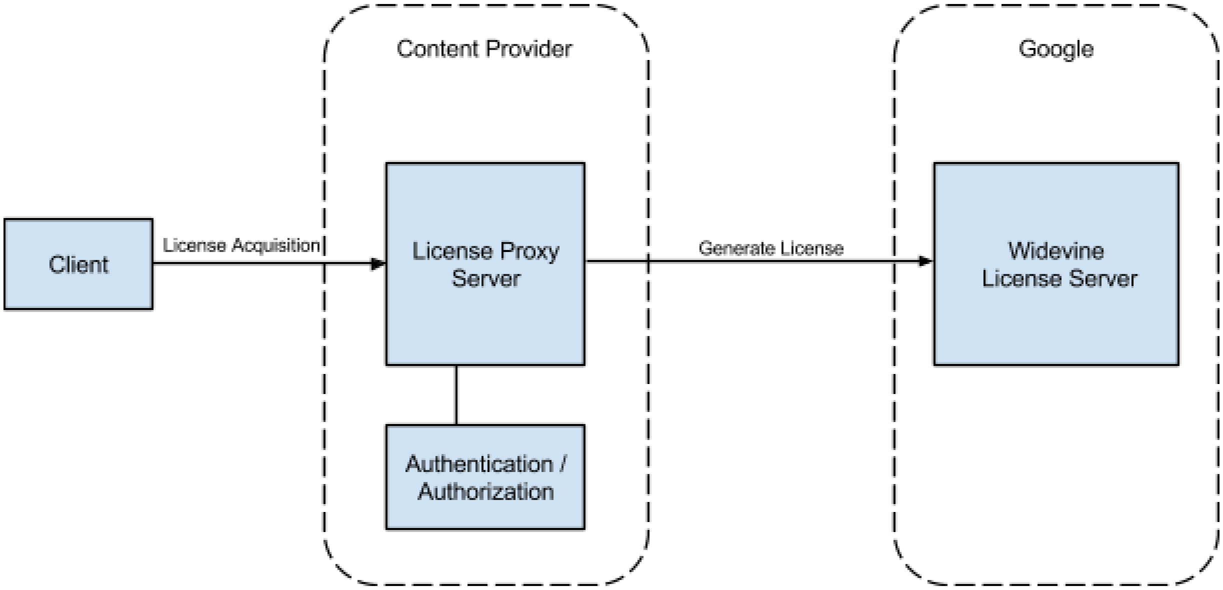
Widevine DRM
Proxy Integration

version 2.6

Introduction

The document describes how external content providers integrate their license proxy component with the Widevine Cloud License Service. This solution is intended for content providers who have their own proxy servers for user authentication and access authorization, but relies on Widevine service for policy and content key managements and generating the actual license.

System Diagram



Workflow

1. A client generates a license request and sends it to the license proxy.
2. The proxy authenticates the user and authorizes the request.
3. The proxy determines the desired policy (if there is any) and allowed track types.
4. The proxy calls GenerateLicense API (using HTTPS POST).
5. Widevine License service receives the GenerateLicense request
    a. obtains content keys
    b. generates a license
    c. returns license to the proxy.
6. The proxy inspects GenerateLicense response status and forwards the license blob to client.

GenerateLicense API

This API allows a Content Provider proxy to fetch a Modular DRM license.

Request

The request is an HTTP POST request to the GenerateLicense URL. The POST body is JSON formatted data in the following syntax:

```
        {
                “request”: “<message>”,
                “signature”: “<signature>”
                “signer” : <signer name>”
        }
```

Name	Value	Description
request	Base64 encoded string.	The actual message. This is a serialized JSON message containing the request or the response. The message is in the clear.
signature	Base64 encoded string.	AES CBC encryption of the SHA1 hash of <message> Credentials are provided by Widevine. Please contact us for more details.
signer	string	Identifies the entity sending the message. Credentials are provided by Widevine. Please contact us for more details.

<message> is a JSON formatted string containing the license challenge, policy overrides and license configurations. The syntax is

```
        {
                “payload”: “<license challenge>”,
                “content_id”: “<content id>”
                “provider”:”<provider>”
                “allowed_track_types”: “<types>”,
                “content_key_specs”: [
                        { “track_type”: “<track type 1>” },
                        { “track_type”: “<track type 2>” }, … ],
                “policy_overrides”: {
                        “can_play”: <can play>,
                        “can persist”: <can persist>,
                        “can_renew”: <can renew>,
                        “rental_duration_seconds”: <rental duration>,
                        “playback_duration_seconds”: <playback duration>,
                        “license_duration_seconds”: <license duration>,
                        “renewal_recovery_duration_seconds”: <renewal recovery duration>,
                        “renewal_server_url”:”<renewal server url>”,
                        “renewal_delay_seconds”:<renewal delay>,
                        “renewal_retry_interval_seconds”:<renewal retry interval>,
                        “renew_with_usage”:<renew with usage>
                }
        }
```

Fields below are optional unless marked as required.

Name	***Value***	***Description***
payload	Base64 encoded string	**REQUIRED** The license request sent by a client.
provider	string	**REQUIRED** Used to look up content keys and policies. Credentials are provided by Widevine. Please contact us for more details.
content_id	Base64 encoded string max 36 bytes	This is the content identifier. Used to associate KeyId(s) and Content Key(s) for each content_key_specs.track_type. Note: convert your string to base64.
policy_name	string	Name of a previously registered policy. This field is provided for backwards compatibility with legacy Widevine systems. As a general recommendation, all business rules should be specified in the license request according to this document.
allowed_track_types	enum	SD_ONLY or SD_HD. Controls which content keys should be included in a license.
content_key_specs	array of JSON structures	A finer grained control on what content keys to return.

	(see Content Key Spec)	See Content Key Spec below for details. Only one of allowed_track_types and content_key_specs can be specified.
use_policy_overrides_exclusively	boolean. true or false	Use policy attributes specified by policy_overrides and omit all previously stored policy.
policy_overrides	JSON structure (see Policy Overides)	Policies settings for this license. In the event this asset has a pre-defined policy, these specified values will be used.
session_init	JSON structure (see Session Initialization)	Optional data passed to license.
parse_only	boolean. true or false	The license request is parsed but no license is issued. However, values from the license request are returned in the response.
session_key	Base64 encoded string.	If specified, the content keys and IVs specified in ContentKeySpec are encrypted with this key using AES-CBC (PKCS5 padding). The session_key itself is encrypted with the partner's AES key that was provided by Widevine.
session_iv	Base64 encoded string.	**REQUIRED** *if session_key was specified.* If specified, the content keys and IVs specified in ContentKeySpec are encrypted with this IV using AES-CBC (PKCS5 padding). The session_iv itself is encrypted with the partner's AES key that was provided by Widevine.

`license_start_time` is the time when the license is being requested. Playback is halted when `license_start_time` + `license_duration_seconds` is exceeded.

Policy Overrides

Fields below are optional unless marked as required.

Name	***Value***	***Description***
policy_overrides. can_play	boolean. true or false	**REQUIRED** Indicates that playback of the content is allowed. *Default = false.*
policy_overrides. can_persist	boolean. true or false	Indicates that the license may be persisted to non-volatile storage for offline use. *Default = false*.
policy_overrides. can_renew	boolean true or false	Indicates that renewal of this license is allowed. If true, the duration of the license can be extended by heartbeat. *Default = false.*
policy_overrides. license_duration_seconds	int64	Indicates the time window for this specific license. A value of 0 indicates unlimited. *Default = 0*. See Appendix.
policy_overrides. rental_duration_seconds	int64	***DEPRECATED*** Indicates the time window while playback is permitted. A value of 0 indicates unlimited. *Default = 0.* See Appendix.
policy_overrides. playback_duration_seconds	int64	The viewing window of time once playback starts within the license duration. A value of 0 indicates unlimited. *Default = 0.* See Appendix.
policy_overrides. renewal_server_url	string	All heartbeat (renewal) requests for this license shall be directed to the specified URL. This field is only used if can_renew is true.

policy_overrides.renewal_delay_seconds	int64	How many seconds after license_start_time, before renewal is first attempted. This field is only used if can_renew is true. *Default = 0.*
policy_overrides.renewal_retry_interval_seconds	int64	Specifies the delay in seconds between subsequent license renewal requests, in case of failure. This field is only used if can_renew is true. *Default = 0.*
policy_overrides.renewal_recovery_duration_seconds	int64	The window of time, in which playback is allowed to continue while renewal is attempted, yet unsuccessful due to backend problems with the license server. A value of 0 indicates unlimited. This field is only used if can_renew is true. *Default = 0.*
policy_overrides.renew_with_usage	boolean true or false	Indicates that the license shall be sent for renewal when usage is started. This field is only used if can_renew is true. *Default = false.*
policy_overrides.always_include_client_id	boolean true or false	Indicates to client that license renewal and release requests must include client identification (client_id). *Default = false.*

Session Initialization

This structure is used to pass optional data when issuing a license.

Fields below are optional unless marked as required.

Name	***Value***	***Description***
provider_client_token	Base64 encoded string	Client token to send back in the license response. If the license request contains a client token, this value is ignored. The client token will persist beyond license sessions.
override_provider_client_token	boolean. true or false	If false and the license request contains a client token, use the token from the request even if a client token was specified in this structure. If true, always use the token specified in this structure. *Default = false.*

session_id	Base64 encoded string	Identifies the session. This value will exist in all **subsequent license renewals** associated with this license.

Content Key Specs

If a pre-existing policy exists, there is no need to specify any of the values in the Content Key Spec. The pre-existing policy associated with this content will be used to determine the output protection such as HDCP and CGMS. If a pre-existing policy is not registered with the Widevine License Server, the content provider can inject the values into the license request (recommended approach).

Each content_key_specs must be specified for all tracks, regardless of the option *use_policy_overrides_exclusively*.

Fields below are optional unless marked as required.

Name	***Value***	***Description***
content_key_specs.track_type	string String options: SD HD AUDIO	A track type definition. If content_key_specs is specified in the license request, you must specify **all track types explicitly**.
content_key_specs. security_level	uint32	Defines client robustness requirements for playback. 1 - Software-based whitebox crypto is required. (SW_SECURE_CRYPTO) 2 - Software crypto and an obfuscated decoder is required. (SW_SECURE_DECODE) 3 - The key material and crypto operations must be performed within a hardware backed trusted execution environment. (HW_SECURE_CRYPTO) 4 - The crypto and decoding of content must be performed within a hardware backed trusted execution environment. (HW_SECURE_DECODE) 5 - The crypto, decoding and all handling of the media (compressed and uncompressed) must be handled within a hardware backed trusted execution environment. (HW_SECURE_ALL) *Default = 1.*

		See Appendix.
content_key_specs. required_output_protection. hdcp	string - one of: HDCP_NONE, HDCP_V1, HDCP_V2	Indicates whether HDCP is required. *Default = HDCP_NONE.* *Note: These must be capitalized, lowercase will result in a malformed request.* See Appendix.
content_key_specs. required_output_protection. cgms_flags	string - one of: CGMS_NONE, CGMS_FREE, CGMS_ONCE, CGMS_NEVER	Indicates whether HDCP is required. *Default = CGMS_NONE.* *Note: Do not specify this parameter for Chrome browser.* *Note: These values must be capitalized, lowercase will result in a malformed request.*
content_key_specs.key	Base64 encoded string	Content key to use for this track. If specified, the track_type **or** key_id is required. This option allows the content provider to inject the content key for this track instead of letting Widevine license server generate or lookup a key. Note: convert your key value to binary and base64.
content_key_specs.iv	Base64 encoded string	IV associated with the content key for this track.
content_key_specs.key_id	Base64 encoded string binary, 16 bytes	Unique identifier for the key. Note: convert your string or hex value to binary and base64.

Example of specifying and protecting content_key_specs

ContentKeySpecs (request)

[
  {
    "key_id":"MGRmMjVjYzUtOWFkMS01MjhiLTkwZGMtNTYyY2QxODExODNi",
    "track_type":"SD",

```
            "key":"Pv38THb6rTQbIpzg04/qag==",
            "iv":"REVBREJFRUZERUFEQkVFRg=="
          },
          {
            "key_id":"NzVlYThiYWMtZjg3Yy01YzMxLWFmYTYtNWY3YmUwNjE1MGU5",
            "track_type":"HD",
            "key":"xiE14IkMIRKxSPBQL+HksA==",
            "iv":"REVBREJFRUZERUFEQkVFRg=="
          },
          {
            "key_id":"NDczMWNhNzItOWExYy01YjQ1LThjMTMtMGY4MDJiMDRiMWNj",
            "track_type":"AUDIO",
            "key":"y9DDm30ttT07GqXfxKMwdQ==",
            "iv":"REVBREJFRUZERUFEQkVFRg=="
          }
]
```

License Response

The response is in JSON format.

```
{
        “operation_status_code”: <status code>,
        “licenses”: <license>,
        “license_medata”: [
        {
                “drm_status_code”: <drm status code>,
                “asset_id”:<asset identifier>,
                “policy”: {
                        “organization_name”:”<organization>”,
                        “name”:”<policy name>”,
                        “track_policy”: [
                        {
                                “track_type”:”<track type>”,
                                “level”:<security level>,
                                “output_protection”:”<output protection>”
                        }
                        … ],
                }
        }
   ...]
}
```

Name	***Value***	***Description***
status	uint32	Result of the request.
license	Base64 encoded string	License response that needs to be passed to the CDM after Base64 decode.
license_metadata	JSON structure (see License_Metadata)	License attributes.
make	string	Manufacturer of the device making the license request.
model	string	Model of the device making the license request.
security_level	uint32 (see Client Security Level)	Widevine-defined security level as referenced within the EME specification.
drm_cert_serial_number	string	Globally unique serial number of certificate associated with this device. Example: 7f6f3150b373262cbdc3b2dade075dbb
message_type	string - one of: LICENSE, SERVICE_CERTIFICATE	Indicates the type of message contained in the license going back to the CDM. This field may be missing or empty if the server failed to generate a license or certificate.

License_Metadata

Name	***Value***	***Description***
content_id	string	content id used to generate the license.
license_type	uint32	1 - indicates streaming. 2- indicates offline.
request_type	uint32	1 - indicates NEW license. 2 - indicates RENEWAL license. 3 - indicates RELEASE license.

Rejecting a request

A proxy may reject a license request if, for example, a user is not allowed to watch the video. In this case, a proxy may just return an empty HTTP response with appropriate response code, e.g. 403 Forbidden, or whatever response that the content provider's client application wishes to receive. No license blob is expected by the Widevine client module in this case.

Heartbeat (License Renewal) support

A proxy uses the same GenerateLicense API to obtain a heartbeat response. There is no need to pass in allowed_track_types and content_key_specs. A proxy controls whether the playback is allowed to continue by setting policy.can_play. The license renewals do not contain content keys, they only extend the duration of keys contained in the initial license.

- Related documentation
  - License Renewal Overview
  - Using License Renewal

Protecting proxy specified content keys

If the proxy is specifying content keys to the Widevine license service, it’s strongly recommended the proxy encrypts the content keys. The process for the proxy is as follows:

1. Generate a random 16 byte session key and 16 byte session IV.
    - Example session key: `0x8727a40ebd82329e6b3b4e29fa3b004b`
    - Example session IV: `0x000102030405060708090a0b0c0d0e0f`
2. Encrypt each content key and content IV (AES-CBC, PKCS5 padding) using the session key and session IV. Example of using the following clear content key and iv.
    - Clear content key: `0x2038fa82910c404bee9200e8108baf29`
    - Clear content IV: `0x00112233445566778899aabbccddeeff`

    Encrypting the content key and iv with the session key/iv, results are:
    - Encrypted content key: `0x228107d15be6f2b0fc4945a3c7b2b0bac22a9bc0b154c1f19d181916ea008da5`
    - Encrypted content IV: `0x63a53152184e64f640992b7c6efa90c75890eaffd77200c0aadca51d9eab268a`
3. Encrypt the session key and session IV (AES-CBC, PKCS5 padding) using the provider’s AES signing key/iv that is registered with Widevine. In this example, we’ll use widevine_test signing key:
    - Widevine_Test (32 byte) signing key: `0x1ae8ccd0e7985cc0b6203a55855a1034afc252980e970ca90e5202689f947ab9`
    - Widevine_Test IV: `0xd58ce954203b7c9a9a9d467f59839249`

    Encrypting the session key and IV with the signing key, results are:
    - Encrypted session key: `0x2895244b8f656616c2813de6ebe6d3f55b803d5ce2f51ff1f78c2d4365f2aa1f`
    - Encrypted session IV: `0x1c1f219112972db8adb58e15d4ee95a7f9dbd813c308083798c275be939070ae`

4. Specify the Base64 encoded encrypted (binary) session key as the value to “`session_key`” in the JSON license request.
    - `session_key:KJUkS49lZhbCgT3m6+bT9VuAPVzi9R/x94wtQ2Xyqh8=`
5. Specify the Base64 encoded encrypted (binary) session IV as the value to “`session_iv`” in the JSON license request.
    - `session_iv:HB8hkRKXLbittY4V1O6Vp/nb2BPDCAg3mMJ1vpOQcK4=`

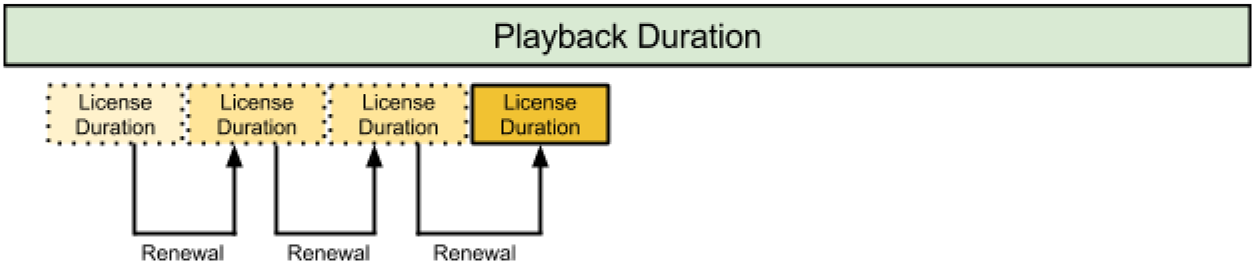
Status Codes

Code	***Description***
OK	Success.
SIGNATURE_FAILED	The server was unable to authenticate the message. Perhaps: ● The specified <signer> is unknown ● The <signature> was not specified, but is required in this case.
INVALID_LICENSE_CHALLENGE	The license challenge blob generated by client is invalid.
INVALID_WIDEVINE_HEADER	The Widevine header embedded in an encrypted asset is invalid.
POLICY_UNKNOWN	The specified <policy> was not found for this <signer>.
MALFORMED_REQUEST	The request is not formatted correctly.
INTERNAL_ERROR	Widevine server internal error.
INVALID_REQUEST	The request is not recognized, or is missing a required field.

Appendix

License duration relationships

The diagram below illustrates the relationship and hierarchy of the duration parameters.



License duration governs the length of time allowed for the decryption and display of content media. It indicates how long the existing license is valid.

The playback duration value is optional.
If playback duration is unspecified, it is unlimited.

The additional durations provide additional business logic:

- To enable support of license renewals.
  - With each successful renewal, the license duration is extended.
  - With continued successful renewals, the license duration can be extended to encompass the length of the media.
  - A renewal failure would result in premature license expiration.
- To provide a limited time period in which a license is valid.

Client security level (content_key_specs.security_level)

A license may be issued with a requirement for the minimum client security level.

The table below illustrates the general mapping between the EME security level definitions and Widevine device robustness levels.

Definition	EME Level	Widevine Device Security Level
SW_SECURE_CRYPTO	1	3
SW_SECURE_DECODE	2	3
HW_SECURE_CRYPTO	3	2
HW_SECURE_DECODE	4	1
HW_SECURE_ALL	5	1

Note: For CrOS (Chromebooks), you will need to specify security_level=3 or higher to enable Widevine L1 support. Otherwise, the device will default to Widevine L3.

HDCP Output Matrix

The table below is a results matrix for setting output protection HDCP_V1 in the license request.

`Error` indicates playback will fail.

HDCP v1 Required	No External Display	HDCP-compliant External Display	Non-HDCP -compliant External Display	Analog External Display	ChromeCast	AirPlay
ChromeOS ARM	OK	OK	Error	N/A	Error	N/A
ChromeOS x86	OK	Error	Error	N/A	Error	N/A
Linux	Error	Error	Error	N/A	OK	N/A
Mac	OK	Error	Error	N/A	OK	Error
Windows	OK	OK	Error	N/A	OK	N/A

Sample Request (with signing)

Sending a licensing request to Widevine service:

```
wget -O /tmp/license --post-data
'{"request":
"ewogICJwYXlsb2FkIiA6ICJDQUVTaEFFS1RBZ0FFa2dBQUFBQ0FBQVFXUFhiaHRiL3E0M2YzU2Z1QzJ
WUDNxMGplQUVDVzNlbVFrV24yd1hDWVZPbnZsV1BETnFoOFZWSUI0R21zTkE4ZVZWRmlnWGtRV0lHTjB
HbGdNS2pwVUVTTEFvcUNoUUlBUklRSk1QQ3psMmJWaXlNUUV0eUsvZ3RtUkFCR2hBeU5XWTNPRE16TVR
jeU1tSmpNMkV5R0FFZ3Y1aVFrQVVhSUMzT04xelZnZVYwclA3dzJWbVZMR29ycUNsY01RTzRCZGJIUHl
rM0dzblkiLAogICJwcm92aWRlciIgOiAid2lkZXZpbmVfdGVzdCIsCiAgImNvbnRlbnRfaWQiOiAiWm1
0cU0yeHFZVk5rWm1Gc2EzSXphZz09IiwKICAiY29udGVudF9rZXlfc3BlY3MiOiBbCiAgICB7ICJ0cmF
ja190eXBlIjogIlNEIiB9LAogICAgeyAidHJhY2tfdHlwZSI6ICJIRCIgfSwKICAgIHsgInRyYWNrX3R
5cGUiOiAiQVVESU8iIH0KICBdCn0K","signature":
"xPkAbb3tjOY/ybdz0tmJMq9erH9ILnS5natMZr3QEW8=","signer": "widevine_test" }'
https://license.uat.widevine.com/cenc/getlicense/widevine_test
```

The signature is generated as follows.

1. Take the raw value of the “request” field. Compute its SHA1 hash:
   `xu5tN6lK+EEwMGA90tTbsP5upNY=`
2. Encrypt the hash with AES CBC. The result is the signature. Convert it to base64 encoding and put it in the “signature” field.
   For integration tests, you can use “widevine_test” as the signer. Its AES key is
   ```
   0x1a, 0xe8, 0xcc, 0xd0, 0xe7, 0x98, 0x5c, 0xc0,
   0xb6, 0x20, 0x3a, 0x55, 0x85, 0x5a, 0x10, 0x34,
   0xaf, 0xc2, 0x52, 0x98, 0x0e, 0x97, 0x0c, 0xa9,
   0x0e, 0x52, 0x02, 0x68, 0x9f, 0x94, 0x7a, 0xb9
   ```
   The IV is
   ```
   0xd5, 0x8c, 0xe9, 0x54, 0x20, 0x3b, 0x7c, 0x9a,
   0x9a, 0x9d, 0x46, 0x7f, 0x59, 0x83, 0x92, 0x49
   ```

Expected Response

```
{"status":"OK","license":"CAIS6AIKKgoQMjVmNzgzMzE3MjJiYzNhMhIQMjVmNzgzMzE3MjJiYz
NhMhoAIAEoABIGCAEQARgBGkYSELuvgFd3JF2KbY8Gr8wkSREaMNhP8g6Hn4Rdp7yG6F1/Fey5ZzNTOJ
0+G4IBaTQRg7L2HjGMK8CE6kjR4B9weSp7syABGkoKEAKUuVmddV3iu/D9yj+l6rcSEC46Gde9xaQN12
q/Fgqv250aICR+Q+j0flfRYMdgbgC0jS+6RqFdXvvg94h5ZMKdZEU1IAIoARpKChDrZ2q7yzRelrvPYW
Yw8aPaEhA2lTBnCeoVTeQ5bxydasrqGiDhgFnPjqr5W4S04cwzJaIlwMe0QJaakRZJ461DhIvvfiACKA
EaSgoQY52oDPI7VfO4yrP2TPpd9hIQGmUhfWKWnSgDGJjl9P/Lyhog6oKAeGNHF9AjeVJU47riqgKZh0
6qdMurDfKFywUMXUcgAigBIL+YkJAFOAAaIHP6Qy564dbSl+/ORuMlYm5U4CzSdtLBZkBn4W5V+jTH",
"license_metadata":{"content_id":"ZmtqM2xqYVNkZmFsa3Izag==","license_type":"STRE
AMING","request_type":"NEW"},"supported_tracks":[{"type":"SD","key_id":"ApS5WZ11
XeK78P3KP6Xqtw=="},{"type":"HD","key_id":"62dqu8s0Xpa7z2FmMPGj2g=="},{"type":"AU
DIO","key_id":"Y52oDPI7VfO4yrP2TPpd9g=="}],"make":"","model":"","internal_status
":0,"session_state":{"license_id":{"request_id":"MjVmNzgzMzE3MjJiYzNhMg==","sess
ion_id":"MjVmNzgzMzE3MjJiYzNhMg==","purchase_id":"","type":"STREAMING","version"
:0},"signing_key":"cEeLYQ+ZjldTOEnFy4e66+yhjo4jg6mMNiQqpEk0gBo=","keybox_system_
id":4184,"license_counter":0},"cert_serial_number":"","device_whitelist_state":"
DEVICE_WHITELISTED","message_type":"LICENSE","platform":"","device_state":"RELEA
SED","pssh_data":{"key_id":["JMPCzl2bViyMQEtyK/gtmQ=="],"content_id":""}}
```

Sample Request (injecting content keys)

This example includes content_key_specs. The JSON message before Base64 encoding is:

```
{
   "allowed_track_types":"SD_HD",
   "content_key_specs":[
      {
         "key_id":"MGRmMjVjYzUtOWFkMS01MjhiLTkwZGMtNTYyY2QxODExODNi",
         "track_type":"SD",
         "key":"Pv38THb6rTQbIpzg04/qag=="
      },
      {
         "key_id":"NzVlYThiYWMtZjg3Yy01YzMxLWFmYTYtNWY3YmUwNjE1MGU5",
         "track_type":"HD",
         "key":"xiE14IkMIRKxSPBQL+HksA=="
      },
      {
         "key_id":"NDczMWNhNzItOWExYy01YjQ1LThjMTMtMGY4MDJiMDRiMWNj",
         "track_type":"AUDIO",
         "key":"y9DDm30ttT07GqXfxKMwdQ=="
      }
   ],
   "use_policy_overrides_exclusively":true,
   "provider":"widevine_test",
   "content_id":"WU9VVFVCRTpDd3E3bWVXZWM3MA==",
   "policy_overrides":{
      "license_duration_seconds":36000,
      "can_play":true
   },
   "payload":"<bytes-intentionally removed>"
}
```

```
wget -O /tmp/license --post-data '{"signer": "widevine_test", "request":
"eyJhbGxvd2VkX3RyYWNrX3R5cGVzIjogIlNEX0hEIiwgImNvbnRlbnRfa2V5X3NwZWNzIjogW3sia2V
5X2lkIjogIk1HUm1NalZqWXpVdE9XRmtNUzAxTWpoaUxUa3daR010TlRZeVkyUXhPREV4T0ROaSIsICJ
0cmFja190eXBlIjogIlNEIiwgImtleSI6ICJQdjM4VEhiNnJUUWJJcHpnMDQvcWFnPT0ifSwgeyJrZXl
faWQiOiAiTnpWbFlUaGlZV010WmpnM1l5MDFZek14TFdGbVlUWXROV1kzWW1Vd05qRTFNR1U1IiwgInR
yYWNrX3R5cGUiOiAiSEQiLCAia2V5IjogInhpRTE0SWtNSVJLeFNQQlFMK0hrc0E9PSJ9LCB7ImtleV9
pZCI6ICJORGN6TVdOaE56SXRPV0V4WXkwMVlqUTFMVGhqTVRNdE1HWTRNREppTURSaU1XTmoiLCAidHJ
hY2tfdHlwZSI6ICJBVURJTyIsICJrZXkiOiAieTlERG0zMHR0VDA3R3FYZnhLTXdkUT09In1dLCAidXN
lX3BvbGljeV9vdmVycmlkZXNfZXhjbHVzaXZlbHkiOiB0cnVlLCAicHJvdmlkZXIiOiAid2lkZXZpbmV
fdGVzdCIsICJjb250ZW50X2lkIjogIldVOVZWRlZDUlRwRGQzRTNiV1ZYWldNM01BPT0iLCAicG9saWN
5X292ZXJyaWRlcyI6IHsibGljZW5zZV9kdXJhdGlvbl9zZWNvbmRzIjogMzYwMDAsICJjYW5fcGxheSI
6IHRydWV9LCAicGF5bG9hZCI6ICJDQUVTeGdzSzR3b0lBUkx1Q1FxeEFnZ0NFaE5EYUhKdmJXVkRSSRTB
0VEdsdWRYZ3RlRGcyR05Db2pvNEZJbzRDTUlJQkNnS0NBUUVBNXJiVkFoYThBbVZLMUI3OWlPMGkzeU1
5TFJCMGNIVUJZRzBTT3JtS2NpcWVkbms1ZSsrM3lWeHBvWHR4UzhWTVhIeUpEYUVKZEQwV1pVZWl3Szh
aQjcxQ0ZpUlcwUEhMbDZCTThXeFdtcW1KbzlmdmNndUt2WVF3dlQyMkF6bGd5L2dnUVN5aUc0MFBOSEU
3ZWVVcnZESWJ5OVVvWEtWdmQzczJMVXVVQllBOGxtd1RWaktqM0ErYmZXTmtiODZKNjU3djF0bmxNbmN
WNUkxK1FKZnMrcnB2eXRYbEIrNG40c3M0NTRUNElBMGJmN0dKSHJYeXBJWVZOZERqeTVRaXU2c2UxV1V
JVFJGZlR5ZjBYVTJzU3NyLzJ2VHl3NlVCdUp5V0U5V2tpd29tdFdPei9JL0pUU3g0eVAvWkxnelpLVEV
MK2Z6Z2VNdFp6ZFRyMW1OQjBRSURBUUFCS05jZ0VvQUNnUU1xY05MRHBncTEvcS91Y0t3UmovdFRjc0J
yMkNPSmRSMnhJTDVlYXA4Nmo3UXNZemwxRnRJYWZQY1lzMkRWTDI4VDJ4SnBaQUd2VDlMK3kvb1FwaEU
xNFc2amFIYlQrNW9SNDFZcGtuaUlZbGxqSU5aZU5oVkc4ZzIyTUhNRGg4bjBtNVZPNmk2bTBFUFU4aTN
acUY3UlBveXY0THJSNjdtTzR3NjFMbGFkR1lhejQvdXoyRVdIbDNEVW9sN2hYT2hCVHNpOWZOc2VNZG5
ycC83TFZaOHBTUThEeDFmaGVDZmlQSERzcHArMzlvTXFQcHdFdGdrcmF4MnNSQ3hDc2ZyMFNhRGZNWmN
2aGtIbWhVRTdBM1hmTEhrREJqUmhvS00zMFFXTjE0M1FLei9mWCtxWUFBOXJvS2xnZEl4TTAxWkNBMFR
HZE1GN0pIVkRna1hLS0JxMEJRcXVBZ2dCRWhDRmVOVWs4RUxrMXl4RVpNTVpnOTRhR0wya2pvNEZJbzR
DTUlJQkNnS0NBUUVBNkRXemNocDJwakFrY0U5bWVPT2p0ZmxTNXNPQnRoZGZKSVI3T21QL1ZzTVAvMzR
JL2F5WWt6Yjh0Tm42TXJqQkZHL1I2VTM0dUlrV00ybGY1SldHQ3hZTlVQbVhDc0hXVDdvb3lFQkVHR1J
uczIzWDREEcEt4ODVEajhEcVdrR2VGOHBJQldwa01KaEk0V3dLNkJoMCtJSng2UGJyVStzYjNYUTRqcHB
VWm55ci9nWVZRNkRXRmJ5KzFUWXNKN0RsUDBHTWZ0Z0FwTUFSWWtQa1NlY2VwSlBrU01pUGRzSllvQUV
YK2lIMGFnWkZBZmkyYjRsTThxUmJBd2xrTUxNQ0E4aXdQU0hlWklKTnpIc1E4WDhxMFRkdFFlb3VveXl
VRndrdnBPeEtiWUpGaXl6UVF3ZURib0xWVm9JK1ZZN3VyVFdaOWd2V2U1eTRQR3phQjFJeDBRSURBUUF
CS05jZ0VvQURuYy9nK2JqSlZjOWcxZG9PZzVScENNST0g0ZnNFM0NyR3BzZ0VBSnA2NFdrVTFsQk9GdXk
vQ1YyY1Z4YVVMWlQ3cEJLM2c5NU1qSVg3OG40eGhPWDM5SVR5a3Juc09WdTNNWFA4Mmo4NkZhV3lzbEJ
yck5KOHVhZ0xwVGQra3ExbHRZNVZZhNk8yV3BFWDNjeXA1aFdDampDTFUySDAxT3FONWZNWG4zSHByMFd
qV05CQmF6cnExYWY3Y3pUbkJ4VEpBUTFzWWVYZEo5cVAwZHFDaktCY0FBR2tzczVxWXJZK053Q2hDd1Q
2R3REemRmaVVsSlZOWVFIamNSdTlBdEJzRFJUZy9zUVRWYW9Nc0ZoaVM1c3BYaEloNXd6UTNwdXFSbFN
EN01KcHBrc3llbVRwZm0vZmJuOVZ4cGVnNC9oemh3aTZ1Z3c0Qk8wdFN3dWRSdFNaNW5SWm5Dd0RmSWg
rUHJpWVpCQWFvSlpzajgrMjhpSEJKc1ZNdDkvNUI3TFN2SkIrSTJUR2FWSGNpdStEZXQ2VWZla3VrZXd
tanB3eDRUR1B0cVRJLzVaVXV5N0VNNnJRdUxCUlcvNDB6UWRxWDNuZEhqWVpDDdlpUM3Zxd3I2cUZtT0h
pTmVRMW1Idk5iemczR1VEWUxxMVVpT0ZBUUNTWHp3eTJFd3kyUno0NUdoc0tFV0Z5WTJocGRHVmpkSFZ
5WlY5dVlXMWxFZ1o0T0RZdE5qUWFGZ29NWTI5dGNHRnVlVjl1WVcxbEVnWkhiMjluYkdVYUZ3b0tiVzl
rWld4ZmJtRnRaUklKUTJoeWIyMWxZMkZJOR2hZS0RYQnNZWFJtYjNKdFlnNKdFgyNWhiV1VTQ1V4cGJuRnZ2d
JQVJBQkdBRWdBQkpVQ2xJBBZ0JFaERsQUg1dW5jMWF3SlVnTHROMWc0TE5HZzEzYVdSbGRtbHVaVjk
wWlhOMEloRlVSVk5VWDBOUFFRsUkZUbFJmMU5VSZk1Tb0NVMFF5QUJBQkdoQnVWVCsxUkJ4dUkzc3FNUzJ
WWGZDU0dBRWd6ZVM2bmdVd0ZTcUFBcG5SSVRSiUkV0OU9JZzhQZHJFSU0wRVU4NUFUUbHczZVI5VU42Vzl
VUUw4UWdEM0RBWEx4blVxT0I4V1pCVEJ5a3pKOTDRIUVZtQzhoQcW5wYUxGWHlrSXZZMcFBTb2I0MFdHaHNNRFV
RSUJwNzRkTkdON21YcTlNS2xmeG9DTlJ1cUVSVkRndzFzMThSZlh2QVNtbmcyZ2ViUjVmaDg3dmZ2R2l
TTG9TamhEUUdINGpoSXRWK24ySkJxR1BYZWEyaXFGdVVUrZmlhem82ZTRLMnpsMTJqQ1hBMVJPUzdUZyt
oeEhscURvR2oydDRvRXFCdVNlVzhVRkg0WU1ucUxxlOHB0V0dRMDBVR1gvcUY5K1pTTUnpnU1RLTGd0V0p
lTStTMGNOSUtnSEZUWmkvSmNNaU0N5VmtzY0VqOW5pTVU5a1BqQkJ6YlIzTHJQZXZ2WFNkZmFOUTVWWVVN
iU2U4PSJ9", "signature": "VYLAAktBcg3nAVexs/owBk95mNCs7a9ATVNlu+E2nj8="}'
```

```
https://license.uat.widevine.com/cenc/getlicense/widevine_test
```

Expected Response

```
{"status":"OK","license":"CAISyAMKKgoQblU/tQ28biN7KjEtlV3wkhIQblU/tQ28biN7KjEtlV
3wkhoAIAEoABIKCAEQARgBMKCZAhpmEhAnLYoeF1JERtTGJn3PZ43xGlAuk4wxrzu3jBux9utLoVA2Ag
xlTskglmzmVxZxqsz9CgIq24i3DSRJAblpFscyG4bg1SEAyOCWBCMKsNZ5leVnADRnVH20gfwoj/uc4i
z8FSABGl4KJDBkZjI1Y2M1LTlhZDEtNTI4Yi05MGRjLTU2MmNkMTgxMTgzYhIQd9H5HL/xbXCSnLbjrs
xZ4Rog0b01gbD9VbfpmklAQol7oAC1/wZpXNie/BqJniPehBAgAigBGl4KJDc1ZWE4YmFjLWY4N2MtNW
MzMS1hZmE2LTVmN2JlMDYxNTBlORIQcKCpMlX8UpJ1lazb6QAyehogckuIGdK43wCoChZ8qlpXJ+6fsR
gnkC0rLg/hx4Kn2oYgAigBGl4KJDQ3MzFjYTcyLTlhMWMtNWI0NS04YzEzLTBmODAyYjA0YjFjYxIQMs
zrg257yeHnZH71T11g6Rogys15dHsuh8M+H0LpJORM5JLRzQ6Q6t9i+qJDxVfsfjkgAigBIM3kup4FOA
AaIG2P7dWmCAY2a59Vu/HGJ8IbLbqA0UQ0yUETDC0VuJIKIoAC019cqv20XIGOmbAJPLAhhelyBDTYXk
Wuzk3oZX7G3tbbGUgHBBVoVXyMGiqr2ZTPKsc7EfPnofdMhgfdtomKr/ezsQM9TWAeKy8s9X04cBG42/
N7CtR2186WxJdwA2E+uipptJKBzEpB9huAKbGpxQLA5onli6tBn3ft7qz3fPAWAr9JkpjiCIcqosXM03
A6n1AQP6GvAPXN2S9aiy28JacHQCyE97OLjAL9yiG9MuTIvEBxdG/cQ5wloOTdK2ICGFq3vEOfLfVCqM
vgumKQ4QMXZmuH/NUMIRT89Z8Ibq4wHo1nFMjyrKFpmkj57Xuh3w7CzCEWnD43rp+IMtN1qQ==","lic
ense_metadata":{"content_id":"WU9VVFVCRTpDd3E3bWVXZWM3MA==","license_type":"STRE
AMING","request_type":"NEW"},"supported_tracks":[{"type":"SD","key_id":"MGRmMjVj
YzUtOWFkMS01MjhiLTkwZGMtNTYyY2QxODExODNi"},{"type":"HD","key_id":"NzVlYThiYWMtZj
g3Yy01YzMxLWFmYTYtNWY3YmUwNjE1MGU5"},{"type":"AUDIO","key_id":"NDczMWNhNzItOWExY
y01YjQ1LThjMTMtMGY4MDJiMDRiMWNj"}],"make":"Google","model":"ChromeCDM-Linux-x86"
,"security_level":3,"internal_status":0,"session_state":{"license_id":{"request_
id":"blU/tQ28biN7KjEtlV3wkg==","session_id":"blU/tQ28biN7KjEtlV3wkg==","purchase
_id":"","type":"STREAMING","version":0},"signing_key":"9Y+m1Rf4AmqWRT01EpagStZ/z
REV4Dz8es611yBiZ5wIo0wEQ4FfLNHwTnq2shN218OlOx3reeR/+R05ltH/wg==","keybox_system_
id":4183,"license_counter":0},"cert_serial_number":"NDM2ODcyNmY2ZDY1NDM0NDRkMmQ0
YzY5NmU3NTc4MmQ3ODM4MzY=","device_whitelist_state":"DEVICE_WHITELISTED","message
_type":"LICENSE","platform":"","device_state":"RELEASED","pssh_data":{"key_id":[
"5QB+bp3NWsCVIC7TdYOCzQ=="],"content_id":"VEVTVF9DT05URU5UX0lEXzE="},"client_hdc
p_version":"HDCP_NONE"}
```

Revision History

Version	Date	Description
1.6	12/5/2014	- Expanded available parameters. - Clarification on string value types. - Added Appendix (License duration, Security levels, HDCP output)
1.7	1/29/2015	- Additional edits after review.
1.9	10/12/2015	- Clarification on required / optional fields. - key id has max length of 16 bytes. - Expanded license response details including message_type, request_type, cert_serial_number.
2.0	4/15/2016	- Content_id is now optional in the request. - Added missing content_key_specs.iv field.
2.1	5/11/2016	- Rental duration parameter has been deprecated.
2.2	6/1/2016	- Added section on how to protect external content keys. (draft)
2.3	6/9/2016	- Completed section on how to protect external content keys.
2.4	6/29/2016	- Added always_include_client_id.
2.5	7/7/2016	- Clarification on key_id and key value generation.
2.6	12/19/2016	- Updated renewal document links.