3GPP TSG SA WG3 Security — S3#35 **S3-040781**

5 - 8 October 2004

St Paul's Bay, Malta

Title:	**Extensions to OMA DRM V2.0 DCF for MBMS Download Protection**
Source:	**Nokia**
Document for:	**Discussion/Decision**
Agenda Item:	**6.20**
Work Item:	

1. Introduction

Various mechanisms have been considered for Download protection in MBMS. For instance, use of S/MIME and XML has been discussed in [S3-040557]. In the joint SA3/SA4 MBMS Security meeting, we presented an alternative approach [tdoc7], based on OMA DRM V2.0 DRM Content Format, Discrete Media Profile (DCF) [OMA-DCF]. This paper provides the extensions needed for DCF to support MBMS Download protection.

2. Background

For downloaded content, OMA DRM V2.0 defines the DRM Content Format (DCF), which is specified in [OMA-DCF]. The DCF format, copied from the specification, is shown in Figure 1.

A DCF object is encrypted with a Content Encryption Key (CEK) using symmetric key mechanisms. The DRM agent at the terminal, after receiving a DCF object, is supposed to acquire the Rights Object (RO) from the RightsIssuer (specified in the RightsIssuerURL in the Common Headers). To use only the content format in MBMS, no RO is required. Therefore it needs to be specified how to indicate that the object is MBMS protected, in which case the DRM agent should not attempt to acquire the RO, but rather should use the corresponding MTK for decryption.

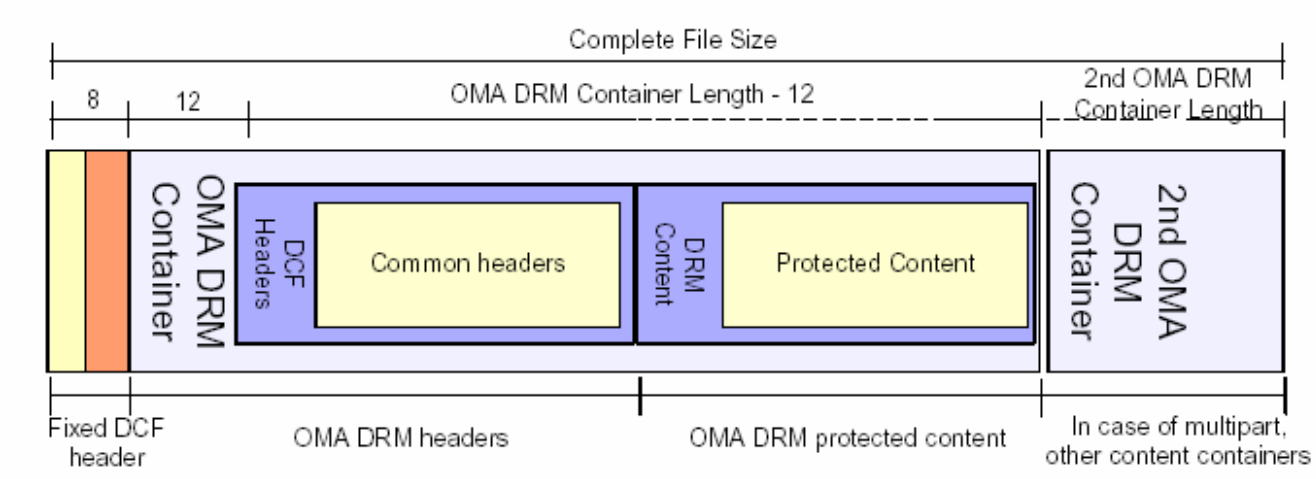

Figure 1 OMA DRM V2.0 DCF Format.

For integrity protection in DRM V2.0, a hash of the object is embedded in the RO. For MBMS, RO is not required. Integrity protection of the object can be provided by the XML-signature of the FDT as suggested in [S3-040557] or by including the object hash, calculated according to the OMA DRM V2.0 specification [OMA-DRM], in the FDT.

3. 3GPP MBMS Extensions to DCF

The proposal described in this section needs to be standardized as an extension to DRM V2.0 specification in OMA in order for MBMS to reuse the DCF format.

To distinguish an MBMS protected DCF content from ordinary DRM V2.0 DCF content, the proposal is to define a 3GPP MBMS flag in the Common Headers Box (See Section 5.2.1 of [DCF]). The Common Headers Box inherits the ISO FullBox type, which can be represented as follows:

Name	Type	Value
Size	unsigned int(32)	Offset to the end of the box
Type	unsigned int(32)	Box type 4CC
Version	unsigned int(8)	Version field
Flags	unsigned int(24)	Additional flags

The Flags field for the Common Headers Box is defined to be 0 in the current version. We propose to define the following 3GPP MBMS flag:

```
3GPP-MBMS-DCF = 0x000001 // or any other value assigned by OMA
```

Therefore, conforming file parser can distinguish between 3GPP MBMS download content and DRM V2.0 download content.

For MBMS download content, we further specify any needed new usage of the headers in the Common Headers Box:

Field	Type	Description	MBMS specific usage
EncryptionMethod	unsigned int(16)	Encryption Method	Same as DRM V2.0 DCF
EncryptionPadding	unsigned int(16)	Padding Type	Same
PlaintextLength	unsigned int(64)	Plaintext content length in bytes	Same
ContentIDLength	unsigned int(16)	Length of ContentID field in bytes	Same
RightsIssuerURLLength	unsigned int(16)	Rights Issuer URL field length in bytes	Same
TextualHeadersLength	unsigned int(16)	Length of the TextualHeaders array in bytes	Same
ContentID[]	char	Content ID string	MBMS specific Content ID
RightsIssuerURL[]	char	Rights Issuer URL string	Used to carry MBMS Key_ID
TextualHeaders[]	string	Additional headers as Name:Value pairs	Same

Usage of fields in the Common Headers Box does not deviate from the original DRM V2.0 DCF specification, except that the RightsIssuerURL field will be used to carry MBMS Key_ID information (See below). The ContentID field can be used by MBMS to convey MBMS specific content identification information.

As mentioned, RO is not used in MBMS. The Key_ID information includes the MSK_ID as well as other information necessary to derive the MTK (See Section 6.6.1 of [33.246]), which will be used to decrypt the content. To transport the Key_ID information, a new URL scheme has to be defined for MBMS, e.g. mbms-key. The RightsIssuerURL may then contains:

```
mbms-key://key_id
```

where `key_id` is defined as the base64 encoded Key_ID string. Conforming parser will be able to extract the MBMS Key_ID from the RightsIssuerURL header.

3.1. Overheads

The storage overhead added depends on the additional information added to the content (e.g. Title, Author, etc…). The minimum overhead (per content, if a single piece of content is embedded in the DCF file) is estimated to be around:

```
110 bytes + ContentIDLength + ContentTypeLength + Length(key-id)
```

The DCF also allows multiple pieces of content to be embedded within one file.

4. Processing at the Terminal

For a 3GPP-MBMS conforming DRM agent, the following pseudo-codes describes the processing logic when receiving a DCF content:

```
Perform normal parsing until Common Headers Box is reached;
If (Flags | 3GPP-MBMS-DCF) {
          // 3GPP MBMS download content
          Parse RightsIssuerURL to extract MBMS key_id;
          Return key_id to ME/UICC and request MTK;
          Decrypt content using MTK returned;
} else {
          // Normal OMA DRM V2.0 download content
          Perform normal DRM V2.0 operations; // e.g. Run ROAP to acquire
          RO
}
```

For a non-3GPP-MBMS conforming DRM agent that happens to receive a MBMS download content, it will probably report parsing error, due to the unknown flag in the Common Header Box, or an unknown URL type of *mbms-key* being used in RightsIssuerURL.

5. Conclusions

This paper proposes extension to OMA DRM V2.0 DCF data format for MBMS download protection. A flag is defined in the Common Headers Box to indicate MBMS download content, and the RightsIssuerURL is reused for carrying the MBMS Key ID information (by defining a new URL scheme for MBMS key_id). Apart from these, there are no other changes needed to the original DCF specification.

The use of DCF for MBMS download protection allows DRM agents in terminals to be reused for MBMS, thereby reducing terminal complexity. It should be noted that for terminals without DRM functionalities, MBMS download content could still be parsed with the correct parser implemented.

6. References

[S3-040557] MBMS Download Protection, S3#34, July 2004, Ericsson.

[Tdoc7] Previous contribution to the S3/S4 joint MBMS security meeting in Aug 04.

[DRM-DCF] DRM Content Format, OMA-DRM-DCF-v2_0-20040715-C, www.openmobilealliance.org.