Interface specifications

PRM Signaling

Version 1.0.8

Status: Approved

Filename	PRM-Signaling_IRS.docm
Date	July 17, 2017
Document number	**01ABCL0000000073**
Author(s)	Lejosne François, Gaiddon Frédéric
Information Domain	DVS
Client/Project	PRM/MDRM
Owner	CAS-PU > Head-End

CH-1033 Cheseaux, Switzerland.

Tel: +41 21 7320311 Fax: +41 21 7320100

www.nagra.com

Security Policy of Nagravision Kudelski Group

Any recipient of this document, without exception, is subject to a Non-Disclosure Agreement (NDA) and access authorization.

STRICTLY CONFIDENTIAL

Contents

List of tables

NO TABLE OF FIGURES ENTRIES FOUND.

List of figures

FIGURE 1.

Preface

Document purpose

Purpose of this document is to describe the PRM signaling.

Audience

This document is intended for the following people:

- The partner development team who are to use this interface
- The internal R&D teams who are to implement this interface
- The integration teams who are to validate this interface

Document Structure

This document is divided into the following parts:

Chapter 1: Signaling with PRM DRM

Chapter 1 describes which signaling the PRM DRM can support and the purpose of signaling.

Chapter 2: HLS Signaling

Chapter 2 describes the HLS signaling by PRM DRM.

Chapter 3: DASH Signaling

Chapter 3 describes the DASH signaling by PRM DRM.

Chapter 4: CUSTOM Signaling

Chapter 4 describes the CUSTOM signaling by PRM DRM.

Chapter 5: SMOOTH Signaling

Chapter 5 describes the SMOOTH signaling by PRM DRM.

Used Conventions

- The used convention for dates in numerical format is YYYY-MM-DD.

  Example: 2009-08-10 is equivalent to August 10th, 2009.

Related Documents

[1] Nagra IKeyAndSignalization IRS / 01ABCL0000000065/ 1.1.0 / //HEP/Interfaces/Encoder-KSS/PublishedDoc/tags/deliveries/Encoder-KSS_1.1STD0/
[2] PRM Syntax / 1.0.0
[3] HLS format specification
[4] ISO/IEC 23009-1: Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation and segment format (DASH MPD and segment format)
[5] ISO/IEC 14496-12: Information technology – Coding of audio-visual objects – Part 12: ISO Base Media File Format (ISOBMFF description)
[6] ISO/IEC FDIS 23001-7: Information technology – MPEG systems technologies – Part 7: Common encryption in ISO base media file format files (ISOBMFF common encryption)
[7] Protected Interoperable File Format (PIFF)
[8] IETF RFC 4122, A Universally Unique IDentifier (UUID) URN Namespace, July 2005

1. Overview of signaling with PRM DRM

1.1 Introduction

Each PRM generates its own signaling referencing the license required to decipher a content.

The signaling is generated by PRM key server (KSS) and retrieved by the encoder/chunker upon content key request. The encoder then encrypts the content using the content key and inserts the signaling in some content metadata before publishing the encrypted content. When a player later accesses this published content, it detects thanks to this signaling that the content is encrypted, transmits this signaling to the PRM client so that it can retrieve, or find back in its cache, the corresponding license. Player can then ask the PRM client to decipher the content.

In case of live key change, same flow is processed, a new signaling is found in the linked content metadata and PRM client can retrieve, or find back in its cache, the new license to use.

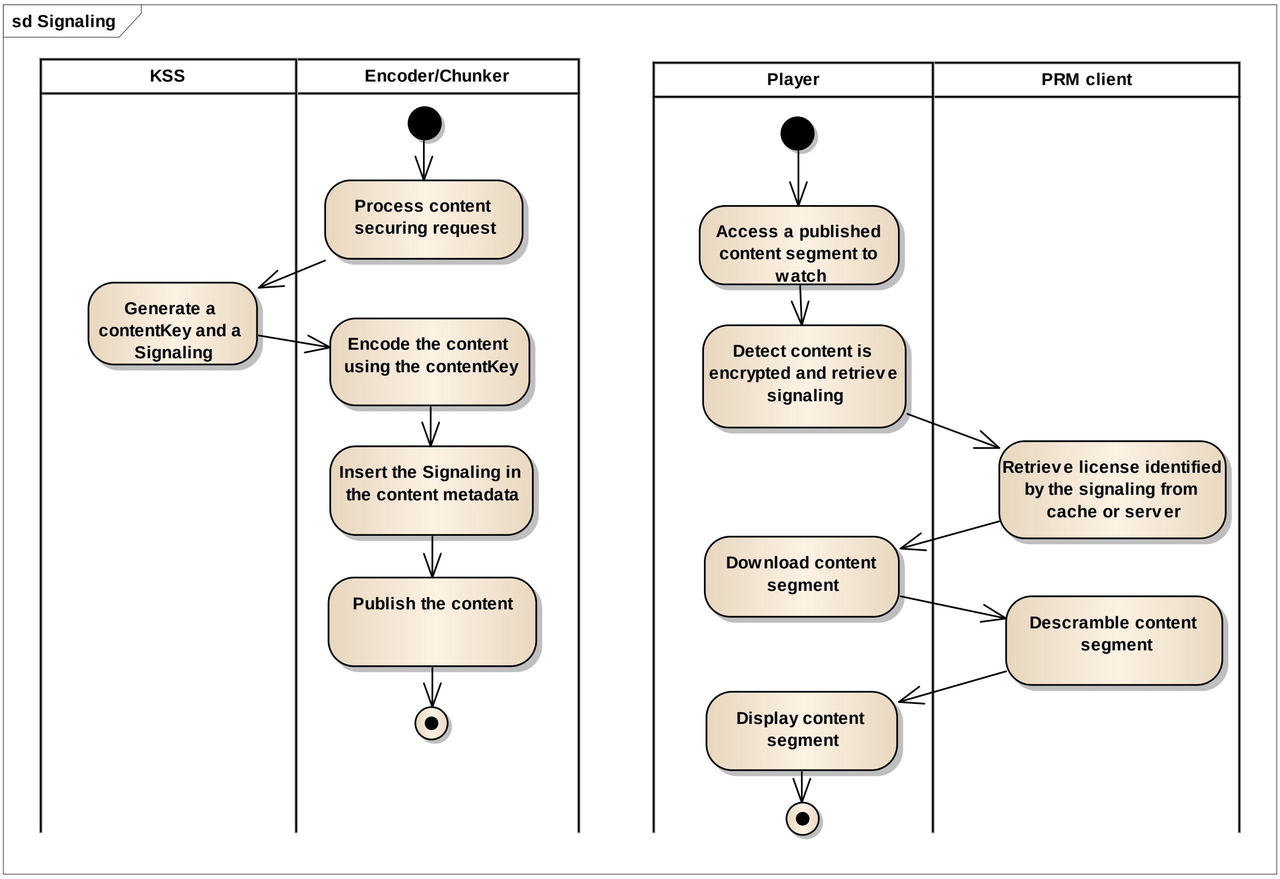
sd Signaling
KSS
Encoder/Chunker
Process content securing request
Generate a contentKey and a Signaling
Encode the content using the contentKey
Insert the Signaling in the content metadata
Publish the content
Player
PRM client
Access a published content segment to watch
Detect content is encrypted and retrieve signaling
Retrieve license identified by the signaling from cache or server
Download content segment
Descramble content segment
Display content segment

Figure 1. Signaling usage overview

There are 3 types of signaling depending on the targeted streaming protocol and a custom one for other transmission modes (like progressive download):

- HLS,
- DASH
- SMOOTH
- CUSTOM

It is mandatory to embed the signaling in the content metadata in order to allow key rotation, license caching and offline consumption use cases on devices.

1.2 PRM Signaling creation interfaces

The KSS supports several interfaces that are able to return PRM signaling. The table below presents ability of KSS to return PRM signaling according to the interface trigged.

Interfaces published	Streaming mode			
	HLS	DASH	SMOOTH	CUSTOM
HTTP Envivio	N/S	N	N	N
Nagra IKeyDeliveryV2: - GetKey - ImportKey	N/S	N	N	N
Nagra IKeyAndSignalization: - GetKeyAndSignalization	S	S	D	S
Harmonic IKey: - GetKey	S	N	N	N
Harmonic IKey: - GetKeyAndSignalization	N	S	N	N
OIPF (in progress) 1: named 'HAS' by OIPF. 2: named 'PRIVATE' by OIPF.	D[1]	D	N	D[2]

Where letters S, D, N means:

- **S** – This interface can return PRM or 3rd party DRM signaling for this streaming mode.
- **D** – This interface can return 3rd party DRM signaling but not yet PRM signaling for this streaming mode.
- **N** – This interface does not generate signaling for this streaming mode.

Multi DRM Context:

The KSS is able to manage several DRM systems and to return the signaling of other DRMs allowing other DRMs to find back licenses containing the same content Key.

DRM systems can be configured through the PRM configuration console specifying the Drm name ('PRM' for PRM DRM) and for third party DRM, the DRM server URL (ex. 'http://*hostname*:*port*/drmproxystub-ear-drmproxystub-keyAndSignalization-WS/MarlinKeyAndSignalization?wsdl').

This document only focuses on PRM signaling.

The resulting signaling returned to encoder complies with xml schema of Nagra IKeyAndSignalization interface.

Document structure:

Following chapters present for each supported streaming standard :

- x.1: a description of the signaling syntax.
- x.2: a description of where this signaling shall be located inside the streaming metadata format.
- x.3: a description of the format output from the PRM key server to a signaling request. Note that key requests done in this context relate only to DRM system PRM.
- x.4: a description on how the player extracts PRM syntax from the streaming metadata .

2. HLS Signaling

2.1 PRM HLS Signaling description

Following figure contains a PRM HLS signaling example:

```
http://www.nagra.com/key=Gone+in+the+wind;prm=eyJjb250ZW50SWQiOiJHb25lIGluIHRoZSB
3aW5kIiwia2V5SWQiOiI5MWExZTQ0Ny02ODRiLTRhY2UtYjZjZS00MDExNjBmMDdmMDEifQ
```

Figure 1. PRM HLS signaling example.

HSL PRM signaling is composed of the following 4 parts:

1. A global valid URI prefix ending with "="

   In above example, it is the string "***http://www.nagra.com/key=***". This is the parameter 'HlsKeyUriPrefix' to be set through the PRM configuration console.

2. A content identifier **encoded to be URL safe** and corresponding to the contentId field.

   In above example, it is the string `Gone+with+the+Wind`

Parameter value extracted from URL	Parameter decoded
Gone+with+the+Wind	`Gone with the Wind`

The encoding of key parameter follows the section 17.13.4 of HTML specification (http://www.w3.org/TR/html4/interact/forms.html#h-17.13).

3. A string called PRM syntax prefixed with "&prm="

   It is encoded in Base64 URL safe and contains some information private to PRM system allowing retrieving the necessary license to be able to decipher the corresponding segments. Its format is described in [2].

   In above exemple, it is the string: " `&amp;prm=eyJjb250ZW50SWQiOiJHb25lIGluIHRoZSB3aW5kIiwia2V5SWQiOiI5MWExZTQ0Ny02ODRiLTRhY2UtYjZjZS00MDExNjBmMDdmMDEifQ`"

   Note that in the get key answer, the character '&' is represented by the predefined entity '&amp;' which is the XML representation of '&'.

4. An optional URL suffix (and not provided in the example). This parameter 'HlsKeyUriSuffix' can be also set through the PRM configuration console and must start with character '&'.

2.2 Legacy PRM HLS Signaling description

Prior the introduction of PRM syntax in PRM 3.0, encoders were using a simpler signaling. This signaling does not contain any PRM syntax.

Following figure contains a PRM HLS legacy signaling example:

```
http://www.nagra.com/key=Gone+in+the+wind
```

Figure 1. PRM legacy HLS signaling example.

It is composed of:

1. a valid URI prefix ending with character '='

   Which is in the example above the string "***http://www.nagra.com/key=***"? This is the parameter 'HlsKeyUriPrefix' to be set through the PRM configuration console.

2. A content identifier which is here the string **`Gone+with+the+Wind`** encoded to be URL safe and corresponding to the contentId field.

3. An optional URL suffix (and not provided in the example). This is the parameter 'HlsKeyUriSuffix' to be set through the PRM configuration console and must start with character '&'.

2.3 PRM Signaling location in HLS metadata

HTTP Live Streaming sends audio and video as a series of small files, typically of about 10 seconds duration, called media segment files. An index file, or playlist, in the form of an m3u8 file, provides an ordered list of the URLs of the media segment files and information tags.

Each Tag is prefixed by a # and followed by: and an attribute list. #TAG:<attribute-list>

Attribute list is composed of comma separated couples of attributeName =AttributeValue.

Refer to [3] for detailed information on HLS playlist format.

The EXT-X-KEY tag specifies how to decrypt one or multiple encrypted media segments.

The HLS signaling returned by PRM key server is to be inserted as the attribute value of the attribute named "URI" in the tag named "EXT-X-KEY" preceding the encrypted segments URLs in the playlist.

Note that encoder may use the attribute "KEYFORMAT" to include multiple DRM signaling allowing multiple DRM clients to retrieve licenses containing the same content Key. The value of the keyformat shall be known by the player to allow to select the signaling targeting PRM.

Example of an EXT-X-KEY tag where PRM signaling was inserted:

```
EXT-X-KEY:METHOD =AES-128,URI="http://key-
location?contentID=1234&prm=ewqTY29udGVudElklDogkzEyMzQ1Njc4OZQsIAqTa2V5SWSUICAgI
CAgOiCTNzM4ZTk3NDgtOWQ5Ni00N2ViLWEzYTctZjA4NzFmODkyMzIylAp9Cg",
KEYFORMAT="PRMNAGRA",KEYFORMATVERSION="1"
```

2.4 PRM signaling creation

PRM signaling can be obtained from one the KSS provided interfaces as listed in §1.2 PRM Signaling creation interfaces. The HLS PRM signaling is given by the text node of the tag **<keyUri>**.

2.5 PRM signaling retrieval

The player is in charge of retrieving the signaling from the playlist and sending it to the PRM client. API to send the signaling to the PRM client is detailed in the DVL/PAK/PRMC APIs.

Since PRM4.0, PRMC provides an interface so that player can get the value of the KEYFORMAT.

Previous PRM versions do not and this value shall be hardcoded in the client. "PRMNAGRA" is the commonly used value.

3. DASH Signaling

3.1 PRM Signaling location in DASH metadata

Under the MPEG DASH standard, the media segments can contain any type of media data. However, the standard provides specific guidance and formats for use with two types of segment container formats – MPEG-2 Transport Stream (MPEG-2 TS) and ISO base media file format (ISO BMFF).

MPEG-2 TS is the segment format that HLS currently uses, while ISO BMFF is what Smooth Streaming and HDS currently use.

DASH presents available stream content to the media player in a manifest (or index) file called the Media Presentation Description (MPD).

PRM signaling can be found in two areas of DASH signaling:

- in media presentation description (MPD), refer to [4] for detailed information. It can either be present directly under the PRM descriptor (legacy) and/or in a pssh box inserted in the MPD under a pssh descriptor (current standard) as described in [6].
- in Protection System Specific Header (PSSH) box included in movie 'moov' or movie fragment 'moof', refer to [5]for detailed information.

3.1.1 DASH PSSH format

The Protection System Specific Header (PSSH) box has a syntax that conforms to the following definition as specified in [5]:

```
aligned(8) class ProtectionSystemSpecificHeaderBox extends FullBox('pssh', version=0, flags=0)
{
         unsigned int(8)[16]       SystemID;
         unsigned int(32)          DataSize;
         unsigned int(8)[DataSize] Data;
}
```

The *SystemID* is the UUID identifying the protection system. In order to identify PRM signaling data, the *SystemID* must be matched against the DASH PRM system identifier.

The DataSize is the size of the protection specific data that must be provided to the DRM system for accessing the content – obtaining a right and then a key for descrambling. It is a 4 bytes integer in MSBF representation.

The Data is a table of bytes containing the DRM data and which size is DataSize. The PRM signaling is the full Data field and must be provided as such to the PRM client.

The pssh box is provided as a base64 representation of a byte array:

AAAAinBzc2gAAAAArbQcJC2/Sm2Vi0RXwNJ7lQAAAGpleUpqYjI1MFpXNTBTV1FpT2lKSGIyN
WxJR2x1SUhSb1pTQjNhVzVrSWl3aWEyVjVTV1FpT2lJNU1XRXhaVFEwTnkwMk9EUmlMVFJoWT
JVdFlqWmpaUzAwTURFeE5qQm1NRGRtTURFaWZR

Same example of the previous pssh in a hexadecimal byte array representation:

0000008A7073736800000000adb41c242dbf4a6d958b4457c0d27b950000006A65794A6A6
23235305A57353053575751694F694A486232356C49476C754948526F5A5342336157356B49
697769613256355357751694F6949354D5745785A5451304E7930324F4452694C545268593
25574596A5A6A5A5330304D4445784E6A426D4D44646D4D4445696651

It is interpreted as:

Field	**Value**	**Comment**
Box size	0000008A	138 bytes
Box type	70737368	'pssh'
version, flags	00000000	-
SystemID	adb41c242dbf4a6d958b4457c0d27b95	Shall matches PRM system ID.
DataSize	0000006A	112 bytes of PRM syntax
Data	65794A6A623235305A573530535751694F694A486232356C49476C754948526F5A5342336157356B496977696132563553535751694F6949354D5745785A5451304E7930324F4452694C54526859325574596A5A6A5A5330304D4445784E6A426D4D44646D4D4445696651	PRM Syntax [2] to be provided to the PRM client: hexadecimal values of the base64 url safe representing the PRM syntax

3.1.2 DASH MPD signaling

MPD is an XML file which layout conforms to the schema provided in DASH MPD and segment format specification [4]. In this XML file a ContentProtection element might be found at:

- AdaptationSet/ContentProtection
- Representation/ContentProtection
- SubRepresentation/ContentProtection

This ContentProtection element is specified by the *DescriptorType* complex type defined as follow:

```
  <!-- Descriptor -->
  <xs:complexType name="DescriptorType">
    <xs:sequence>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="schemeIdUri" type="xs:anyURI" use="required"/>
    <xs:attribute name="value" type="xs:string"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
```

It exposes the PRM uuid urn [RFC4122] as the *schemeIdUri* attribute. This value uniquely identifies the DRM system provider ('urn:uuid:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx').

Even if this schema states that the DRM data may be provided either as a child element or as an attribute (each in a specific namespace). PRM always provide it as a child element as illustrated in the following example. The contentProtection element shown here, contain an embedded cenc:pssh element containing a base64 encoded 'pssh' box.

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:DASH:schema:MPD:2011"
  xmlns:drm="http://example.net/052011/drm"
  xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011 DASH-MPD.xsd"
  type="static"
  mediaPresentationDuration="PT3256S"
  minBufferTime="PT10.00S"
  profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">

  <BaseURL>http://cdn.example.com/movie23453235/</BaseURL>

  <Period>
    <!-- Audio protected with a PAK entitlement -->
    <AdaptationSet  mimeType="audio/mp4" codecs="mp4a.0x40" lang="en"
      subsegmentStartsWithSAP="1"
      subsegmentAlignment="true">
      <ContentProtection schemeIdUri="urn:uuid:adb41c24-2dbf-4a6d-958b-4457c0d27b95">
        <prm:PRM>
```

```
          <prm:PRMSignalization>
        eyJjb250ZW50SWQiOiJHb25lIGluIHRoZSB3aW5kIiwia2V5SWQiOiI5MWExZTQ0Ny02ODRiLTRhY2UtYjZjZS00M
DExNjBmMDdmMDEifQ
          </prm:PRMSignalization>
        </prm:PRM>
         <!-- base64 encoded 'pssh' box with this PRM SystemID -->
        <cenc:pssh>
        AAAAinBzc2gAAAAArbQcJC2/Sm2Vi0RXwNJ7lQAAAGpleUpqYjI1MFpXNTBTV1FpT2lKSGIyNWxJR2x1SUhSb1pTQ
jNhVzVrSWl3aWEyVjVTV1FpT2lJNU1XRXhaVFEwTnkwMk9EUmlMVFJoWTJVdFlqWmpaUzAwTURFeE5qQm1NRGRtTURFaWZR
        </cenc:pssh>
      </ContentProtection>
      <Representation id="1" bandwidth="64000">
        <BaseURL>audio/en/64.mp4</BaseURL>
      </Representation>
    </AdaptationSet>
    <!-- Audio protected with embedded information defined by 'ZZZZ' -->
    <AdaptationSet  mimeType="audio/mp4" codecs="mp4a.0x40" lang="fr"
      subsegmentStartsWithSAP="1"
      subsegmentAlignment="true">
      <ContentProtection schemeIdUri=" urn:mpeg:dash:mp4protection:2011" value="ZZZZ"/>
      <Representation id="3" bandwidth="64000">
        <BaseURL>audio/fr/64.mp4</BaseURL>
      </Representation>
    </AdaptationSet>
    <!-- Timed text in the clear -->
    <AdaptationSet  mimeType="application/ttml+xml" lang="de">
      <Representation id="5" bandwidth="256">
        <BaseURL>subtitles/de.xml</BaseURL>
      </Representation>
    </AdaptationSet>
    <!-- Video protected with a specified license -->
    <AdaptationSet  mimeType="video/mp4" codecs="avc1" subsegmentAlignment="true"
subsegmentStartsWithSAP="2">
      <ContentProtection schemeIdUri="urn:uuid:adb41c24-2dbf-4a6d-958b-4457c0d27b95">
        <prm:PRM>
          <prm:PRMSignalization>
        eyJjb250ZW50SWQiOiJHb25lIGluIHRoZSB3aW5kIiwia2V5SWQiOiI5MWExZTQ0Ny02ODRiLTRhY2UtYjZjZS00M
DExNjBmMDdmMDEifQ
          </prm:PRMSignalization>
        </prm:PRM>
         <!-- base64 encoded 'pssh' box with this PRM SystemID -->
        <cenc:pssh>
        AAAAinBzc2gAAAAArbQcJC2/Sm2Vi0RXwNJ7lQAAAGpleUpqYjI1MFpXNTBTV1FpT2lKSGIyNWxJR2x1SUhSb1pTQ
jNhVzVrSWl3aWEyVjVTV1FpT2lJNU1XRXhaVFEwTnkwMk9EUmlMVFJoWTJVdFlqWmpaUzAwTURFeE5qQm1NRGRtTURFaWZR
        </cenc:pssh>
      </ContentProtection>
      <BaseURL>video/</BaseURL>
      <Representation id="6" bandwidth="256000" width="320" height="240">
        <BaseURL>video256.mp4</BaseURL>
      </Representation>
      <Representation id="7" bandwidth="512000" width="320" height="240">
        <BaseURL>video512.mp4</BaseURL>
      </Representation>
      <Representation id="8" bandwidth="1024000" width="640" height="480">
        <BaseURL>video1024.mp4</BaseURL>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```

Player will have to provide the PRM Client with the full `ContentProtection` element. Two instances of this element for PRM (The Nagravision DRM) appear in the previous example (the highlighted parts).

Both elements expose the PRM uuid urn [8] as the `schemeIdUri` attribute. This value uniquely identifies the Nagravision DRM system (`'urn:uuid:adb41c24-2dbf-4a6d-958b-4457c0d27b95'`).

3.2

3.2 PRM signaling creation

PRM signaling can be obtained from one the KSS provided interfaces as listed in §1.2 PRM Signaling creation interfaces.

The response contains DASH signaling for DRM system PRM. Both signaling type are generated.

- MPD signaling : part enclosed in the manifestHeader starting with <ContentProtection> and ending with </ContentProtection>
- pssh signaling : part starting with <psshBox> and ending with </psshBox>

3.3 PRM signaling retrieval

The player is in charge of retrieving the signaling from the playlist and sending it to the PRM client. API to send the signaling to the PRM client is detailed in the DVL/PAK/CCL APIs.

Since PRM4.0, PRMC provides an interface so that player can get the PRM UUID to be matched against the DASH PRM system identifier, this value uniquely identifies the Nagravision DRM system.

- for MPD signaling :

In this case the expected PRM signaling to be provided for the related media segments is the full *ContentProtection* element.

- for pssh signaling :

The full Data field must be provided as such to the PRM client.

4. CUSTOM Signaling

4.1 PRM Custom Signaling description

Following figure contains a PRM Custom signaling example:

```
eyJjb250ZW50SWQiOiJHb25lIGluIHRoZSB3aW5kIiwia2V5SWQiOiI5MWExZTQ0Ny02ODRiLTRhY2UtY
jZjZS00MDExNjBmMDdmMDEifQ
```

Figure 1. Key and CUSTOM signaling request.

It is only composed of the PRM Syntax described in [2].

4.2 PRM Custom Signaling location in content metadata

A custom specification must be defined and shared between encoders and players to define how to include the PRM signaling in the content metadata.

4.3 PRM signaling creation

PRM signaling can be obtained from one the KSS provided interfaces as listed in §1.2 PRM Signaling creation interfaces.

The response shows CUSTOM signaling for DRM system PRM. The CUSTOM signaling (see custom tag <custom>) handles the tags **<drmsystemId>**, **<drmName>** and **<data>**. The text node of the tag **<data>** handles a string as PRM Syntax.

4.4 PRM signaling retrieval

The player is in charge of retrieving the signaling from the playlist and sending it to the PRM client. API to send the signaling to the PRM client is detailed in the DVL/PAK/CCL APIs.

5. SMOOTH Signaling

5.1 PRM Smooth signaling description

As specified in [7] the DRM interoperability is provided by the use of an extended Protection System Specific Header (PSSH) box located only in movie fragment box ('moov'). It has the following syntax:

```
aligned(8)   class   ProtectionSystemSpecificHeaderBox   extends   FullBox(‘uuid’,
extended_type=0xd08a4f18-10f3-4a82-b6c8-32d8aba183d3, version=0, flags=0)

{

unsigned int(8)[16] SystemID;

unsigned int(32) DataSize;

unsigned int(8)[DataSize] Data;

}
```

It is very similar to DASH PSSH signaling but with an modified extended header. The three fields described here have the same semantic as those described for the DASH PSSH signaling.

The SystemID is again an UUID identifying the protection system. It must be matched directly against the SMOOTH PRM system identifier.

The PRM signaling is the full *Data* field and must be provided unmodified to the PRM client.

5.2 PRM Signaling location in smooth streaming metadata

A Movie Box (“moov“) is a container box whose sub-boxes define the metadata for a presentation.

The data objects used by the DRM specific methods for retrieving the decryption key and rights object or license associated with the file are stored in the Protection System Specific Header box.

Any number of these boxes MAY be contained in the Movie Box (“moov“), each corresponding to a different DRM system. The Boxes and DRM system are identified by a SystemID. The data objects used for retrieving the decryption key and rights object are stored in an opaque data object of variable size within the Protection System Specific Header Box.

Refer to [7] for a detailed description of the PIFF format.

The ProtectionSystemSpecificHeaderBox is composed of 3 fields:

- SystemID : to be filled with received DrmSystemId
- DataSize : specifies the size of the Data member
- Data : to be filled with the generated signaling

5.3 PRM signaling creation

The PRM DRM accepts requests with SMOOTH streaming mode but does not build SMOOTH signaling and does not stop signaling process. In case of Multi DRM context, the SMOOTH signaling could be returned by third part DRM like PlayReady Server.

Notes: The PRM DRM returns an error to indicate that SMOOTH signaling is not managed until KSS version 2.0STD2. On higher KSS version, it acts like others DRM.

5.4 PRM signaling retrieval

The player is in charge of retrieving the signaling from the playlist and sending it to the PRM client. API to send the signaling to the PRM client is detailed in the DVL/PAK/CCL APIs.

Since PRM 4.0, SMOOTH PRM system identifier can be retrieved from the PRMC by the player.

The PRM signaling is the full *Data* field and must be provided unmodified to the PRM client.

Glossary

Acronym	Definition
CAS	**Conditional Access System** The overall security system for providing and preventing access to digital interactive television system. It is composed of an IMS, ciphering units and a CA/DRM Server.
DASH	**Dynamic Adaptive Streaming over HTTP**
DRM	**Digital Rights Management** Umbrella term to cover all sorts of active or passive mechanisms controlling the availability of locally stored digital content
HLS	**HTTP Live Streaming**
KSS	**Keys and Signalization Synchronizer** The server which generates the PRM signaling.
PRM	**Persistent Rights Management** Mechanisms used to protect the content recorded in the STB. PRM allows the pay TV operator to control the recording, the consumption and the export of content by means of usage rights.
SMOOTH	an IIS Media Services extension that enables adaptive streaming of media to clients over HTTP
URI	**Uniform resource identifier**
URL	**Uniform Resource Locator** Address of a file (resource) accessible on the Internet.
UTF-8	**UCS Transforming Format 8** UTF-8 is an alternative coded representation form for all the characters of the UCS. It can be used to transmit text data through communication systems which assume that individual octets in the range 00 to 7F have a definition according to ISO/IEC 4873. UTF-8 is a good way to go for using Unicode under Unix-style operating systems.
VOD	**Video On Demand** umbrella term for a wide set of technologies whose common goal is to enable individuals to select video streams from a central server for viewing on a television or computer screen

End of document