3GPP TSG SA WG3 Security — S3#35 **S3-040796**
5 - 8 October 2004
Malta

Source: **Ericsson**

Title: **The need for and use of salt in MBMS streaming**

Document for: **Discussion and decision**

Agenda Item: **MBMS**

1 Introduction

This paper discusses the use of salt as a countermeasure to pre-computation attacks against the MBMS streaming system.

2 The need for salt

A salt is some random bits that are incorporated in the encryption process, and cannot be predicted before the encryption process begins. The salt does not need to be secret; the important thing is that it is unknown before the encryption begins.

The purpose of the salt is to protect against key collision attacks, which basically works as follows:

- The attacker guesses (or knows) some portion of plaintext that will later be encrypted. This is reasonable to assume, most protocols include fields that are known or easily guessable [1].
- He then encrypts this plaintext under a number of distinct keys, and stores the (cipher-text, key)-tuple in a database.
- The attacker records encrypted traffic from the streaming source (which changes the encryption key from time to time) and looks for a cipher-text that collides with one he has in his database.
- If a collision is found, he can look up which key he used to produce that cipher text.

Of course there may be one or more cipher texts in the database that matches, but the number is small enough that a brute force search is feasible.

More advanced key collision attacks do not require known plain-text, but only linear relations between bits in the plain-text. This *will* be present in the MBMS streams since we are to encrypt codec data etc., and especially if normal FEC is used.

The effective key length is reduced from n bits down to $n - log(M)$, where M is the number of distinct keys the attacker has in his database.

Two things can be noted. First, the attack works against *any* stream cipher, and secondly, the more frequent the re-keyings are, the faster the attacker will find a collision.

The salt, which is used to “extend” the key length, prevents the attacker from performing the pre-computation step without also guessing the salt.

3 How the salt is used in the encryption

As is stated in RFC3711 [2], the salt is incorporated in the encryption process by xor:ing it into the IV. In other words, the mechanism for using the salt is already present in SRTP, and all interfaces are in place.

Note that this use of salt relaxes the assumptions on the block cipher used. Since the salt (which is randomly generated) is xor:ed into the IV, the block cipher only has to be assumed to be good for random IVs, whereas it would have to be good for *all* fixed IVs. The word “good” is to be read in the sense that the output of the block cipher looks random.

4 Transportation of salt

The salt is sent from the BM-SC in the KEMAC payload (which also contains the key) to the UE. It is already specified in MIKEY [3] how the salt is incorporated in the KEMAC payload. MIKEY also provides the possibility to derive the salt from the key in the message, but to provide enough entropy, the delivered MTK would have to be longer if this technique is used.

5 Conclusion and proposal

The use of salt is required in MBMS to not shorten the effective key length, and all mechanisms to use it are already in place in the protocols used. We propose that the accompanying pseudo CR is implemented.

6 References

[1] McGrew and Fluhrer, “Attacks on Additive Encryption of Redundant Plaintext and Implications on Internet Security”, http://www.mindspring.com/~dmcgrew/dam.htm
[2] Baugher et. al., “The Secure Real-time Transport Protocol (SRTP)”, RFC3711, IETF
[3] Arkko et. al., ”Multmedia Internet KEYing (MIKEY)”, RFC3830, IETF