication domains and by GBA_ME, and a new authentication procedure, which is called by GBA_U applications. Calling the new procedure will result in a different Ks even with the same RAND as input. In addition, the RES would be different so the BSF can detect a user trying to defeat the network.

Again we make the assumption (similar as with solution 1) that first a Rel-6 AuC shall be introduced within the network. But if GBA_U capable UICC would be introduced within a pre-Rel6 network this will create no problem as the GBA_U function wouldn't be usable by the ME.

A GBA_U capable UICC introduced in a Rel-6 Network will then be able to enforce GBA_U handling. Such a UICC shall hold back CK and IK and not give it to the ME if requested by the ME. A user running GBA_ME but telling the network that GBA_U is run (and vice versa), results in a failed GBA-authentication.

The ME shall indicate within the initial GET request on the Ub-interface that GBA_U has to be run. 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 has to use a different type of AV dependent on whether GBA_ME or GBA_U is invoked. The AuC shall be able to generate a GBA_U type AV when requested by the BSF. The HSS (AuC) has to know whether the sequence number resynchronization token AUTS was generated on the basis of GBA_U or GBA_ME when coming from the BSF.

The ME shall know what type of GBA-run shall be invoked (e.g. depending of the type of service it is intended for, or according to stored settings).

<u>Evaluation:</u>

Proposal 2 has the advantage that it allows to generate Ks in a complete secure way (unless key K or the authentication algo's would be compromised). A medium disadvantage is that it impacts the core of the AKA procedures. Extra signalling during SQN-resynchronization may be necessary.

Proposal 1 creates minimal impacts at the UICC and the AuC. The UICC controls the security of the GBA-generated keys based on an explicit indication within RAND. A disadvantage is that HSS administration is necessary.

Based on this assessment and the described consequences (taking into account the section 2.1 requirements and assumptions), proposal 1 is preferred.

2.5 Migration and impacts to introduce GBA_U.

This section is described under the assumption that the preferred solutions from previous sections are adopted.

Impacts:

- Zh-interface: The GBA relevant profile parameter shall be transported via the BSF to the NAF. An indication that GBA_U has to be run needs to be introduced (i.e. the BSF has to ask GBA_U authentication vectors).
- Ub-interface protocols: The initial GET message has to include a GBA_U request parameter.
- 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_ext_NAF to the NAF.
- UICC: shall implement GBA_U procedures and the key derivation functions (KDF) to generate Ks_ext_NAF and Ks_int_NAF from Ks.
- The ME needs to be able to handle the new GBA_U procedures on the UICC.

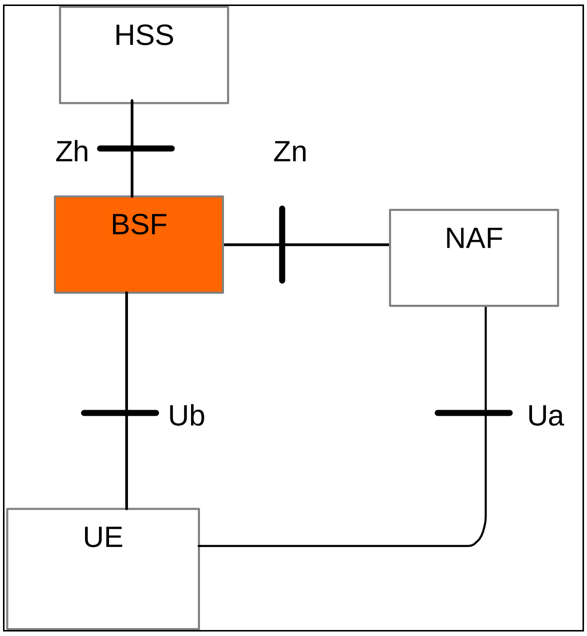

Figure 2: GBA architecture and references points from TS 33.210

Migration issues:

The HSS(AuC) shall be upgraded first before NAFs are introduced in the network that use the GBA_U services and the UICC GBA setting has to be administrated.

The BSF needs to be upgraded as well, however the upgrade is considered small.

A NAF using only GBA_ME type of service can ignore Ks_int_NAF if forward to him.

3 Conclusion

Siemens proposes to adopt the proposed concept after deciding on each of the proposed preferences in section 2.3 and 2.4. Assuming SA3 agrees with realizing GBA_U, Siemens volunteers to finalize the stage 2 concept for GBA_U till next SA3-meeting by writing the needed CRs.