Protecting Digital Content Utilizing Standards

DRM Plugin Software Development Kit

Version 2.0.1
21 June 2006

Prepared By:
Mutable, Inc.
info@mutablemedia.com

Table of Contents

1 INTRODUCTION 3
2 OVERVIEW OF THE SDK 3
3 BUILDING, INSTALLING AND UNINSTALLING 4
4 NECESSARY STUFF 5
5 DEMO USAGE 8
6 API 14

info@mutablemedia.com

1 Introduction

DRM plugin software development kit provides a solution for protecting media content and presenting protected media content. It is a continuation of existing *OpenIPMP* project. The SDK consists of core plugins which provide DRM functionality and plugins written as a support for specific media content formats.
It is intended to be used in combination with media content encoding and decoding applications. Thus DRM protection can be easily added independently of specific application details.

2 Overview of the SDK

Core plugins contained in the DRM plugin SDK provide full DRM functionality:

- Creation of content encryption keys.
- Creation of rights associated with a user and a content.
- Saving encryption keys and associated rights on a DRM server, using a messaging system for communication with the DRM server.
- Creation of qualified content identifiers.

DRM protection can be summarized as DRM method and encryption. DRM method defines protection details, like encryption keys and rights handling. Encryption defines the actual symmetric key encryption of content data.

Core plugins provide the following DRM methods and encryption for protecting media content:

- ISMA with AES encryption (integer counter mode) with 128-bit key.
- *OpenIPMP* with AES encryption (counter mode) with 128-bit key and Blowfish encryption..
- OMA with AES encryption (counter or cipher block chaining mode) with 128-bit key.

Each DRM method uses a messaging system to communicate protection details to a DRM server. Communication is carried out via web services. The SDK implements two messaging systems:

- *OpenIPMP* messaging system (used by ISMA and *OpenIPMP* DRM).
- OMA messaging system as defined in OMA DRM documentation (used by OMA DRM).

info@mutablemedia.com

Besides the core DRM functionality, the SDK provides plugins for protecting specific content formats according to specific protection standards. Supported content formats and protection standards are:

- MPEG-4 format and protection standard using mpeg4 protection atoms as defined by ISO base media file format.
- MPEG-2 format and IPMP-X protection standard.

SDK also provides plugins for creating qualified content identifiers according to DOI specifications.

3 Building, Installing and Uninstalling

The SDK has been built, installed and tested on Windows and Linux platform.

On **Windows** platform, one can then build the SDK either with Microsoft Visual Studio 6 or Microsoft Visual Studio .NET. Follow these steps:

- Load src/DRMPlugin/DRMPluginAll/DRMPluginAll.dsw (Visual Studio 6 workspace) or src/DRMPlugin/DRMPluginAll/DRMPluginAll.sln (Visual Studio .NET solution).
- Build the DRMPluginAll project to build all the code for the SDK. Use Debug or Release configuration.
- After the build is completed, all libraries and headers are copied to src/DRMPlugin/DRMPluginAll/DRMPlugin directory. Headers are copied to include directory. DRMPluginAll.dsw copies all binaries to lib/win32/VC6/debug directory for Debug configuration and to lib/win32/VC6/release directory for Release configuration. DRMPluginAll.sln copies all binaries to lib/win32/VC7/debug directory for Debug configuration and to lib/win32/VC7/release directory for Release configuration.

Note that during testing some compilation problems in Microsoft VisualStudio v6 have been noticed when the project is located in a deep path structure (such as those associated with the “My Documents” or “Desktop” folders). Apparently, Microsoft VisualStudio v6 has some limitations on path lengths. If you notice that the compiler has difficulty locating header files that seem to be in the path, try moving the project folder closer the root of the drive to reduce include path length.

There is no uninstalling on Windows platform.

On **Linux** platform, one needs aclocal, autoconf, autke and libtool tools to build the SDK. Follow these steps:

info@mutablemedia.com

- Go to src/DRMPlugin directory.
- Add execution permission on autogen.sh script (chmod +x autogen.sh).
- Run ./autogen.sh to create the makefiles.
- Run make sdk_install (needs root privileges). This builds the code and copies the headers to /usr/local/include/DRMPlugin directory and the binaries to /usr/local/lib/DRMPlugin directory.

To uninstall on Linux platform, go to src/DRMPlugin directory and run make sdk_uninstall (needs root privileges).

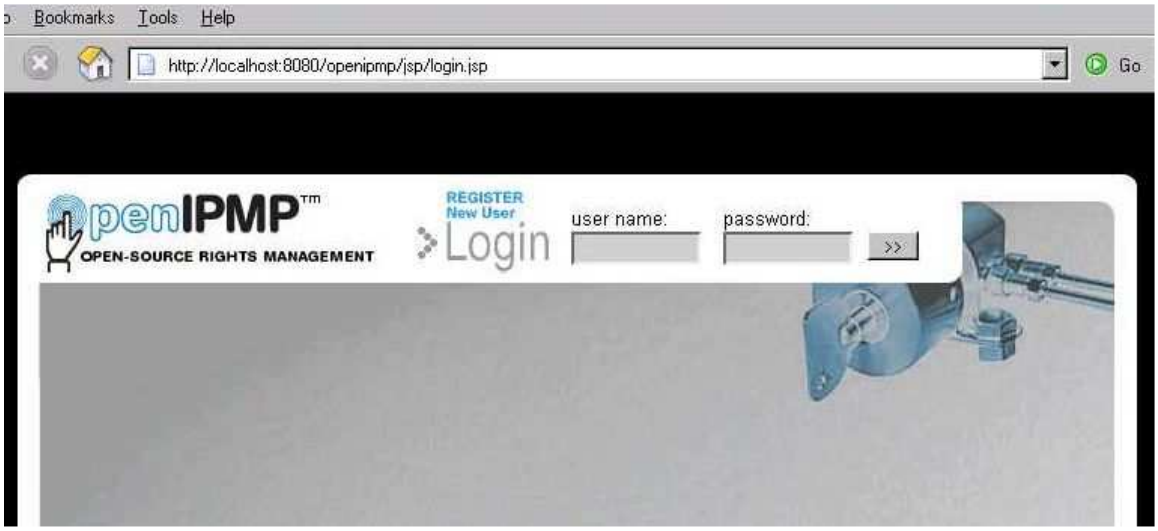
Please note that *OpenIPMP* has been tested and is dependent on the following Linux development toolsets:

- gcc 3.2 or greater
- libtool 1.5
- autoconf 2.57
- automake 1.7.6

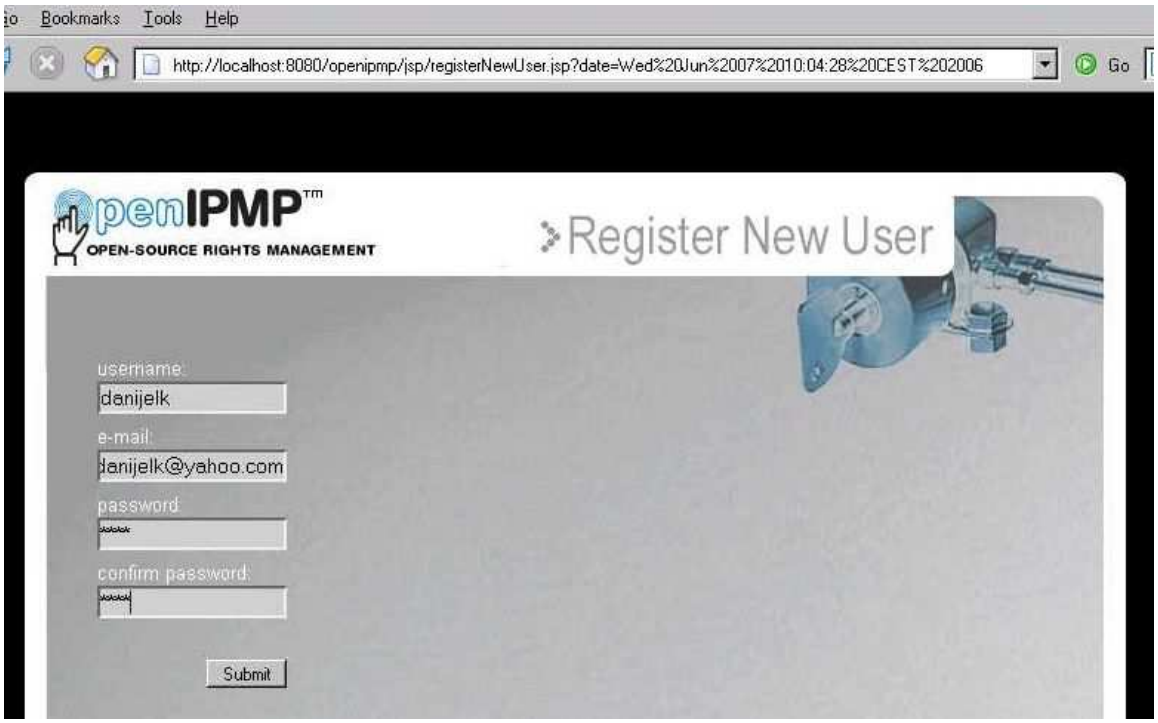
4 Necessary stuff

DRM plugin SDK communicates with DRM server, which requires user authentication to perform any action. User is authenticated with user name and password. Associated rights and protection keys are saved to user's .p12 file locally.

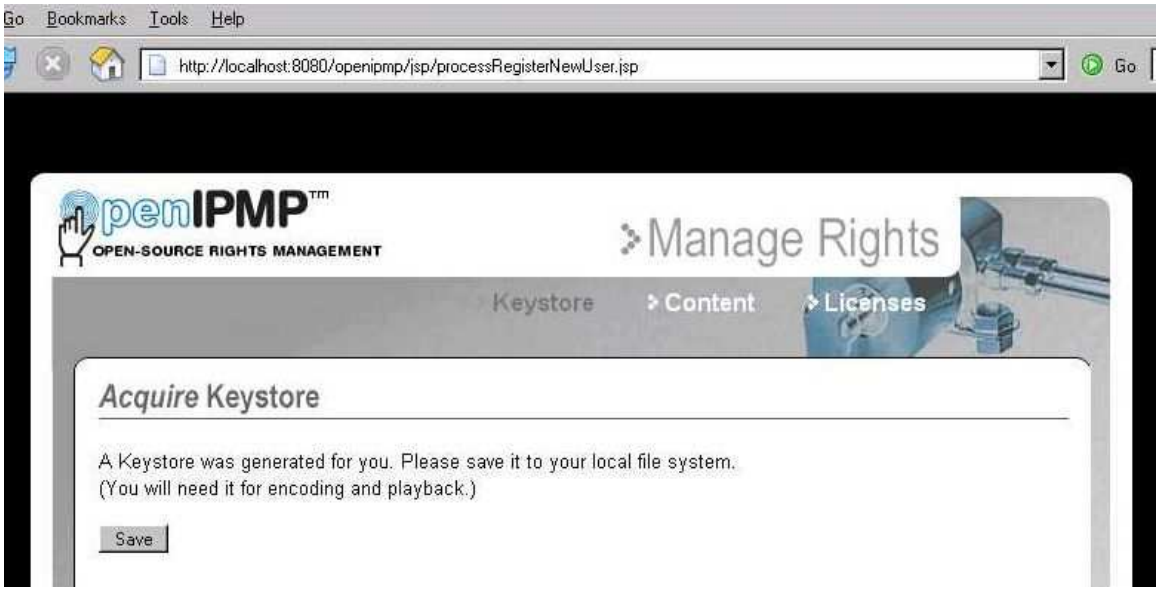
Under assumption that DRM server is installed on local machine and 8080 port, it can be accessed via http://localhost:8080/openipmp/jsp/login.jsp. If this is not the case, replace localhost and 8080 with IP address and port of the host where DRM server is installed.



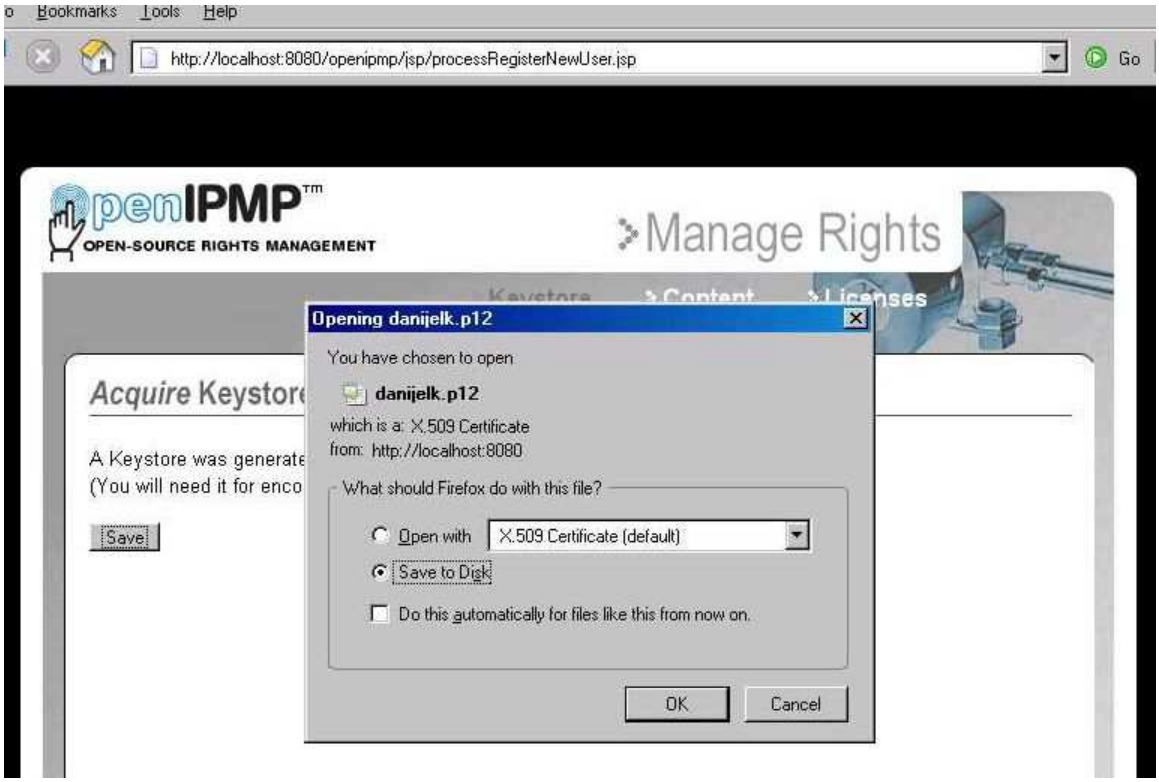
To create a new user, follow the “REGISTER New User” link and write in necessary data.



By clicking on the “Submit” button, new user with given information will be created.



Click on the "Save" button to save user's p12 file.



Registered user can login with user name and password. Following "Keystore" link, registered user can download p12 file for saving associated rights and protection keys.

info@mutablemedia.com

For OMA DRM, it is necessary to have device certificate file, CA certificate file and device private key file. Optionally, one can have trusted CA certificate files, which are used as a CA for server certificates. Certificate files are expected to contain DER encoded X509 certificates. Public keys in certificates are expected to be RSA with MD5 keys. Private key file is expected to contain a DER encoded unencrypted PKCS8 private key. Private key is expected to be RSA with MD5 key. For more information on generating such keys and certificates, please see the ***OpenIPMP* DRM Server Documentation** for more details.

If trusted CA certificates are used by a device, then the server must identify itself with a certificate chain which roots in any of the CA certificates trusted by the device. Otherwise, the server can identify itself with any certificate. The situation is symmetric on the server side. If the server uses trusted CA certificates, then device CA certificate, which signs the device certificate, must be one of the CA certificates trusted by the server. Otherwise, the device can send any certificate to identify itself.

For testing purposes, one can use provided deviceCertSigned.der (device certificate file), OMACACert.der (CA certificate file and trusted CA certificate file) and deviceKey.der (device private key file) from src/Demo/data directory.

5 Demo usage

The SDK has been integrated with the mpeg4ip project. The mpeg4ip project provides mpeg4 files encoding and decoding applications. Integration with DRM plugin SDK augments its functionality with DRM support as explained above.

To build and use mpeg4ip with DRM support, unpack src/Demo/mpeg4ip/mpeg4ip-1.5.rar to some directory.

On **Windows** platform, use Microsoft Visual Studio 6 workspace DRMPluginAll.dsw (see Building, Installing and Uninstalling section) to build the SDK. The build will create a directory containing headers and binaries under src\DRMPlugin\DRMPluginAll\DRMPlugin. Copy the DRMPlugin directory to mpeg4ip-1.5 root directory.

Here is a brief overview for Windows platform:

- Load mpeg4ip-1.5/encoding60.dsw in Microsoft Visual Studio 6 and build mp4creator application.
- Load mpeg4ip-1.5/player/src/player60.dsw in Microsoft Visual Studio 6 and build mp4player application.

info@mutablemedia.com

- Either use mp4creator and mp4player applications from Microsoft Visual Studio or copy them wherever you like together with all the dynamic link libraries from DRMPlugin directory and mpeg4ip plugin libraries.

On **Linux** platform, follow the instructions from Building, Installing and Uninstalling section to build and install the SDK.

One can now follow the mpeg4ip instructions on how to build and install it.

A brief overview for Linux platform:

- Go to mpeg4ip-1.5/SDL directory.
- Run chmod +x configure.
- Run ./configure, make and make install (needs root privileges).
- Go to mpeg4ip-1.5 directory.
- Run chmod +x cvs_bootstrap configure.
- Run ./cvs_bootstrap --disable-mp4live, make and make install" (needs root privileges).
- Export necessary environment variables: export LD_LIBRARY_PATH=/usr/local/lib:/usr/local/lib/mp4player_plugin:/usr/local/lib/DRMPlugin.
- Use mp4creator and mp4player from shell.

DRM Plugin SDK supports both creating protected files and decoding protected files. In mpeg4ip project, mp4creator application is used to encode content. Besides standard mpeg4ip parameters, it now takes certain parameters needed for DRM encryption.

- -E option is used to set encryption of all (audio and video) tracks in a file, and -E=<track ID> is used to set encryption of a track corresponding to track ID (identifier).

DRM options are the following:

- -W=<xml file name>, [mandatory], set XML configuration file name.
- -X=<sensitive data>, set sensitive information.
- -Y=<drm method>;<encryption method>, [mandatory], set protection methods.

XML configuration file contains necessary information for DRM encryption. Sensitive information is set via command line if it shouldn't appear in the XML configuration file (user name, password, licenses...). -X option can be used multiple times. <sensitive data> can be:

- UserName;<user name>, [mandatory, only one], user name for DRM server access.
- UserPass;<password>, [mandatory, only one], user password for DRM server access.
- License;<user name>;<start date>;<end date>;<license format>, [mandatory for ISMA and *OpenIPMP* DRM, can be more than one], where start and end date must be

info@mutablemedia.com

formatted as <YYYY>-<MM>-<DD>, and license format must be ODRL. License will be added for user identified by user name and for given time period.

- License;<device identifier>;<start date>;<end date>, [mandatory for OMA DRM, can be more than one], where start and end date must be formatted as <YYYY>-<MM>-<DD>T<HH><MM><SS>Z, and one can use OMADRMDeviceIDCalc application (from DRMPlugin binaries) to calculate the device identifier from DER encoded X509 certificate file. License will be added for device identified by device identifier and for given time period.

DRM and encryption method must be one of the following combinations: oma;cbc, oma;ctr, isma;icm, openipmp;bfs, openipmp;ctr. oma, isma and openipmp stand for OMA DRM, ISMA DRM and *OpenIPMP* DRM protection, respectively. cbc, ctr, icm stand for AES encryption with 128-bit key, and cipher block chaining, counter and integer counter modes, respectively. bfs stands for blowfish encryption.

Look at a sample XML configuration file for encryption (Windows paths must be changed to unix paths if encrypting on a Linux platform): doc/html/EncoderInfo.xml.

```
- <EncoderInfo>
    <!-- Rights host address -->
    <RightsHostURL>localhost:8080</RightsHostURL>
    <!-- OMA silent header -->
    <SilentHeader>Silent:on-demand;www.silent.com</SilentHeader>
    <!-- OMA preview header -->
    <PreviewHeader>Preview:instant;www.preview.com</PreviewHeader>
    <!-- OMA content URL header -->
    <ContentURLHeader>ContentURL:www.content.com</ContentURLHeader>
    <!-- OMA content version header -->
    <ContentVersionHeader>ContentVersion:original-content-identifier:1.1</ContentVersionHeader>
    <!-- OMA content location header -->
    <ContentLocationHeader>Content-Location:www.content.com</ContentLocationHeader>
    <!-- OpenIPMP p12 file path -->
    <P12FilePath>\p12\</P12FilePath>
    <!-- OpenIPMP random number file -->
    <RandomNumberFilePath>.\entropy.dat</RandomNumberFilePath>
    <!-- Content title, used by DOI content info manager -->
    <ContentTitle />
    <!-- DOI content information -->
  + <doi:KernelMetadata xmlns:doi="http://www.doi.org/">
  </EncoderInfo>
```

RightsHostURL tag in XML configuration file must be set to IP address of a host where DRM server is installed and working. P12FilePath tag in XML configuration file must be set to the path where p12 file for the user identified with UserName and UserPass (see above) resides. See the Necessary stuff section for p12 file details. RandomNumberFilePath tag in XML configuration file must be set to the path where entropy.dat will reside. It can be left as it is.

info@mutablemedia.com

For testing purposes, other tags in XML configuration file can be left as they are.

For all the following sample command lines, it is assumed that test30.mp4 is a plain mp4 file which will be protected and saved as enc-test30.mp4.

Note that for each of the following examples, the *OpenIPMP* DRM Server must be running.

Here is a sample command line for OMA encryption on Windows platform:

```
mp4creator -E -W=EncoderInfo.xml -X=UserName;danijelk -X=UserPass;ipmp
-X=License;dG2IhDioBnMdX1faxsuX0MCCmaQ=;2006-01-02T00:00:00Z;2006-12-31T00:00:00Z -
Y=oma;cbc test30.mp4 enc-test30.mp4.
```

Here is a sample command line for OMA encryption on Linux platform:

```
mp4creator -E -W="EncoderInfo.xml" -X="UserName;danijelk" -X="UserPass;ipmp"
-X="License;dG2IhDioBnMdX1faxsuX0MCCmaQ=;2006-01-02T00:00:00Z;2006-12-31T00:00:00Z" -
Y="oma;cbc" test30.mp4 enc-test30.mp4.
```

Here is a sample command line for ISMA encryption on Windows platform:

```
mp4creator -E -W=EncoderInfo.xml -X=UserName;danijelk -X=UserPass;ipmp
-X=License;danijelk;2005-08-31;2006-08-31;ODRL -Y=isma;icm test30.mp4
enc-test30.mp4.
```

Here is a sample command line for ISMA encryption on Linux platform:

```
mp4creator -E -W="EncoderInfo.xml" -X="UserName;danijelk" -X="UserPass;ipmp"
-X="License;danijelk;2005-08-31;2006-08-31;ODRL" -Y="isma;icm" test30.mp4
enc-test30.mp4.
```

Here is a sample command line for *OpenIPMP* encryption on Windows platform:

```
mp4creator -E -W=EncoderInfo.xml -X=UserName;danijelk -X=UserPass;ipmp
-X=License;danijelk;2005-08-31;2006-08-31;ODRL -Y=openipmp;bfs test30.mp4
enc-test30.mp4.
```

Here is a sample command line for *OpenIPMP* encryption on Linux platform:

```
mp4creator -E -W="EncoderInfo.xml" -X="UserName;danijelk" -X="UserPass;ipmp"
-X="License;danijelk;2005-08-31;2006-08-31;ODRL" -Y="openipmp;bfs" test30.mp4 enc-test30.mp4.
```

In mpeg4ip project, mp4player application is used to play the content. Besides standard parameters, mp4player now takes certain parameters needed for playing DRM protected content.

info@mutablemedia.com

DRM parameters are the following:

- -W=<xml file name>, [mandatory], set XML configuration file name.
- -X=<sensitive data>, set sensitive information.

XML configuration file contains necessary information for DRM decryption. Sensitive information is set via command line if it shouldn't appear in the XML configuration file (user name, password...). -X option can be used multiple times. <sensitive data> can be:

- UserName;<user name>, [mandatory, only one], user name for DRM server access.
- UserPass;<password>, [mandatory, only one], user password for DRM server access.

Look at a sample XML configuration file for playing (Windows paths must be changed to unix paths if playing on a Linux platform): doc/html/PlayerInfo.xml.

info@mutablemedia.com

```
- <PlayerInfo>
    <!-- OpenIPMP p12 file path -->
    <P12FilePath>\p12\</P12FilePath>
    <!-- OpenIPMP random number file -->
    <RandomNumberFilePath>.\entropy.dat</RandomNumberFilePath>
    <!-- OMA device registration database file -->
    <RegDatabasePath>\p12\ncacheagentregdb.xml</RegDatabasePath>
    <!-- OMA device rights database file -->
    <RODatabasePath>\p12\agentrodb.xml</RODatabasePath>
    <!-- OMA device certificate file -->
    <CertPath>\p12\deviceCertSigned.der</CertPath>
    <!-- OMA device private key file -->
    <PrivateKeyPath>\p12\deviceKey.der</PrivateKeyPath>
    <!-- OMA device CA certificate file -->
    <RootCAPath>\p12\OMACACert.der</RootCAPath>
    <!-- OMA device manufacturer -->
    <Manufacturer>manufacturer</Manufacturer>
    <!-- OMA device model -->
    <Model>model</Model>
    <!-- OMA device version -->
    <Version>1.0</Version>
    <!-- OMA device ROAP version -->
    <ROAPVersion>1.0</ROAPVersion>
    <!-- OMA device caching database indicator -->
    <Caching>false</Caching>
    <!-- OMA device trusted CA certificate file -->
    <TrustedCAPath>\p12\OMACACert.der</TrustedCAPath>
    <!-- OMA device supported algorithm -->
    <SupportedAlgorithm>md5</SupportedAlgorithm>
    <!-- OMA device supported algorithm -->
    <SupportedAlgorithm>md5rsa</SupportedAlgorithm>
  </PlayerInfo>
```

P12FilePath tag in XML configuration file must be set to the path where p12 file for the user identified with UserName and UserPass (see above) resides. See the Necessary stuff section for p12 file details. RandomNumberFilePath tag in XML configuration file must be set to the path where entropy.dat will reside. It can be left as it is.

RegDatabasePath, RODatabasePath, CertPath, PrivateKeyPath, RootCAPath and TrustedCAPath tags in XML configuration file are used for playing OMA DRM protected content. RegDatabasePath tag and RODatabasePath tag must be set to registration database file path and rights database file path, respectively. If the given files do not exist, they will be created. These files are expected to be XML files. CertPath tag, PrivateKeyPath tag, RootCAPath, TrustedCAPath tag must be

info@mutablemedia.com

set to device certificate file path, device private key file path, CA certificate file path and trusted CA certificate file path. See the Necessary stuff section for details.

For testing purposes, other tags in XML configuration file can be left as they are.

For all the following sample command lines, it is assumed that enc-test30.mp4 is a protected mp4 file.

Here is a sample command line for playing protected content on Windows platform:

```
mp4player -W=PlayerInfo.xml -X=UserName;danijelk -X=UserPass;ipmp
enc-test30.mp4.
```

Here is a sample command line for playing protected content on Linux platform:

```
mp4player -W="PlayerInfo.xml" -X="UserName;danijelk" -X="UserPass;ipmp"
enc-test30.mp4.
```

6 API

To see the API provided by the SDK, go to doc/html/index.html. To see the mpeg4ip integration details, go to doc/html/mpeg4ip/index.html.

info@mutablemedia.com