3GPP TSG SA WG3 Security — S3#34 **S3-040557**
6 - 9 July 2004
Acapulco, Mexico

Source: **Ericsson**

Title: **MBMS Download Protection**

Document for: **Discussion and decision**

Agenda Item: **MBMS**

1 Introduction

This paper discusses the protection of downloaded objects in MBMS.

2 S/MIME

As agreed in SA3#33 S/MIME [2] was supposed to be the working assumption for protection of MBMS download. However, it seems that it does not suffice to use S/MIME alone since S/MIME will not be able to protect the integrity of the file delivery table (FDT) in FLUTE. This implies that if S/MIME is to be used, it probably should be combined with another mechanism that protects also the FDT. The protection of the FDT will be discussed in Section 4.

For the confidentiality part S/MIME can be used with a pre-shared secret. This pre-shared secret should be used to wrap the actual encryption key, which is carried in the S/MIME container. A new attribute specifying the MSK ID and MTK ID could be specified in the FDT XML-schema to allow the UE to retrieve the correct keys. When it comes to the integrity protection there are several possibilities.

2.1 Signature

To protect the integrity of the S/MIME container, the protocol provides the use of signatures. But the use of signatures implies that there has to be a public key infrastructure in place. Furthermore, public key operations consume more resources, both computational and bandwidth wise.

If the FDT is to be protected, it can be provided with a signature as well, e.g., the coverage of the S/MIME signature could be changed so that the FDT is covered as well.

2.2 Message authentication code

S/MIME does not include integrity protection of the container by symmetric key methods, so to have this functionality the protocol must be extended with a MAC.

- The obvious approach would be to compute HMAC/SHA-1 over the S/MIME container using the MTK_I. The MAC would then be appended to the S/MIME container. This approach is not specified in any other specification, so it would be a pure MBMS extension.

- Integrity protection of the S/MIME container can be achieved by, e.g., letting the HMAC described in Section 4 cover also the container. This can be done by setting the URI attribute of the Reference element in SignedInfo equal to the URI in the Content-Location attribute in the FDT.

3 XML encryption

An alternative to using S/MIME is to use XML-encryption [3]. If XML-signatures are chosen to protect the FDT and possibly also the downloaded MIME object, it seems natural to also use XML-encryption to confidentiality protect the MIME object. The use of S/MIME for the confidentiality part only is a bit of a waste, since the integrity protection is not used.

An example of the usage of XML encryption of the downloaded object is given below.

```
<?xml version='1.0'?>
        <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
          MimeType='text/plain'>
        <CipherData>
                <CipherValue>A23B45C56</CipherValue>
        </CipherData>
</EncryptedData>
```

This XML-document would be transferred instead of the file. The actual file is present in encrypted form inside the CipherValue-tag.

4 Protection of the FDT

An issue not previously discussed in SA3 is the protection of the FDT in FLUTE. The FDT does not really contain any sensitive information. If one has privacy concerns, it could be an idea to confidentiality protect the FDT (note that the MSK ID and the MTK ID must be left in the clear). The XML-encryption schema mentioned in Section 3 can be used to selectively encrypt everything but the key ID:s. However, since the data is broadcast (or multicast), it is difficult (although maybe not impossible) to pinpoint a particular user that downloads a particular file.

What is more of concern is the integrity of the FDT. This is because an attacker could insert a fraudulent FDT, and fool a user into downloading a file different from the one ordered (the attacker would have to also broadcast the fraudulent file).

Since the FDT is described as an XML schema, it is natural to use XML-signatures [1] to protect the FDT. It should be noted that signatures are used in a wider sense than normally in [1]; it also includes Message Authentication Codes (MACs), such as HMAC/SHA-1. That is, an XML-signature does not necessarily have to be based on public key cryptography. This allows usage of symmetric key cryptography to integrity protect the FDT in a standards adherent way.

The following attributes of XML-signatures are useful to MBMS::

- SignatureMethod/DigestMethod: HMAC/SHA-1.

- The KeyName element of the KeyInfo element should be set to “MSKID:xxx… MTKID:yyy…”, where yyy… is the ID of the MTK used as input to the MAC and xxx is the ID of the MSK used to protect the MTK with ID yyy….

The MTK is used to derive the integrity key MTK_I, which is used as input to HMAC/SHA-1. The details of the other attributes must also be specified.

5 Conclusion

The problem that is handled in this paper is that S/MIME does not provide integrity protection using symmetric keys, and that the public key based signatures that are provided do not cover the FDT.

Confidentiality protection of the download can be achieved through S/MIME or XML-encryption. As was mentioned in Section 2 there are ways to achieve integrity protection using S/MIME with public keys or integrity protection using symmetric keys if enhancements to the protocol are made.

There is also the possibility to integrity protect the download using XML-signatures.

It is noted that the FDT must also be considered to provide a complete protection of the download.

6 References

[1] Eastlake et al, “XML-Signature Syntax and Processing”, RFC 3275, IETF
[2] Housely, S/MIME, RFC 3369, IETF
[3] Eastlake et al, “XML Encryption Syntax and Processing”, W3C