3GPP TSG SA WG3 Security — S3#32 **S3-040107**
09 - 13 February 2004
Edinburgh, Scotland, UK

Source: **Gemplus, Oberthur CS.**
Title: Discussion paper on MBMS key compromising and fraud recovery

Document for: **Discussion**
Agenda Item: **6.20**

1 Introduction

At several previous meetings, questions have been raised about the security of a broadcast scheme where the system keys are stored in a UICC: what happens if one of the UICC is compromised and the keys end up in the possession of a pirate. This contribution aims at analysing the reality of the threat and proposes strategies to recover from an attack.

2 Threat analysis

This section summarizes some concerns derived from the usage of a key management procedure whose security relies solely on a secure key repository. The basic assumption in 3GPP is that the UICC is a tamper-resistant key repository, thus alleviating these concerns… however, with enough time and money, everything is possible, from corrupting a disgruntled member of the operator's staff to breaking a badly realized smart-card
So what happens if the secrets are out and in the hands of a pirate?

In any 2-tiered broadcast security model, all the users share at a given time two sets of keys: the short lived keys (SK) that are used to actually protect the broadcasted data and the higher level keys (BAK) that are used to produce such short lived keys.
SK keys are usually associated directly with the data and the decision as to whether one can access this data or not is reduced to whether the user has the corresponding BAK key or not.
The whole security scheme is based on the timely availability of these BAK keys, as the SK keys are deemed to short lived to present a viable means of attacking the system. This means that to set up a profitable piracy service, a pirate provider has to be able to distribute, preferably in advance, the BAK keys that will be used for the broadcast. This is possible since the BAK keys are updated well in advance to provide a seamless transition.
In a broadcast scheme, there is no way to recognize which user is accessing the service because, by essence, all of them get the broadcasted signal. So a pirate user only has to (1) access the broadcasted signal and (2) know the BAK keys to be able to access the data.

The conclusion is that everything hangs on the security of the BAK keys, and this is inherent to the 2-tiered model, wherever the keys are stored.
In the 3GPP approach, the BAK keys are distributed in a point-to-point manner. If a user has access to the BAK keys, either during the transmission or while in storage, he can re-distribute them and thus compromise the whole system. Furthermore, since the BAK keys are the same for everybody, there is no way to know where they come from.
Note that knowing the identity of the traitor, it is very easy to end the threat by simply not sending the new BAK keys to this particular user, since it is done point-to-point.

3 Fraud recovery (i.e. tracing the traitor)

Suppose a pirate has found a way to access the BAK keys and distributes them to his "clients". Our goal is to find the user identity corresponding to the leakage of the keys, so that it is possible to block this account and thus effectively stop the leak.
Foundation remarks:

- the pirate provides a return channel (the only one!) by sending the BAK keys to the pirate users. Accessing this information (by subscribing to the pirate service for example), one has access to the return channel.
- since the keys are distributed point-to-point, it is possible to know exactly which user knows which key.

A first approach is to distribute different keys to different sets of users for a same service. For the user terminal to be able to decrypt the data, the corresponding SK keys have to be sent in such a manner that each user gets the same SK for the data decryption. The system has to allow the same content to be decrypted with different BAK keys, which means that for different SK_Rand's, we can extract the same SK. This is easy to do provided that the algorithm to derive the SK keys from SK_Rand and BAK is reversible. In this case, given a SK, it is easy for the service provider to compute different SK_Rand for each BAK.
Another way to do is to duplicate the data for each set, so that it works with the 3GGP2 system as it is currently defined. That means a bigger load on the point to multipoint, trading 'multiple SK_Rand for one content' versus multiple 'SK_Rand and content', all else being the same.
In both cases, by dividing the set of all users in two groups and doubling the amount of data or SK keys broadcasted, one can isolate the set that contains the pirate by checking which BAK has been distributed to the pirate's "clients". From there on, by dichotomy, it is clear that the traitor can be find quite rapidly and efficiently, in $\log_2$ (number of users) updates of BAK keys.
One can generalise that to separating the subscriber in N sets and updating the BAK keys of each set with a different BAK. On publication of the leaked BAK, it is easy to find the originator set. Doing a new BAK update on this set only (redistributing the already issued BAK keys), the new BAK published will identify the new subset containing the pirate subscriber. The scheme needs $\log_N$(#subscribers) steps to identify the subscriber. The number of point-to-point updates is Sum [x=0…$\log_N$(#subscribers)] (#subscribers)/N^x.
Example: 1 million subscribers to MBMS and 2 sets. Then the pirate subscriber is found in 20 steps, needing about 2 million point-to-point update, equivalent to 2 full updates. With N = 10, it works in 6 steps and requires about 1.11 million updates, 11% more than one full update.

Another approach is to trick the traitor in sending out the BAK key before they are used (which he should do to provide a good service), then replace the BAK with a new one. In this case, each user will be provided with a unique BAK key, dully registered, and the BAK that is sent by the pirate uniquely identifies him. It is then just matter of updating all the users but the pirate with a new BAK key. One would have to update also all the BAK keys corresponding to the services this particular user has subscribed to. The cost of this countermeasure is only two updates of BAK keys for all users.

4 Conclusion

In the 2-tiered model chosen for MBMS, we have shown that there are simple ways to find a traitor that is leaking the BAK keys and to exclude him from the system, effectively blocking large-scale piracy. The cost in number of BAK upgrades is for each scenario two-times the number of users, in one case also requiring the doubling of the data/SK traffic.