3GPP TSG SA WG3 Security — S3#35 **S3-040811**
5 - 8 October 2004, Malta, EU

Source: **3**
Title: **Enhanced key freshness in GBA**
Document for: **Discussion/Decision**
Agenda Item: **GBA**

Introduction

One possible weakness of GBA, as it is currently specified is that there is no way for a NAF or a UE to guarantee a KS(_ext/int)_NAF is fresh without a re-run of AKA. To counter this, it is proposed to allow the UE and NAF to both generate a random number that is used as input to the key derivation function. The additional functionality needed to achieve this is described in this contribution. It is also considered whether it is worth allowing the BSF to include a random input into the key generation.

Problem description

A weakness of GBA as a key agreement scheme is that there is no guarantee of freshness of the KS(_ext/int)_NAF without forcing a re-run of AKA. That is if a NAF that requests a key from the BSF, that key might have already been used. In general key freshness is a desirable property of any method used to establish keys and should be included in GBA.

The following describes a situation where the lack of key freshness with GBA may cause a problem. A UE contacts a NAF and establishes a secure connection with that NAF using some Ks(_ext/int)_NAF. This connection is finished with and all the data related to it gets deleted. A short time after that the UE connects back to the same NAF and the same Ks(_ext/int)_NAF is calculated and used to secure this new connection (this assumes that no run of AKA had been done between connections). This means that the same key can be used to secure two successive connections between a UE and a particular NAF. Furthermore this implies that replay attacks may be possible on the Ua interface unless the particular protocol used does something to avoid the possibility of replay attacks between successive and unrelated sessions. This is a requirement that is currently not expected of the Ua interface.

Proposed solution

An easy way to avoid this problem is to have some random input generated by the UE and NAF (strictly only one is needed but using both provides freshness guarantees to both ends) used in the generation of Ks(_ext/int)_NAF. Implementing this functionality requires the following additions to GBA

- generating random numbers at the UE and NAF,
- passing the UE generated number to the NAF and vice versa,
- passing both the random numbers from the NAF to the BSF,
- the chosen key derivation function needs to support having these random numbers as inputs
- extension to B-TID to support multiple keys for same NAF under one run of AKA

Adding this functionality also delay the use of the Ks(_ext/int)_NAF by one message, since it is no longer possible for the UE to calculate the key until it has received a message from the NAF. As the random numbers can be passed in already existing messages, adding this functionality really only adds a small cost in overall complexity. Of course any or all of these random inputs to the key derivation function could be made optional.

A logical (although perhaps unnecessary) extension to this is to allow the BSF to input its own random number into the key derivation function. This guarantees that every KS(_ext/int)_NAF sent out from the BSF is unique. This would protect against a spoofed NAF_id obtaining the key in use between a UE and NAF.

GBA_U and Key freshness

When using GBA_U, the ability to of GBA to generate a **fresh** KS_ext_NAF from a KS_ext held on the UICC has an additional advantage, namely to authenticate the presence of the UICC without a run of AKA. Checking the presence of the UICC by generating a new Ks_ext_NAF from the current Ks_ext rather than by generating a new Ks_ext to then calculate a new Ks_ext_NAF saves the need to run

AKA between UE and BSF, saves possible signalling between the BSF and the HSS to fetch AVs and avoids the consumption of an additional AVs. These savings are inline with the design principles of GBA and provide further justification for the functionality proposed in this contribution (given the ability of GBA_U to provide a Ks_ext_NAF to the ME derived from a key held on the UICC – alternatively this could provide some justification for generating Ks_ext_NAF from a key held on the UICC).

To implement this functionality would require the BSF to signal to the UE that is acceptable to generate a new Ks_ext_NAF from a key currently held on the UICC rather than run AKA to provide a new Ks_ext_NAF. Outside this no additional functionality is required and for a particular Ua interface, it would be optional to use.

Conclusion

This contribution discusses a weakness of GBA, in that it does not provide any key freshness, and proposes a possible solution to that problem. It is proposed that SA3 accept the proposal in principle at this meeting and then CRs can be prepared for agreement at the S3#36 in November. Additionally it may be necessary to send an LS to CN1 and CN4 to inform them of this new functionality in order for them to get it into their specifications (24.109 for Ua interface and 29.109 for the Zn interface respectively) for the December plenary.