3GPP TSG SA WG3 Security — SA3#34 **S3-040534**
06 - 09 July 2004
Acapulco, Mexico

Source: **Ericsson**

Title: **On the introduction and use of UMTS AKA in GSM**

Document for: **Discussion**

Agenda Item: **GERAN network access security**

1 Introduction

Since the August 2003 publication of the Biham et al. attack on GSM A5/2, a number of countermeasures have been suggested. While they solve the immediate problems, we believe it may be desirable to solve other, more general security problems in GSM at the same time as doing a general "revision" of GSM security. The use of UMTS AKA in GSM would mean great improvements. Besides protecting against many of the problems that enable the various A5/2-related attacks, the improved AKA as such would increase the security and possibly mitigate yet unknown attacks on e.g. A5/1 or GSM in general. As discussed below, UMTS AKA can also co-exist with and, in fact, support already proposed A5/2 attack fixes.

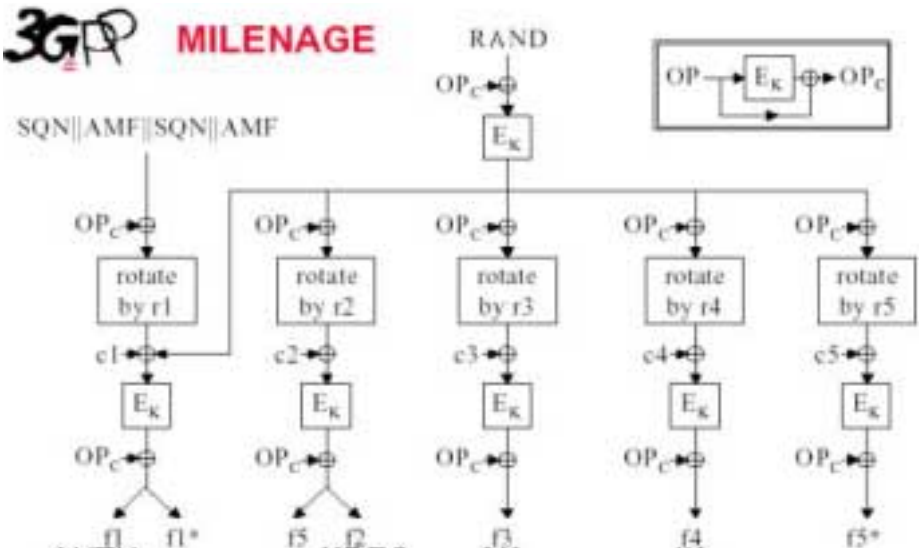
However, it may be difficult to convince users to replace their SIMs by new ones supporting the f1-f5 algorithms, many users would probably like to keep their old GSM SIM as long as they are GSM subscribers, at least for simple convenience reasons. The following text discusses an inexpensive way to introduce UMTS AKA in GSM without the need for SIM replacements.

2 Overview of Proposal

The following should be considered only a draft for discussion; details need~~s~~ to be more carefully investigated. However, the text should suffice to judge the merits of the approach, including security issues.

UMTS AKA specifies the algorithms f1-f5 to derive keys, response etc. In the following we shall for concreteness and brevity assume that Milenage is used to implement f1-f5, though the general principle seems applicable to f1-f5 in general.

Milenage has the following form:



One possible way to implement Milenage using a SIM is as follows. Note that Milenage is built around a keyed algorithm EK (which is AES in the specification). This is the "core" of Milenage. Now, it seems possible to replace this core by another (secure) algorithm, for instance the SIM A3/A8 algorithms. Therefore, a possible solution is to re-use A3/A8 as an instantiation of the EK algorithm, performing all other operations (rotations, XORs, etc) in software in the ME.

There are, however, two problems with this approach. First, it would require at least six "subroutine" calls to the SIM. Since the SIM according to standard is allowed to have response times up to 500ms, this procedure may be too slow. Also, while the input to A3/A8 is, just as AES, 128 bits, the output of A3/A8 is only 96 bits. Hence some "padding" or post processing of A3/A8 outputs would be needed to get 128 bit values in some places where needed (e.g. in the first intermediate value and the CK output).

With this in mind, the following approach may be more attractive. On input RAND, run it through the SIM (A3/A8) to get a 96 bit output

$$K' = Kc \,||\, RES.$$

This K' is now padded/expanded (in some way, to be determined) into 128 bits, denote the padded value K. Next, K is used as a key for EK (AES), and the whole Milenage algorithm, using EK as "core" is run entirely in software in the terminal. This approach has the advantage of requiring only one call to the SIM and to limit the need for padding (key expansion) to only one place.

We again stress that we only use Milenage as an example, in general we would implement f1-f5 in a similar way, making the assumption that f1-f5 are built around a core crypto-algorithm EK, simulated by the SIM.

3 Considerations

3.1 Security

What is the risk of running part of UMTS AKA in software. As we see it, the main risk is that of a "spy-ware" (virus, Trojan, etc) having access to input/output pairs of the A3/A8 algorithm. Unless a "bad" A3/A8 algorithm is used this should, however, not reveal information about the subscriber key, Ki. Conversely, in case of a bad A3/A8 algorithm, this threat is anyway present since hypothetically, a "spy-ware" could feed the SIM with RAND values and pull the responses without the user noticing. Hence, we feel that the proposal does not increase security risks that are not present anyway, though this should of course be studied further. Another aspect is that the effective key used in AES may (as seen above) be only 96 bits according to one proposed implementation. This means that the ciphering key can be attacked with complexity 2^{96}, which would mean degradation for a 128-bit GSM A5/3 use, but not for a 64-bit A5/1 use. Still, a 2^{96} complexity is not very feasible.

3.2 Replay protection

Some way to handle SQN (in software too?) needs to be looked into, and what synchronization problems might occur if the SIM is moved between MEs if replay protection is desired. One way is to use an "opportunistic" approach in which case each movement of the SIM between (upgraded) terminals causes a re-synch. This means that the initial authentication after each movement is vulnerable to a replay attack, but not consecutive authentications.

An option may be to use the SIM (e.g. SIM application toolkit) to handle/store SQN.

3.3 Provisioning

It seems possible for the operator to "enable" the enhanced GSM AKA as above, e.g. by OTA provisioning, SIM application toolkit etc, enabling the software in the ME. On request by the subscriber for "enhanced security", the ME is unlocked/provisioned to provide the new AKA functionality, and a flag is set in the AuC/HLR so that the correct AKA algorithms is selected when the user in the future authenticates. That is, in this simple solution, the SIM is always assumed to reside in a "default" terminal. Problems to be looked into also here is what happens if a SIM is moved

between upgraded/non-upgraded terminals. What practical relevance this problem with “plastic roaming” has, and how it can be solved, is ffs.

3.4 Access networks

As we understand, the proposed solution is transparent to R99+ (or newer) networks. This may need further study.

3.5 Combination with other A5/2 attack solutions

The two most supported ways to mitigate the A5/2 attack so far has been the so-called “special RAND” and the “authenticated cipher mode command”. Some questions have been raised against these solutions. For “special RAND”, a concern is that randomness is stolen from RAND. For the second solution, an issue has been what key to use for the integrity. We note that the introduction of UMTS AKA can potentially solve both these issues if used in conjunction.

- A special RAND solution could be implemented by using the AMF field to signal which algorithms that are allowed, without reducing the randomness in RAND. In order not to consume all of AMF, an option might be to let one bit of AMF signal the presence of a special RAND, and carry the rest of the algorithm signaling in RAND. This removes the need for a 32.bit “special prefix” in RAND.

- The introduction of UMTS AKA gives, in a natural way, a key to be used for any possible integrity protection, namely IK. A difference here, compared to S3-040262 is that here the home network needs to know ME capabilities, rather than the visited network only. However, this is needed anyway for our proposal to work, or one needs to make default assumption on which ME the SIM is used in.

4 Conclusions

We have proposed a way to upgrade GSM AKA to UMTS AKA by a combination of software upgrade and re-use of existing SIM infrastructure. We believe this is a promising approach that deserves further study,