Software Development Kit

DRM Guidelines (PlayReady)

Status:	Final
Version:	1.0.1
Date:	7 May 2015
Author:	Smart TV Alliance inc.
Category:	Confidential

1. Change history

Version	Date	Changes
1.0	2013-12-20	Final
1.0.1	2015-05-07	Updated “2. References” (URL and version)

2. References

[1] HbbTV Content Protection using Microsoft PlayReady v1.0 (available to PlayReady licensees)

[2] OIPF DAE - Open IPTV Forum Release 1 Specification, Volume 5 - Declarative Application, V1.1, October 2009 - http://www.oipf.tv/docs/oipf-archive/OIPF-T1-R1-Specification-Volume-5-Declarative-Application-Environment-V1_1-2009-10-08.pdf

[3] Microsoft Smooth Streaming - http://www.iis.net/download/SmoothStreaming

[4] Microsoft, "[MS-SSTR]: Smooth Streaming Protocol Specification", v6.0
http://msdn.microsoft.com/en-us/library/ff469518%28v=prot.10%29.aspx

[5] Microsoft PlayReady - http://www.microsoft.com/PlayReady/Default.mspx

[6] Smart TV Alliance Technical Specification Version 2.0, 2.5, 3.0, 4.0

[7] DASH Content Protection using Microsoft PlayReady v1.2
http://download.microsoft.com/download/7/7/6/7762455C-D254-4C84-BE17-16B0C60E31FD/MPEG%20DASH%20PlayReady%201.2%20-%202014-10-08.pdf
(Note that [7] refers to CENC (ISO/IEC FDIS 23001-7:2014, Second Edition, “Information technology – MPEG systems technologies – Part 7: Common encryption in ISO base media file format files” but please also refer to [6]. One of the streaming protocols support is MPEG-DASH (ISOBMFF & CENC) according to HbbTV version 1.2.1 profile.)

[8] ISO/IEC 23009-1 Information technology –Dynamic adaptive streaming over HTTP (DASH) – Part 1 : Media presentation description and delivery formats

[9] Microsoft PlayReady Format Specification v1.14 (included in PlayReady Documentation Pack which is available to PlayReady licensees)

3. Introduction

3.1. Overview

This document describes the Guidelines for deploying PlayReady DRM in conjunction with Smooth Streaming and MPEG-DASH (optionally supported) for the Smart TV Alliance platform. It is intended for Content Providers that want to use DRM in their application. Note that information in this document only pertains to specific Alliance platform implementations of the DRM engine, and does not replace or contain specific information regarding the chosen DRM solutions ([1],[7]). These solutions are available separately. Information in this document assumes this specific knowledge of the DRM solution.
Note that this guideline can be used for the Smart TV Alliance Technical Specification Version 2.0, 2.5, 3.0 and 4.0 ([6]).

3.2. HTML5 smooth streaming

Smooth Streaming ([3],[4]) content (unprotected as well as protected) can be played using the HTML5 video element using the instructions below.

```
<video id="videoid" width="320" height="240">
  <source src="manifest_url" type="application/vnd.ms-sstr+xml" />
</video>
..
document.getElementById("videoid").play();
```

3.3. MPEG-DASH (Optional with PlayReady)

Please note that PlayReady in conjunction with MPEG-DASH ([8]) is optional. It can be played using the HTML5 video element using the instructions below.

```
<video id="videoid" width="320" height="240">
  <source src="manifest_url" type="application/dash+xml" />
</video>
..
document.getElementById("videoid").play();
```

4. API for PlayReady

4.1. Overview

This chapter outlines the playback of PlayReady ([5]) content using the HTML5 video object and the DRMAgent object. Applications shall use the OIPF DRMAgent API. It is mandatory for the proactive license acquisition (pre-delivery) case (refer to 4.3). It may be used for the reactive license acquisition (post-delivery). For more information refer to [1].

4.2. application/oipfDrmAgent

This is an example of the DRMAgent object [2], as used for the playback of PlayReady content:

```
<object id="drmplugin" type="application/oipfDrmAgent"
style="visibility:hidden"></object>
```

It is a non-visual embedded object of type "**`application/oipfDrmAgent`**", which is accessed using the Javascript API as described below to enable in-session message exchange from the web page to an underlying DRMAgent.

Please check chapter "7.6.1 The application/oipfDrmAgent embedded object" of [2]. The described properties, methods and events are supported. Refer to [1] for additional implementation details.

4.2.1. API

Below are examples of relevant properties and methods from the oipfDrmAgent object. This example is used for proactive license acquisition using PlayReady Web Initiator [9]. For more information refer to [2].

Proactive license acquisition with sendDRMMessage API.

```
// send licence request using sendDRMMessage and the right MIME type
function sendLicenceRequest(){
	// The PlayReady Initiator is created by the content server.

	var DRMSystemID = "urn:dvb:casystemid:19219";
	var pluginElement = document.getElementById('drmplugin');
	var xmlLicenceAcquisition = (PlayReady License Acquisition Web Initiator XML
string) - (*)
	pluginElement.onDRMMessageResult = HandleOnDRMMessageResult;
	pluginElement.onDRMRightsError = HandleOnDRMRightsError;
	pluginElement.sendDRMMessage('application/vnd.ms-playready.initiator+xml',
	xmlLicenceAcquisition, drm.DRMSystemID);
}
```

Please refer to [1] section 3.1.2.1 for more information.

Handling DRM Message Result with onDRMMessageResult API.

```
function HandleOnDRMMessageResult(msgID, resultMsg, resultCode) {
	var error = "";
	log("HandleOnDRMMessageResult msgID:" + msgID + " resultMsg:" + resultMsg +
	" resultCode: " + resultCode);
	if (resultCode == 0) {
		try {
			var videoid = document.getElementById("videoid");
			videoid.play();
		} catch (e) {
			log("HandleOnDRMMessageResult() :" + e.message);
		}
	} else {
		switch (resultCode) {
			//used for printing the correct error message
			case 1:
				error = "Unknown error";
				error += " - " + HtmlEncode(resultMsg);
```

```
                        break;
                case 2:
                        error = "Cannot process request";
                        break;
                case 3:
                        error = "Unknown MIME type";
                        break;
                case 4:
                        error = "User Consent Needed";
                        break;
            }
            log("Playback failed. Error:" + error);
        }
}
```

4.3. License acquisition

4.3.1. Proactive license acquisition (Pre-delivery)

Apps that require user/device authentication and wish to include an authorization token within the license request may do so via the <CustomData> field of the license initiator. The <CustomData> is created by the content server or the web application. If <CustomData> is filled in it is used in the license acquisition process to verify the user/device.

The oipfDrmAgent API describes the handling of DRM messages. The sendDRMMessage license acquisition is triggered by providing a PlayReady LicenseAcquisition initiator as the DRM message. Note that there are restrictions on the WRMHEADER ([9]) size for some Smart TV platforms. Refer to Annex A for the details.

4.3.2. Reactive license acquisition (Post-delivery)

Reactive license acquisition is the sequence for playback of protected Smooth Streaming content in which the license acquisition is triggered upon playback using the PlayReady Object ([9]) (including WRMHEADER) from the Manifest. Optionally apps that require user/device authentication and wish to include an authorization token within the license request, may set a Challenge CustomData by calling sendDRMMessage API (Please refer to [1] section 3.1.2.6). The app also may override the license acquisition URL by calling sendDrmMessage API (Please refer to [1] section 3.1.2.5).

4.3.3. Optional functionality

Please refer to 3.5.1.1. PlayReady Features in [6]. Optional PlayReady features may not be supported in the Smart TV.
For non-supported initiators, onDRMMessageResult returns 2 (Cannot process request) as resultCode.

4.4. Error handling

4.4.1. Error code

In case of an error, the error attribute of the HTML5 video object shall be set to *MEDIA_ERR_DECODE*.
Please note that there is no option to be notified of DRM licensing errors during playback (e.g. onDRMRightsError).

5. Annex A. Diversity Information

5.1. TP Vision

Some specific information of error handling regarding Smooth Streaming playback on TP Vision platforms (refer also to the diversity handling guidelines):

If the Manifest is not valid, the error attribute of the HTML5 video element (media.error.code) is set to MEDIA_ERR_DECODE. If the Manifest points to an unsupported audio or video format(codec), the error attribute of the HTML5 video element (media.error.code) is set to *MEDIA_ERR_SRC_NOT_SUPPORTED*.

For the WRMHEADER, the following size restrictions apply: Length of the LA_URL in WRMHEADER shall not exceed 1024 bytes.