

EZDRM Shaka Packager DRM Keys Guide

Table of Contents

EZDRM Universal DRM	1
Shaka Packager – Overview for Raw Key Format	3
Universal DRM (Widevine & PlayReady)	3
Generating Keys	4
Universal DRM – Key Value Definitions	7
Universal DRM – Raw Key Format	7
Apple FairPlay Streaming	9
Generating Keys	9
Key Value Definitions	10
Apple FairPlay Streaming – Key Value Definitions	12
Apple FairPlay Streaming – Raw Key Format	13
Additional Information	14

Version 2.2 / Updated October 21, 2021

Shaka Packager – Overview for Raw Key Format

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/cpix.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:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" id="drm-001">
  <cpix:ContentKeyList>
    <cpix:ContentKey kid="582a-06ab-ae533-5ef8" explicitIV="WCq8TPMpe+A==">
      <cpix:Data>
        <pskc:Secret>
          <pskc:PlainValue>SUH48UGD18wgc=</pskc:PlainValue>
        </pskc:Secret>
      </cpix:Data>
    </cpix:ContentKey>
  </cpix:ContentKeyList>
  <cpix:DRMSysSystemList>
    <cpix:DRMSysSystem kid="582ab-06ab-41ef-ae533-5ef8" systemId="edef8ba9-c8-27dcd51d21ed">
      <cpix:PSSH>AAAAAdnBzc2gAAAAA7e-c1R0h7QAAAFYIARIQWcQ8DAarQe+gE65TPMpe+Ij0iV0NxoERBYXJRZStnRTY1VFBNCgUrQT09IwldH3hY2tzIjpbI1NEI119KgJTRA=</cpix:PSSH>
      <cpix:ContentProtectionData/>
    </cpix:DRMSysSystem>
    <cpix:DRMSysSystem kid="582ab-06ab-41ef-ae533-5ef8" systemId="9a04f079-c92-e65be0885f95">
      <cpix:PSSH>AAADnBzc2gAAAAA7e+XXXXXXXXXXXXXXXXXXXXAAAFYIARIQWcQ8DAarQe+gE65TPMpe+BoIbw92aWrVbmUiMnsia21kIjoiV0NxoERBYXJRZStnRTY1VFBNCgUrQT09IwldH3hY2tzIjpbI1NEI119KgJTRA=</cpix:PSSH>
      <cpix:ContentProtectionData>Pglzchi6chDvPjVnsUF8QVU...CAAGACBALwBzAGHAA8LAG0AYQ8zAC4ABQ8pAGHAcgBVAHHA8wBMAHQALg8JAG8ABQAVAEQAUGBNAC6
      FHZ0FkQUIwQ8BQU9nQXZBQzhBY3QcakhZ0FauU3QQUdFQUN3QXVBRzBBYVFCakFISUFId016QUc4QV
    </cpix:ContentProtectionData>
    </cpix:DRMSysSystem>
  </cpix:DRMSysSystemList>
</cpix:CPIX>
```

- **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.

* 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" id="drm-001">
  <cpix:ContentKeyList>
    <cpix:ContentKey kid="582aXXXX-XXXX-41ef-XXX-ae533ccaXXXX" explicitIV="WCq8DAXXXXXXE65TPMpe+A==">
      <cpix:Data>
        <pskc:Secret>
          <pskc:PlainValue>SU+XXXXXXqvSN0aXbXXXXX==</pskc:PlainValue>
        </pskc:Secret>
      </cpix:Data>
    </cpix:ContentKey>
  </cpix:ContentKeyList>
  <cpix:DRMSysSystemList>
    <cpix:DRMSysSystem kid="582aXXXX-XXXX-41ef-XXX-ae533ccaXXXX" systemId="edefXXXX-79d6-XXXX-a3c8-27dcd5XXXXXd">
      <cpix:PSSH>AAAAAdnBzc2gAAAAA7e+XXXXXXXXXXXXXXXXXXXXAAAFYIARIQWcQ8DAarQe+gE65TPMpe+BoIbw92aWrVbmUiMnsia21kIjoiV0NxoERBYXJRZStnRTY1VFBNCgUrQT09IwldH3hY2tzIjpbI1NEI119KgJTRA=</cpix:PSSH>
      <cpix:ContentProtectionData/>
    </cpix:DRMSysSystem>
  </cpix:DRMSysSystemList>
</cpix:CPIX>
```

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=drm-001
```

Universal DRM – Key Value Definitions

Widevine

- **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) = sU+A8UXXXvSN0aXXXwcg==



(KeyHEX) = b14fXXX41b77XXX2374XXXe2f30XX

PlayReady

- **key_id**: This value is the same as the Widevine **kid** above, used for encryption (also known as KID); Base 64 encoded with no dashes.
- **key**: This value is the same as the Widevine **Key** above, because the DRM is using common encryption with a shared key base; Base 64 encoded.

Universal DRM – Raw Key Format

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

```
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
--enable_raw_key_encryption
--keys label=AUDIO:key_id=582aXXXXXXXX41efXXXae533ccaXXX:key=b14fXXX41b77XXX2374XXXe2f30XX,label=SD:key
y_id=582aXXXXXXXX41efXXXae533ccaXXX:key=b14fXXX41b77XXX2374XXXe2f30XX
--protection_systems Widevine,PlayReady
--mpd_output h264.mpd
```




```
C:\Users\TEST01\Downloads>
C:\Users\TEST01\Downloads>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 --enable_raw_key_encryption --keys label
=AUDIO:key_id=582aXXXXXXXXXXa5ef8:key=b14f8XXXXXXXXXX072,label=SD:key_id=582aXXXXXXXXXXa5ef8:key=b14f8XXXXXXXXXX
3072 --protection_systems Widevine,PlayReady --mpd_output h264.mpd
[1217/080723:INFO:demuxer.cc(88)] Demuxer::Run() on file 'BigBuckBunny_320x180_Frag.mp4'.
[1217/080723:INFO:demuxer.cc(160)] Initialize Demuxer for file 'BigBuckBunny_320x180_Frag.mp4'.
[1217/080725:INFO:single_segment_segmenter.cc(107)] Update media header (moov) and rewrite the file to 'h264_360p.mp4'.
[1217/080726:INFO:mp4_muxer.cc(177)] MP4 file 'h264_360p.mp4' finalized.
[1217/080726:INFO:single_segment_segmenter.cc(107)] Update media header (moov) and rewrite the file to 'audio.mp4'.
[1217/080726:INFO:mp4_muxer.cc(177)] MP4 file 'audio.mp4' finalized.
Packaging completed successfully.
```

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
--enable_raw_key_encryption
--keys label=AUDIO:key_id=582aXXXXXXXXXX41efXXXae533ccaXXXX:key=b14fXXXX41b77XXXX2374XXXXe2f30XX,label=SD:ke
y_id=582aXXXXXXXXXX41efXXXae533ccaXXXX:key=b14fXXXX41b77XXXX2374XXXXe2f30XX
--mpd_output h264.mpd
```

Note – for Widevine only remove “--protection_systems widevine, PlayReady”

Output example:

 audio.mp4	12/4/2020 10:01 AM	MP4 File	11,989 KB
 h264.mpd	12/4/2020 10:01 AM	MPD File	5 KB
 h264_360p.mp4	12/4/2020 10:01 AM	MP4 File	51,495 KB

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 to Apple devices with native support in 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/cpix.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/>.

Key Value Definitions

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

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" id="hyb-001">
  <cpix:ContentKeyList>
    <cpix:ContentKey kid="000ae000-533cca5ef8" explicitIV="AArgA...?Mpe+A==">
      <cpix:Data>
        <pskc:Secret>
          <pskc:PlainValue>q4sp<GmTQ==</pskc:PlainValue>
        </pskc:Secret>
      </cpix:Data>
    </cpix:ContentKey>
  </cpix:ContentKeyList>
  <cpix:DRMSysSystemList>
    <cpix:DRMSysSystem kid="000ae000-533cca5ef8" systemId="edef8ba9-79c...-27dcd51d21ed">
      <cpix:PSSH>AAAAdnBzc2J...XnW5s6jyCfc1R0h7QAAAFYIARIQAArgAAArQe+...oIbw92akRvbmUiMnsia2lkIjoiQUFYZ0FBYXJRZStnRTY1VFBNcGUrQ'
      <cpix:ContentProtectionData/>
    </cpix:DRMSysSystem>
  </cpix:DRMSysSystemList>
</cpix:CPIX>

```

- **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
- **explicitIV** – the Apple FairPlay explicit IV 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" id="hyb-001">
  <cpix:ContentKeyList>
    <cpix:ContentKey kid="582de60c-XXXX-XXXX-a013-XXX33cca5ef8" explicitIV="wCXXXXXXXXX+gE65TXXXe+A==">
      <cpix:Data>
        <pskc:Secret>
          <pskc:PlainValue>SU+A8XXXXXXXXXXaXbi8wgc==</pskc:PlainValue>
        </pskc:Secret>
      </cpix:Data>
    </cpix:ContentKey>
  </cpix:ContentKeyList>
</cpix:CPIX>

```

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=hyb-001
```

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) = sU+A8UXXXXvSN0aXXXXwcg==



(KeyHEX) = b14fXXXX41b77XXXX2374XXXXe2f30XX

- **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) = WCq8DXXXXX+gE65TXXXX+A==



iv (HEX no dashes) = 582aXXXXXXXXX41efXXXae533ccaXXXX

- **KeyURI** - Use the command line option **--encryption-key-uri** to specify the license URL for encryption. Build by appending the **kid** value to base URL "skd://fps.ezdrm.com/;" for example:
skd://fps.ezdrm.com/:582de60c-XXXX-XXXX-a013-XXX33cca5ef8






Apple FairPlay Streaming – Raw Key Format

The example raw key format for Apple HLS:

```
packager-win.exe
in=BigBuckBunny_320x180_frag.mp4,stream=video,output=video.mp4,drm_label=HD
in=BigBuckBunny_320x180_frag.mp4,stream=audio,output=audio.mp4,drm_label=AUDIO
--protection_scheme cbcs
--enable_raw_key_encryption
--keys label=AUDIO:key_id=582de60cXXXXXXXXa013XXX33cca5ef8:key=b14fXXX41b77XXX2374XXXe2f30XX:iv=582aXXX
XXXX41efXXXae533ccaXXX,
label=HD:key_id=582de60cXXXXXXXXa013XXX33cca5ef8:key=b14fXXX41b77XXX2374XXXe2f30XX:iv=582aXXXXXXXX41efX
XXae533ccaXXX
--protection_systems FairPlay
--hls_master_playlist_output h264_master.m3u8
--hls_key_uri skd://fps.ezdrm.com/;582de60c-XXXX-XXXX-a013-XXX33cca5ef8
```

```
C:\Users\TEST01\Downloads>packager-win.exe in=BigBuckBunny_320x180_frag.mp4,stream=video,output=video.mp4,
drm_label=HD in=BigBuckBunny_320x180_frag.mp4,stream=audio,output=audio.mp4,drm_label=AUDIO --protection_s
cheme cbcs --enable_raw_key_encryption --keys label=AUDIO:key_id=708b1a811d1fd:iv=708b1a811d1fd:key=e651
9be90545:iv=708b1a811d1fd,label=HD:key_id=708b1a811d1fd:iv=708b1a811d1fd --protection_systems Fair
Play --hls_master_playlist_output h264_master.m3u8 --hls_key_uri skd://fps.ezdrm.com/;708b1a811d1fd:iv=708
c45-1fd
[0721/072305:INFO:demuxer.cc(88)] Demuxer::Run() on file 'BigBuckBunny_320x180_frag.mp4'.
[0721/072305:INFO:demuxer.cc(160)] Initialize Demuxer for file 'BigBuckBunny_320x180_frag.mp4'.
[0721/072306:INFO:single_segment_segmenter.cc(107)] Update media header (moov) and rewrite the file to 'vi
deo.mp4'.
[0721/072306:INFO:mp4_muxer.cc(177)] MP4 file 'video.mp4' finalized.
[0721/072306:INFO:single_segment_segmenter.cc(107)] Update media header (moov) and rewrite the file to 'au
dio.mp4'.
[0721/072306:INFO:mp4_muxer.cc(177)] MP4 file 'audio.mp4' finalized.
Packaging completed successfully.
```

Output example:

 h264_master.m3u8	12/17/2020 8:41 AM	M3U8 File
 stream_0.m3u8	12/17/2020 8:41 AM	M3U8 File
 audio.mp4	12/17/2020 8:41 AM	MP4 File
 h264_360p.mp4	12/17/2020 8:41 AM	MP4 File
 stream_1.m3u8	12/17/2020 8:41 AM	M3U8 File

Additional Information

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