EZDRM Shaka Packager CPIX 2.x DRM Keys Guide

Table of Contents

Version 1.1 (published August 5, 2025)

Introduction

This document outlines the raw key format for Shaka Packager and generating DRM keys with EZDRM. For more details visit: https://google.github.io/shaka-packager/html/tutorials/raw_key.html

To download Shaka-Packager visit: https://github.com/google/shaka-packager/releases

Universal DRM - Widevine & PlayReady

EZDRM Universal DRM is a combination of Google Widevine Modular with Microsoft PlayReady; both using linked CENC keys over DASH streaming. This enables a content owner to encrypt the media once with CENC keys and deliver either a PlayReady License or a Widevine License depending on the player and platform calling for a license.

Generating Keys

To request the DRM keys from EZDRM to package the media, there are two options, you can call the EZDRM web service in a browser, or you can script this process with curl or other web service calls.

Option 1: Request DRM keys using EZDRM CPIX Web Service

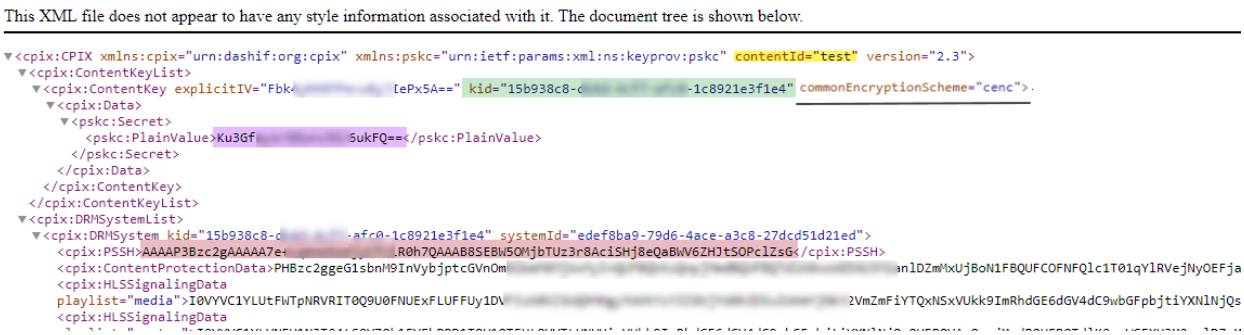
1. Call the EZDRM web service in a browser: https://cpix.ezdrm.com/keygenerator/**cpix2**.aspx?**k=kid**&**u=username**&**p=password**&**c=resourcename**

   The parameters are as follows:

Parameter	Description
k	kid or Key ID value (client generated) in GUID format*
u	EZDRM username
p	EZDRM password
c	Content ID – generic resource name/identifier (client generated) – passed into **id** field

   * To generate a GUID for the k value, you can use a GUID generator like the one found here: https://www.guidgenerator.com/.

2. The response from EZDRM will look like this:



- **id** – c value returned, generic resource name/identifier (client generated)
- **kid** – Key ID in GUID format (client generated)*
- **pskc:Secret key**– the Secret Content Encryption Key in Base 64 generated by EZDRM and returned as a plain value.
- **PSSH** – The modular specific protection system specific header (PSSH) data for the encryption process; Base 64 encoded.

* To generate a GUID for the k value, you can use a GUID generator like the one found here: https://www.guidgenerator.com/.

Here Is the example XML return:

```
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" contentId="test" version="2.3">
<cpix:ContentKeyList>
<cpix:ContentKey explicitIV="Fbk4yNtXXXXXXXyJIePx5A==" kid="15b938c8-XXXX-4cf7-XXXX-1c89XXXXf1e4" commonEncryptionScheme="cenc">
<cpix:Data>
<pskc:Secret>
<pskc:PlainValue>Ku3GfeXXXXXXX+z3O/6ukFQ==</pskc:PlainValue>
</pskc:Secret>
</cpix:Data>
</cpix:ContentKey>
</cpix:ContentKeyList>
<cpix:DRMSystemList>
<cpix:DRMSystem kid="15b938c8-XXXX-4cf7-XXXX-1c89XXXXf1e4" systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed"
>
```

```
<cpix:PSSH>AAAAP3Bzc2gAAAAA7e+LqXnWSs6jyCfc1R0h7XXXXXXXXXX5OMjbTUz3r8AciSHj8eQaBWV6ZHJtSOPclZsG</cpix:PSSH
>
<cpix:ContentProtectionData>PHBzc2ggeG1sbnM9InVybjptcGVnOmNlbmM6MjAxMyI+QUFBQVAzQnpjMmdBQUFBQTdlK0xxWG5XU3
M2anlDZmMxUjBoN1FBQUFCOFNFQlc1T01qYlRVejXXXXXXXXXXjhlUWFCV1Y2WkhKdFNPUGNsWnNHPC9wc3NoPg==</cpix:ContentPro
tectionData>
```

Option 2: Request DRM keys with curl

The second option to request DRM keys from EZDRM is to script the process with curl or another web service call.

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service.

```
curl -v "https://cpix.ezdrm.com/keygenerator/cpix.aspx?k=kid&u=username&p=password&c=test"
```

Universal DRM – Key Value Definitions

Widevine/PlayReady

- **playready_extra_header_data**: this flag is required for PlayReady. The flag is formatted as follows ending with your EZDRM **pX value**. The **pX value** is branded for your account and is account specific. Your PX value is always the same no matter what packager you are using. It is also the last six characters of your EZDRM Profile ID, as shown in this example:

  Your PlayReady DRM Profile ID is: **4BE3XX9X-XX52-5RT5-87XX-34XXXX123456**

  The PlayReady header data is formatted as follow:

  "<LA_URL>**https://playready.ezdrm.com/cency/preauth.aspx?pX=EXXXXX**</LA_URL>"

- **key_id:** The **kid** used for encryption (also known as KID); Base 64 encoded with no dashes.

- **key:** The DRM content encryption key (128 bit key); Base 64 encoded.

  For the **key** value use the **pskc:Secret key** value and decode the Plain Value tag from Base 64 to HEX format in lowercase (no dashes). An example decoder can be found at: https://tomeko.net/online_tools/base64.php?lang=en

  pskc:Secret key (Base 64) = Ku3GfXXXXXXu+z3O/6ukFQ==

  key (HEX) = 2aedXXXXecacXXXXefb3dceXXXXa415

- **PSSH:** The modular specific protection system specific header (PSSH) data for the encryption process; Base 64 encoded.

  For the PSSH value, use the **PSSH** and decode the Plain Value from Base 64 to HEX format. An example decoder can be found at: https://tomeko.net/online_tools/base64.php?lang=en

PSSH (Base 64) =
AAAAP3Bzc2gAAAAA7e+LqXnWSs6jyCfc1R0h7XXXXXXXXXX5OMjbTUz3r8AciSHj8eQaBWV6ZHJtSOPclZsG

PSSH (HEX) =
0000003F707373680XXXXXXXXXXXXXXXXXXXXXACEA3C827DCD51D21ED0000001F121015B938C8DB4D4CF7AFC01C8921E3F1E41A05657A64726D48E3DC959B06

Universal DRM – Raw Key Format

The example raw key format for Universal DRM (Widevine & PlayReady):

```
packager-win.exe
in=c:\code\shaka\fragmented-bunny.mp4,stream=audio,output=audio.mp4,drm_label=audio
in=c:\code\shaka\fragmented-bunny.mp4,stream=video,output=video.mp4,drm_label=sd
--protection_scheme cenc
--playready_extra_header_data "<LA_URL>https://playready.ezdrm.com/cency/preauth.aspx?pX=EXXXXX</LA_URL>"
--enable_raw_key_encryption
--keys label=AUDIO:key_id=15b9XXXXdb4dXXXXafc01c89XXXXf1e4:key=2aedXXXXecacXXXXefb3dceXXXXa415,label=SD:key_id=15b9XXXXdb4dXXXXafc01c89XXXXf1e4:key=2aedXXXXecacXXXXefb3dceXXXXa415
--pssh 0000003F707373680XXXXXXXXXXXXXXXXXXXXXACEA3C827DCD51D21ED0000001F121015B938C8DB4D4CF7AFC01C8921E3F1E41A05657A64726D48E3DC959B06
--protection_systems Widevine,PlayReady
--mpd_output dash.mpd
```

The example raw key format for Widevine only:

```
packager-win.exe
in=BigBuckBunny_320x180_Frag.mp4,stream=audio,output=audio.mp4,drm_label=AUDIO
in=BigBuckBunny_320x180_Frag.mp4,stream=video,output=h264_360p.mp4,drm_label=SD
--protection_scheme cenc
--enable_raw_key_encryption
--keys label=AUDIO:key_id=15b9XXXXdb4dXXXXafc01c89XXXXf1e4:key=2aedXXXXecacXXXXefb3dceXXXXa415,label=SD:key_id=15b9XXXXdb4dXXXXafc01c89XXXXf1e4:key=2aedXXXXecacXXXXefb3dceXXXXa415
--pssh 0000003F707373680XXXXXXXXXXXXXXXXXXXXXACEA3C827DCD51D21ED0000001F121015B938C8DB4D4CF7AFC01C8921E3F1E41A05657A64726D48E3DC959B06
--protection_systems Widevine
```

```
--mpd_output h264.mpd
```

Note – for Widevine only remove "--protection_systems PlayReady"

Apple FairPlay Streaming

EZDRM Apple FairPlay DRM is a hosted Apple FairPlay Streaming (DRM). This enables a content owner to encrypt the media with Apple FPS DRM keys and deliver content Apple devices with native support MAC Safari browser via HTML 5 player or iOS via native App or Safari 11.3.

The packaging process encrypts the media. This is accomplished via a secure web call to the EZDRM Key Servers API. The Key Server API will return an XML response with the DRM key structure.

Generating Keys

Option 1: Request DRM keys using EZDRM CPIX Web Service

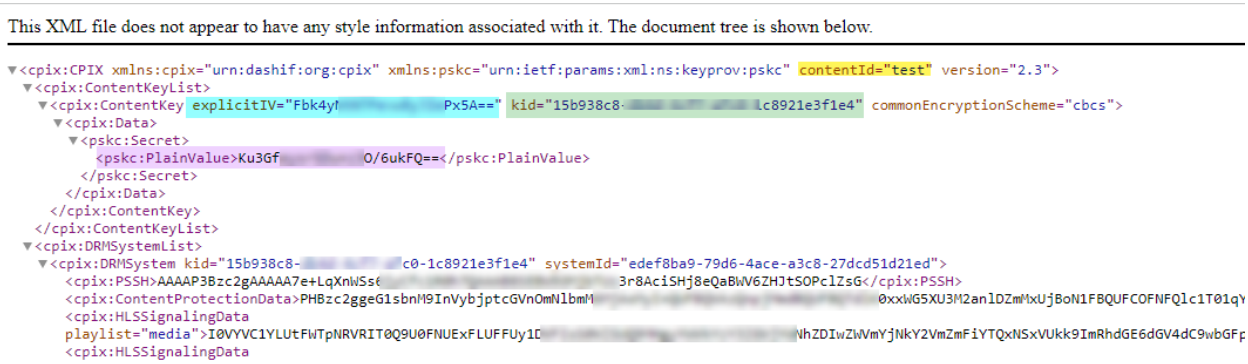
3. Call the EZDRM web service in a browser:
   https://cpix.ezdrm.com/keygenerator/**cpix2**.aspx?**k=kid**&**u=username**&**p=password**&**c=resourcename**

   The parameters are as follows:

Parameter	Description
k	kid or Key ID value (client generated) in GUID format*
u	EZDRM username
p	EZDRM password
c	Content ID – generic resource name/identifier (client generated) – passed into **id** field

   * To generate a GUID for the k value, you can use a GUID generator like the one found here: https://www.guidgenerator.com/.

Here are the descriptions of the key values returned by EZDRM:



- **id** – c value returned, generic resource name/identifier (client generated)
- **explicitIV** – the Apple FairPlay explicit IV value
- **kid** – Key ID in GUID format (client generated)*
- **pskc:Secret key**– the Secret Content Encryption Key in Base 64 generated by EZDRM and returned as a plain value

* To generate a GUID for the k value, you can use a GUID generator like the one found here: https://www.guidgenerator.com/.

Here Is the example XML return:

```
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" contentId="test" version="2.3">
<cpix:ContentKeyList>
<cpix:ContentKey explicitIV="Fbk4XXXXXXXXwByJIePx5A==" kid="15b9XXXX-db4d-XXXX-afc0-1c89XXXXf1e4" commonEncryptionScheme="cenc">
<cpix:Data>
<pskc:Secret>
<pskc:PlainValue>Ku3GfXXXXXXXXz3O/6ukFQ==</pskc:PlainValue>
</pskc:Secret>
</cpix:Data>
```

Option 2: Request DRM keys with curl

The second option to request DRM keys from EZDRM is to script the process with curl or another web service call.

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service.

```
curl -v “https://cpix.ezdrm.com/keygenerator/cpix.aspx?k=kid&u=username&p=password&c=test”
```

Apple FairPlay Streaming – Key Value Definitions

Here are the descriptions of the key values returned:

- **key_id:** The **kid** used for encryption (also known as KID); Base 64 encoded with no dashes (-).

- **key:** The DRM content encryption key (128 bit key); Base 64 encoded.

  For the **key** value use the **pskc:Secret key** value and decode the Plain Value tag from Base 64 to HEX format in lowercase (no dashes). An example decoder can be found at: https://tomeko.net/online_tools/base64.php?lang=en

  pskc:Secret key (Base 64) = Ku3GfXXXXXXu+z3O/6ukFQ==

  (KeyHEX) = 2aedXXXXecacXXXXefb3dceXXXXa415

- **iv: explicitIV** decoded from Base 64 to HEX) combined.

  Decode the **explicitIV** Plain Value Base 64 to HEX format. An example decoder can be found at: https://tomeko.net/online_tools/base64.php?lang=en

  explicitIV (Base 64) = Fbk4yNtNTPevwByJIePx5A==

  IV (HEX no dashes) = 15b938c8db4d4cf7afc01c8921e3f1e4

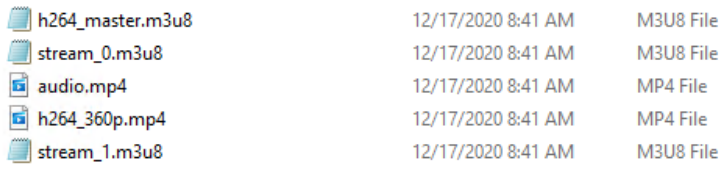
- **KeyURI** - Use the command line option **–hls_key_uri** to specify the license URL for encryption. Build by appending the base URL **"skd://"** with the **kid** value (with dashes) and the **IV (HEX no dashes)** seperated by a colon. For example:
  skd://15b938c8-db4d-4cf7-afc0-1c8921e3f1e4:
  15b938c8db4d4cf7afc01c8921e3f1e4

Apple FairPlay Streaming – Raw Key Format

The example raw key format for Apple HLS:

```
packager-win.exe
in=c:\code\shaka\fragmented-bunny.mp4,stream=audio,output=audio.mp4,drm_label=audio
in=c:\code\shaka\fragmented-bunny.mp4,stream=video,output=video.mp4,drm_label=sd
--protection_scheme cbcs
--enable_raw_key_encryption
--keys label=AUDIO:key_id=15b938c8db4d4cf7afc01c8921e3f1e4:key=2aedXXXXecacXXXXefb3dceXXXXa415,label=sd:ke
y_id=15b938c8db4d4cf7afc01c8921e3f1e4:key=2aedXXXXecacXXXXefb3dceXXXXa415
--protection_systems FairPlay
--iv 15b938c8db4d4cf7afc01c8921e3f1e4
--hls_master_playlist_output h264_master.m3u8
--hls_key_uri skd://15b938c8-db4d-4cf7-afc0-1c8921e3f1e4:15b938c8db4d4cf7afc01c8921e3f1e4
```

Output example:



CBCS/CMAF - Widevine, PlayReady and Apple FairPlay

Below are the steps to create the DRM Keys for CBCS encryption for Bento4 (Open Source) including Widevine, PlayReady and Apple FairPlay Streaming.

CBCS encryption uses PlayReady 4.3 Header. When this encryption is required, an additional flag is available to return when added to the end of the query string, shown in the example as follows*:

https://cpix.ezdrm.com/KeyGenerator/cpix2.aspx?k=kid&u=username&p=password&c=resourcename&EncryptionScheme=cbcs

**CBCS is needed for CMAF*

A query with no EncryptionScheme flag returns CENC using PlayReady 4.0 Header by default.

Generating Keys

Call the EZDRM web service in a browser:
https://cpix.ezdrm.com/KeyGenerator/cpix2.aspx?k=kid&u=username&p=password&c=resourcename&EncryptionScheme=cbcs

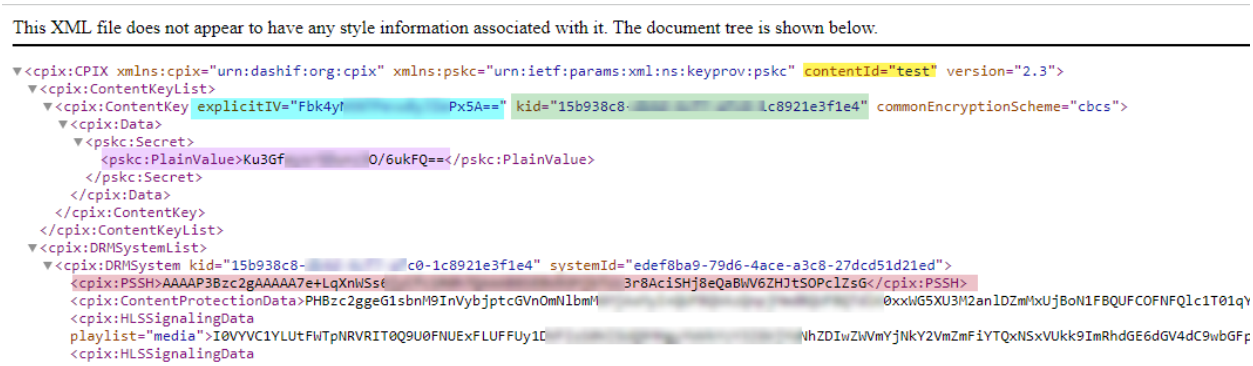
The parameters are as follows:

Parameter	Description
k	kid or Key ID value (client generated) in GUID format*
u	EZDRM username
p	EZDRM password

c	Content ID – generic resource name/identifier (client generated) – passed into **id** field
EncryptionScheme	Encryption using CBCS – PlayReady 4.3 Header

* To generate a GUID for the k value, you can use a GUID generator like the one found here: https://www.guidgenerator.com/.

The response from EZDRM will look like this:



- **contentId** – c value returned, generic resource name/identifier (client generated)
- **explicitIV** – the Apple FairPlay explicit IV value
- **kid** – Key ID in GUID format (client generated)*
- **pskc:Secret key**– the Secret Content Encryption Key in Base 64 generated by EZDRM and returned as a plain value.
- **PSSH** – The modular specific protection system specific header (PSSH) data for the encryption process; Base 64 encoded.

* To generate a GUID for the k value, you can use a GUID generator like the one found here: https://www.guidgenerator.com/.

Here is the example XML return:

```
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" contentId="test" version="2.3">
<cpix:ContentKeyList>
<cpix:ContentKey explicitIV="Fbk4yNtXXXXXXXXyJIePx5A==" kid="15b938c8-XXXX-4cf7-XXXX-1c89XXXXf1e4" commonEncryptionScheme="cenc">
```

```
<cpix:Data>
<pskc:Secret>
<pskc:PlainValue>Ku3GfeXXXXXXX+z3O/6ukFQ==</pskc:PlainValue>
</pskc:Secret>
</cpix:Data>
</cpix:ContentKey>
</cpix:ContentKeyList>
<cpix:DRMSystemList>
<cpix:DRMSystem kid="15b938c8-XXXX-4cf7-XXXX-1c89XXXXf1e4" systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed"
>
<cpix:PSSH>AAAAP3Bzc2gAAAAA7e+LqXnWSs6jyCfc1R0h7XXXXXXXXXX5OMjbTUz3r8AciSHi8eQaBWV6ZHJtSOPclZsG</cpix:PSSH
>
<cpix:ContentProtectionData>PHBzc2ggeG1sbnM9InVybjptcGVnOmNlbmM6MjAxMyI+QUFBQVAzQnpjMmdBQUFBQTdlK0xxWG5XU3
M2anlDZmMxUjBoN1FBQUFCOFNFQlc1T01qYlRVejXXXXXXXXXXjhlUWFCV1Y2WkhKdFNPUGNsWnNHPC9wc3NoPg==</cpix:ContentPro
tectionData>
```

Option 2: Request DRM keys with curl

The second option to request DRM keys from EZDRM is to script the process with curl or another web service call.

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service.

```
curl -v "https://cpix.ezdrm.com/keygenerator/cpix2.aspx?k=kid&u=username&p=password&c=test&EncryptionScheme=cbcs"
```

CBCS/CMAF – Key Value Definitions

- **playready_extra_header_data**: this flag is required for PlayReady. The flag is formatted as follows ending with your EZDRM **pX value**. The **pX value** is branded for your account and is account specific. Your PX value is always the same no matter what packager you are using. It is also the last six characters of your EZDRM Profile ID, as shown in this example:

  Your PlayReady DRM Profile ID is: **4BE3XX9X-XX52-5RT5-87XX-34XXXX123456**

  The PlayReady header data is formatted as follow:

  "<LA_URL>**https://playready.ezdrm.com/cency/preauth.aspx?pX=EXXXXX**</LA_URL>"

- **key_id:** The **kid** used for encryption (also known as KID); Base 64 encoded with no dashes.

- **key:** The DRM content encryption key (128 bit key); Base 64 encoded.

  For the **key** value use the **pskc:Secret key** value and decode the Plain Value tag from Base 64 to HEX format in lowercase (no dashes). An example decoder can be found at:
  https://tomeko.net/online_tools/base64.php?lang=en

  pskc:Secret key (Base 64) = Ku3GfXXXXXXu+z3O/6ukFQ==

  key (HEX) = 2aedXXXXecacXXXXefb3dceXXXXa415

- **PSSH:** The modular specific protection system specific header (PSSH) data for the encryption process; Base 64 encoded.

  For the PSSH value, use the **PSSH** and decode the Plain Value from Base 64 to HEX format. An example decoder can be found at:
  https://tomeko.net/online_tools/base64.php?lang=en

PSSH (Base 64) =
AAAAP3Bzc2gAAAAA7e+LqXnWSs6jyCfc1R0h7XXXXXXXXXX5OMjbTUz3r8AciSHj8eQaBWV6ZHJtSOPclZsG

PSSH (HEX) =
0000003F707373680XXXXXXXXXXXXXXXXXXXXACEA3C827DCD51D21ED0000001F121015B938C8DB4D4CF7AFC01C8921E3F1E41A05657A64726D48E3DC959B06

- **iv: explicitIV** decoded from Base 64 to HEX) combined.

  Decode the **explicitIV** Plain Value Base 64 to HEX format. An example decoder can be found at: https://tomeko.net/online_tools/base64.php?lang=en

  explicitIV (Base 64) = Fbk4yNtNTPevwByJIePx5A==

  IV (HEX no dashes) = 15b938c8db4d4cf7afc01c8921e3f1e4

- **KeyURI** - Use the command line option **–hls_key_uri** to specify the license URL for encryption. Build by appending the base URL **"skd://"** with the **kid** value (with dashes) and the **IV (HEX no dashes)** seperated by a colon. For example:
  skd://15b938c8-db4d-4cf7-afc0-1c8921e3f1e4:
  15b938c8db4d4cf7afc01c8921e3f1e4

CBCS/CMAF – Raw Key Format

The example raw key format for CBCS/CMAF:

```
packager-win.exe
in=c:\code\shaka\fragmented-bunny.mp4,stream=audio,output=audio.mp4,drm_label=audio
in=c:\code\shaka\fragmented-bunny.mp4,stream=video,output=video.mp4,drm_label=sd
--protection_scheme cbcs
--playready_extra_header_data "<LA_URL>https://playready.ezdrm.com/cency/preauth.aspx?pX=EXXXXX</LA_URL>"
--enable_raw_key_encryption
--keys label=AUDIO:key_id=15b938c8db4d4cf7afc01c8921e3f1e4:key=2aedXXXXecacXXXXefb3dceXXXXa415,label=sd:key_id=15b938c8db4d4cf7afc01c8921e3f1e4:key=2aedXXXXecacXXXXefb3dceXXXXa415
--pssh 0000003F707373680XXXXXXXXXXXXXXXXXXXXACEA3C827DCD51D21ED0000001F121015B938C8DB4D4CF7AFC01C8921E3F1E41A05657A64726D48E3DC959B06
--protection_systems Widevine,PlayReady,FairPlay
--mpd_output multi-dash.mpd
--hls_master_playlist_output h264_master.m3u8
--hls_key_uri skd://15b938c8-db4d-4cf7-afc0-1c8921e3f1e4:15b938c8db4d4cf7afc01c8921e3f1e4
--iv 15b938c8db4d4cf7afc01c8921e3f1e4
```

Additional Information

For additional questions and comments please contact: simplify@ezdrm.com