**3GPP TSG SA WG3 Security — S3#34** **S3-040542**
**6 - 9 July 2004**
**Acapulco, Mexico**

**Source:** **Ericsson**

**Title:** **Combined vs Separate Key Delivery**

**Document for:** **Discussion and decision**

**Agenda Item:** **MBMS**

# 1 Introduction

This paper discusses the two approaches of using separate delivery of encryption and integrity keys in contrast to deriving the two keys from one master key. Under some circumstances it may be necessary to also use a salting key with stream ciphers [5]. In this case we need to take this salting key into account as well.

# 2 Separate Key Delivery

It has been proposed in off-line discussions to deliver separate keys for integrity and confidentiality protection. The arguments for this approach has mainly been efficiency concerns in the terminals. Other arguments has dealt with security, but these are weaker compared to the efficiency related ones. The arguments for the use of separate delivery brought forward so far is listed and commented below.

- **The MSK/MTK key generation can be implementation specific; all implementations are not broken if one of the random number generators is broken. It is possible to use hardware based random number generators in the BM-SC.** By having different implementations of random number generators, we put the burden upon the designer to come up with a good generator. Since the competence differs among implementers, the quality of the generators will differ, and an overall high level of security cannot be guaranteed. It is true that hardware based generators have the potential of generating more "random-looking" bits by including entropy from physical measurements. However, a well-designed pseudo-random number generator can generate cryptographically independent bits. The difference lies in the underlying assumptions; for the hardware based generator the assumption is that it generates independent bits, and for the pseudo-random generator it assumption is that some mathematical problem is hard to solve. It should be noted that pseudo-random generators are commonly used for key derivation in 3GPP today (e.g., Milenage).

- **The number of algorithms and the complexity of implementation should be minimized in the terminal.** Since MIKEY [1] was agreed as key management protocol for MBMS at SA3#33, the key derivation function is already specified in MIKEY so there is no need to

re-invent the wheel again. Furthermore, the key derivation function in MIKEY has been scrutinized by IETF for several years.

- **It is easier to update the key generation function if it is only present in a few servers, compared to if it is present in all terminals.** This is true, but then again the key generators used in SSL [2], IKE [3] etc., have been stable for many years without the need for any updates.

Although none of these arguments can be considered as strong, there still may be useful to provide separate delivery. If for instance, the points where confidentiality protection and integrity protection is applied are different (and it is difficult to synchronize the two).

## Pros:

- There is possibly a small efficiency gain in processing on the terminal side. Instead of running a key derivation algorithm (usually consisting of a few applications of a hash-function) one is faced with having to run MGV-F two additional times (consisting of two hashes and a decryption each). The reason for having to run MGV-F three times, is that each of the integrity, confidentiality and salting keys would have to be decrypted and authenticated. Even though efficiency is important, the additional run of AES a couple of times is not likely to contribute considerably to the overall cost.
- Fits the agreed download model; MIKEY can be used to transport several MSK:s which are then used directly in the protection mechanism in the UE.

## Cons:

- More complex message structure in the key delivery.
- Longer messages (although marginally).
- MGV-F has to be run two times more than for combined delivery (see Pros).

# 3 Combined Key Delivery

Another approach to provide the terminal with integrity and confidentiality keys, is to send a seed to the terminal, who then derives the two keys from this seed using a key derivation function as implicitly proposed in S3-040258 [4]. As mentioned in Section 2, key derivation is an old and commonly accepted technique to produce several keys from one seed. The main benefits of combined delivery are:

## Pros:

- Fits the agreed download model; MIKEY can be used to transport one MSK which is then split into separate integrity, confidentiality and salting keys.
- The key derivation function defined in MIKEY is already in place.
- The key derivation function has been under the eye of IETF for years
- This technique has already been satisfactorily applied in 3GPP and elsewhere.
- There will be fewer bits transmitted over the air.

## Cons:

- The Cons has already been mentioned in Section 2.

# 4 Conclusion

There should be only one ``master key´´ delivered to the terminal, and this key should then be split into as many keys as required to satisfy the security protocols.

# 5 References

[1] J. Arkko et.al., ``draft-ietf-msec-mikey-08.txt'', IETF draft, work in progress
[2] T. Dierks, `` The TLS Protocol'', RFC2246, IETF
[3] D. Harkins et. al., ``Internet Key Exchange'', RFC2409, IETF
[4] Ericsson, ``Extension Payloads to MIKEY to Support MBMS'', S3-040258
[5] D. McGrew and S. Fluhrer, ``Attacks on Additive Encryption of Redundant Plaintext and Implications on Internet Security'', SAC 2000