USING OMA DRM 2.0 PROTECTED CONTENT
Ogg Vorbis Protected Audio under Symbian OS

Francisco Pimenta, Carlos Serrão
ADETTI, ISCTE/DCTI, Av. das Forças Armadas, Edifício ISCTE, Lisboa, Portugal

Keywords: OMA, DRM, Symbian OS, Ogg Vorbis, Security, Cryptography.

Abstract: The lack of control inherent to digital content has been put on the spotlight by copyright infringement coupled with massive content distribution online (e.g., Peer-to-Peer). Digital Rights Management seems to be the solution to counter this problem advocating the use of cryptography and other related security mechanisms to protect digital content and to associate rights with it which determine how, when and by whom it can be consumed. The Open Mobile Alliance (OMA) specifies mobile service enablers in order to ensure interoperability throughout the mobile spectrum. As prominent mobile devices, Symbian OS smartphones offer an interesting platform for the demonstration of OMA DRM for the consumption of multimedia content. This article outlines the mechanisms enabling the protected consumption of the open and patent-free audio format (Ogg Vorbis Website), Ogg Vorbis using an OMA DRM 2.0 compliant audio player application running under Symbian OS (directed for mobile devices).

1 INTRODUCTION

The growing proliferation of multimedia digital content throughout, virtually, every digital platform and system available today has produced a profound impact.

Digital Rights Management (DRM) solutions aim to enable content providers to assign and oversee usage permissions or *rights* for multimedia content upon purchase/distribution with the aid of cryptographic mechanisms.

The Open Mobile Alliance (OMA) is an open organization dedicated to specifying mobile service enablers. In line with the mobile driven nature of OMA, OMA DRM is especially oriented for mobile environments. The specification defines the Rights Object Acquisition Protocol (ROAP) (OMA DRM Specification, 2005), the DRM Content Format (DCF) (DRM Content Format, 2005), the Rights Expression Language (REL) (DRM Rights Expression Language, 2005) and more. This version of the specification provides added security to the previous published release (1.0).

Symbian Operating System (OS) mobile phones are a compelling deployment platform for OMA DRM 2.0 as their usage and multimedia capabilities are on the rise.

2 OMA DRM 2.0 OVERVIEW

The main focus for this document will be the OMA DRM Agent entity and in particular it’s content and rights interpretation and consumption capabilities.

The DRM Agent, which is a trusted software component functioning under (in this case) a mobile device oversees all of the functionalities related to the interpretation and enforcement of the usage defined by the OMA DRM 2.0 specification and rights/content issuers.

2.1 DRM Content Format

Prior to distribution, content issuers are responsible for packaging unprotected multimedia content in a secure container known as DCF (DRM Content Format).

Pimenta F. and Serrão C. (2006).
USING OMA DRM 2.0 PROTECTED CONTENT - Ogg Vorbis Protected Audio under Symbian OS.
In *Proceedings of the International Conference on Security and Cryptography*, pages 311-315
DOI: 10.5220/0002103003110315

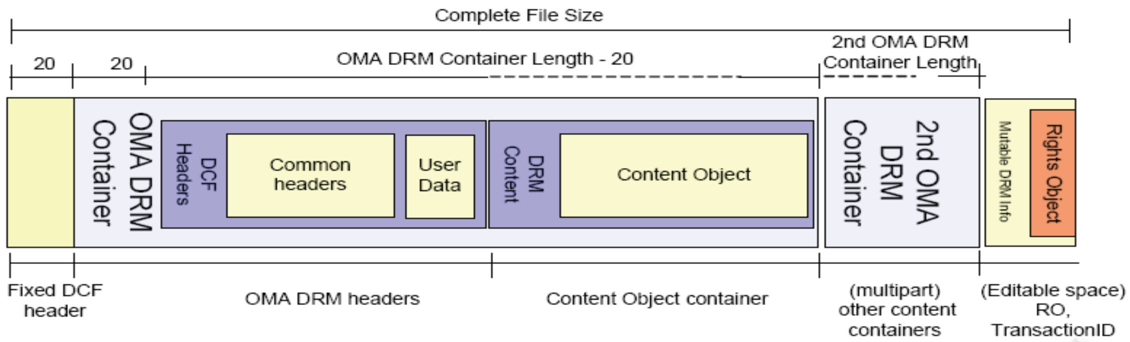


Figure 1: DCF Structure (Information technology, 2005).

Version 2.0 of the DCF offers more extensive information fields for the protected content and a well defined file structure in comparison to version 1.0. It uses object-oriented structure as defined the ISO Base Media File Format (ISOBMFF) specification (Information technology, 2005). The ISOBMFF uses an object-oriented design of *Boxes* each containing mandatory *type* and *size* fields. No data is to be present outside a *box* structure.

DCF employs two distinct profiles with the aim of suiting the particularities different media types. The standard profile (Figure 1) was designed for Discrete Media (i.e. ring tones, applications, images, etc.) protection by nesting the complete encrypted content within the *Box* hierarchy regardless of the actual original content type (audio and video content is supported as well).

Despite the flexibility of the base profile, OMA has specified a second profile with audio and video content in mind deemed Packetized DCF (PDCF). Both profiles use the box class concept defined in the ISOBMFF but the second profile fully implements the ISOBMFF. The motivation behind the PDCF is the fact that audio and video is composed with packets of data.

PDCF deployment is particularly smooth if the original content is in the ISOBMFF such as is the case with 3GPP, MPEG-4 or Apple's QuickTime (QT) (MPEG-4 File Formats white paper, 2005). However, using different types of audio and video file formats as PDCFs implies a conversion effort which may have significant costs and barriers as is for the presented work (clarifications in section 3).

Counter mode (CTR) is supported in this version avoiding padding bits.

2.2 Rights and the Rights Expression Language

In OMA DRM, *Rights* are self contained objects, separate from content and independent from the content type. These objects use the REL for defining their syntax and semantics which in turn is based on the Open Digital Rights Language (ODRL) (DRM Rights Expression Language, 2005). REL XML elements are grouped according to their functionality in different models, namely the: Foundation (root/basis), Agreement (main container), Context (metadata), Permission (i.e. type of permission to: play, display, etc.), Constraint (e.g. time or count based constraints), Inheritance (inheriting *rights* from a parent RO) and Security models (message digest and encryption data).

2.3 Rights Object Acquisition Protocol

The ROAP protocol suite (OMA DRM Specification, 2005) is used to interact directly with RIs creating the means for acquiring ROs from RIs but, it is also composed by other sub-protocols that accomplish other mechanisms, such as, joining/leaving a domain.

There are two ways of initiating ROAP: 1) through a ROAP Trigger (OMA DRM Specification, 2005) or 2) from a DCF. Initiating ROAP from a DCF object partakes to the analysis of the *Box* structure inside, particularly the *Common Headers Box* which should contain the RI URL.

Upon first contact, and sub sequentially whenever deemed necessary, the Agent and RI go through a handshaking process using the Registration Protocol. Successful registration results in the storage of RI context information. After context storage, mutual authentication is made possible on the other ROAP protocol runs.

A successful RO Acquisition protocol run wields a valid RO in the RO Response message.

3 THE OMA-DRM 2.0 IMPLEMENTATION USING OGG-VORBIS

This chapter describes the work done thus far and outlines the future work using the Symbian Environment and an open source Symbian audio player with no initial internal DRM support whatsoever. The goal is then to add OMA DRM 2.0 support to the player and, by extension, the phone in which the player is installed in. This task can be decomposed in: adding support for interpreting DCFs (2.1), ROs (2.2) and implementing ROAP (2.3).

The audio player chosen as a work base is an Open Source project for Symbian Series 60 focusing primarily on the Ogg Vorbis format.

The chosen Symbian User Interface (UI) deployment platform and testbed was Series 60 developed by Nokia which is to date, the most popular UI– higher number of phone models. PC based testing used S60 2nd Edition FP3 Emulator for Symbian v8.1.

3.1 Ogg Vorbis Audio

Vorbis and Ogg are, respectively, an audio codec (Vorbis I specification) and generic encapsulation format (bitstream container) (Pfeiffer, 2003) together making up Ogg Vorbis Audio.

In greater detail, Vorbis is a perceptual audio codec (lossy) similarly to MPEG-4 ACC, AC-3 or RealAudio. A Vorbis bitstream is composed by three initial header packets: identification, comment and setup headers in that order which are necessary in order to initialize the bitstream (Vorbis I specification). After these headers all remaining packets are audio packets with type *0*, mode and block size fields.

The Ogg multimedia bitstream format is agnostic towards the actual media (*logical bitstreams*) it stores – it is media independent. Encapsulation is achieved by taking a packet-by-packet *logical bitstream* and dividing each (packet) into 255 byte chunks or *Ogg segments* (last segment may be smaller), following this groups of contiguous segments are wrapped into variable length (approximately 64Kbytes or 65307 bytes maximum) *Ogg pages*. Metadata is part of the second header packet in a Vorbis bitstream – *comment header* (Vorbis I specification).

A major advantage of adopting Ogg Vorbis is its patent, license free and open nature (Ogg Vorbis Website), another, is its high fidelity and transparency. Ogg Vorbis is estimated to reach transparency at 160 kbit/s whereas MP3 does it at about 192 kbit/s, thus, Ogg Vorbis produces a smaller file size to achieve the same quality.

3.2 DCF Decision

It is clear that that it is advisable that audio/video data should encapsulated into a PDCF. However, Ogg Vorbis is not a ISOBMFF. Vorbis is usually encapsulated onto the Ogg container format but it may also be encapsulated into other formats. With using the ISOBMFF a considerable effort would be required in coding all the data and metadata from the Vorbis bitstream into the new format. It is also a countersense to do this conversion since PDCF was designed so that files look and function like a media file to the outside (DRM Rights Expression Language, 2005). With this in mind, it is unclear what file extension such a file should have.

The solution found was using the DCF in spite of its shortcomings regarding packetized media. This way the Ogg Vorbis can be preserved and encapsulated into the DCF effortlessly. However, it is required to encrypt packets separately before adding the bitstream to the DCF.

Figure 2 marks the (audio) data regions of a Ogg Vorbis file subjected to encryption. *Ogg page* headers and non-audio Vorbis packets (headers) should be left in plaintext form so that the vorbis format can be kept readable. This also permits using the file's original metadata instead of using the *User Data Box* (figure 1). To detect that the DCF was encrypted in this manner it is sufficient to check the content type of the original content present on the

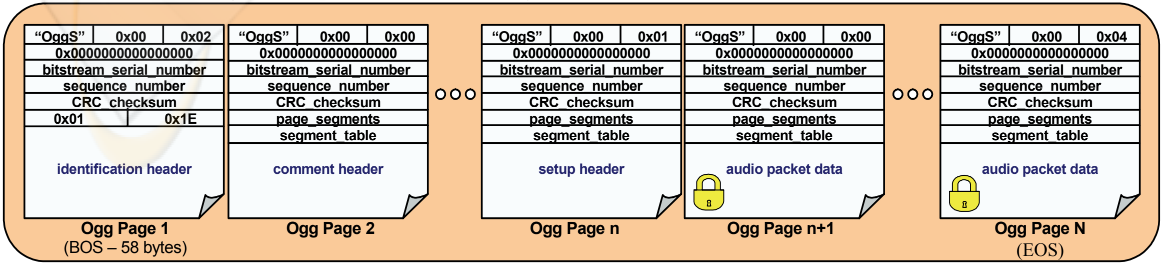
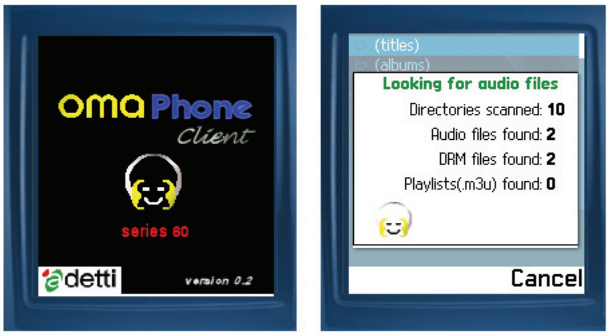


Figure 2: Typical structure of an (encrypted) Ogg Vorbis file.



Looking for audio files

Directories scanned: 10

Audio files found: 2

DRM files found: 2

Playlists(.m3u) found: 0

Figure 3: Application Splash Screen (Left) and File Search Dialog (Right).

Discrete Media Headers Box although the Ogg Vorbis within can also be identified by its *capture pattern* (=″`OggS`″).

3.3 Current Work

The current work has focused primarily on making the audio player support the DCF format.

The play looks for files that are found to be of known audio formats are added to player's local playlist. The player uses a Symbian Series 60 concept (from v7.0s+) which is the Multimedia Framework or MMF (see (Symbian OS SDK v8.1A, 2005)). This framework provides a Client-Controller architecture in which a client can make requests the controller framework to play, convert or record audio and more. On the controller side there are separate controller plugins responsible for playing different media types.

To support DCF (`.odf`) files it was necessary to add a new controller and subset of classes to handle of this format. So, when searching for files (figure 3 – right), each found occurrence launches basic checks on the file structure. In order to keep a reasonable file search speed (a couple of hundred milliseconds).

The most basic check determines if the *File Type Box* is present and matches the structure defined in (DRM Rights Expression Language, 2005) further more, it is necessary that all mandatory *Boxes* be present as well as all the necessary information to read and decrypt (if encrypted) the file (e.g. initialization vector). Any error reading *Box* fields also results in failure and consequent rejection of the file as valid DCF. Failure in any of these checks results in not adding the file to the list at all.

After a successful initial check the file may be played. The major difference to playing a regular `.ogg` file is not only the fact that cryptography is involved but also that the actual original file it kept within an offset range within the physical file. This is achieved with changes in the Ogg controller at the level of the interface code between the Symbian C++ code and the C code. The AES algorithm used is a Symbian C++ port of the optimized 3.0 C code in CTR mode.

3.4 Conclusions and Future Work

The following steps shall focus primarily on introducing the other two key concepts of the OMA DRM system, ROs and ROAP. A common need for these two elements is XML parsing and generating API. For this a porting, shall be done of a XML parser with desirable lightweight characteristics. XML functionality is a significant bulk of the work to be done in completing the system. As ROs are supported an important issue shall be *installing* them, particularly when they contain *stateful* rights. In which case state information should de saved in encrypted form for DRM Agent management.

ROAP support shall include the coding of the appropriate behaviour of the suit of protocols with inherent mechanisms concerning rights and security including the storage of RI contexts with eventual integrity protection using hashing. Enabling hashing capabilities for instance requires the porting of the SHA-1 algorithm to Symbian.

With the deployment of ROAP the playing of a DCF file shall contemplate: 1) initiating (silently or not) a ROAP Registration (if necessary) and a 2) RO Acquisition protocol which results in the return of the appropriate RO if successful to allow file playing.

REFERENCES

"OMA DRM Specification", Candidate Version 2.0 – 15 September 2005, Open Mobile Alliance, OMA-TS-DRM-DRM-V2_0-20050915-C.

"DRM Content Format", Candidate Version 2.0 – 01 September 2005, Open Mobile Alliance, OMA-TS-DRM-DCF-V2_0-20050901-C.

"DRM Rights Expression Language", Candidate Version 2.0 – 25 August 2005, Open Mobile Alliance, OMA-TS-DRM-REL-V2_0-20050825-C.

"Information technology — Coding of audio-visual objects – Part 12: ISO Base Media File Format", International Organisation for Standardisation, ISO/IEC 14496-12, Second Edition, April 2005.

"MPEG-4 File Formats white paper", International Organisation for Standardisation, ISO/IEC JTC 1/SC 29/WG 11N7609, October 2005, Nice.

Vorbis I specification, *http://www.xiph.org/vorbis/doc/Vorbis_I_spec.html*.

Pfeiffer, S., "The Ogg Encapsulation Format Version 0", RFC 3533, May 2003.

Symbian OS SDK v8.1A, Symbian Limited, 2005

"Ogg Vorbis Website", *http://www.vorbis.com/*