3GPP TSG SA WG3 Security — SA3#33 **S3-040259**
10-14 May 2004
Beijing, China

Source: **Ericsson**

Title: **Key deletion in the UICC model using S3-04235**

Document for: **Discussion and decision**

Agenda Item: **MBMS**

1 Introduction

To exclude a member of a MBMS group of ME:s, there are basically two options; either all ME:s except the one to be excluded are given a new MSK, or the MSK currently in use is deleted from the ME to be excluded. In S3-040232 [1] it is suggested that key deletion is only possible in the OTA model. This document discusses some problems with key deletion and how it can be solved in the combined model. Note that we are only considering deletion of the MSK, since there is no apparent benefit in the ability to remove MTK:s.

2 Usability of the Delete Function

There are mainly two arguments for having a delete function in MBMS. One is to remove ME:s that no longer whishes to be part of the service (as mentioned in the introduction). Since it is cumbersome to re-key all terminals that should remain in the service, the idea is to instead remove the ones that should no longer be part of the service. It is questionable if this is any simpler. It all depends on how much “mobility” there is in the set of members. The key delete function is only usable as long as there are very few members leaving the service before the MSK they have has expired. This of course also depends on the lifetime of the MSK; there have been many suggestions for the lifetime ranging from hours to months. It could be argued that once a user has received (and paid for) and MSK, he should not be able to “release” the contract and get a refund. This would also make charging much more simple. If the lifetime of the MSK is short enough and the price for an MSK is reasonable, this should be a viable solution and there is no need for a delete function.

The second argument for having a delete function is that a compromised MSK should be deleted. This makes no sense, since once it is compromised the only remedy is to distribute a new MSK, i.e., the compromised MSK should not be used anywhere.

The key deletion mechanism is used as an argument for the OTA model in MBMS in S3-040232, without specifying why it is a benefit. Furthermore, S3-040243 [2]specify the functionality of the delete mechanism, but does not really argue why this solution is preferable.

Note that in the case only a pull mechanism is used, the BM-SC would simply not give the new MSK to an ME that no longer is a part of the group. This makes the key delete mechanism useless in the pull only case.

3 Traffic Analysis and Key Identities

It seems as the MSK would be updated in the MBMS system as seldom as once a day or even more rarely. This means that the ME either pulls a new MSK from the BM-SC or that the BM-SC pushes a new MSK to the ME at regular points in time.

The only way to be sure the key deletion message is passed from the ME to the UICC would be to make the ME unaware of the content of the message received by from the BM-SC, and that the ME can not deduce that the message is a delete message purely on the time it was received. If the ME receives a message from the BM-SC at a point in time that is different from the expected “regular” MSK-update, it can be assumed to be a delete-message, and a malicious ME would not forward this to the UICC.

Hence, it must not be possible for the ME to detect if a given message contains a MSK or an MTK, and key deletion messages must be delivered at a time when an MSK or MTK update message can be expected.

There is however another approach that can be used to circumvent the above problems. If the BM-SC requires a cryptographically verifiable response from the ME, the user would be charged for the entire lifetime of the MSK if no such response is received by the BM-SC (see Section 4).

4 A Sketch of a solution

Assuming the entire MIKEY message sent from the BM-SC is passed to the UICC for verification of the integrity. The ME must send a response back to the BM-SC which is integrity protected by the UICC. That is, the ME constructs a response message, passes this to the UICC, which computes the MAC using a key derived from the MUK (known only to the UICC). Then the UICC returns the MAC value, which the ME inserts in the MIKEY response, and send the message to the BM-SC.

A malicious ME that refuses to send the delete message to the UICC, can not create a valid response message to send to the BM-SC, and hence the BM-SC will charge the ME for the entire lifetime of the MSK.

5 Conclusion and Proposal

Based on the analysis in this paper, it is questionable if a key deletion function is needed. We propose that SA3 decides on whether a key deletion function is needed in MBMS or not. If SA3 finds a key deletion function useful, this paper shows that feasible implementations are possible in the combined model.

6 References

[1] Telecom Italia, S3-040232, MBMS key management: *OTA*- versus *GBA-based* Point-to-Point key distribution

[2] Axalto, Gemplus, Oberthur, S3-040243, MBMS UICC-based solution

[3] Ericsson, Nokia, S3-040235, Extension payloads to MIKEY to support MBMS