Widevine DRM for HLS

version 1.0

Contents

Revision History

Version	Date	Description
0.1	9/13/2017	Initial draft
0.2	10/12/2017	~~Expanded Attributes section with X- prefix tags.~~ Removed changes
0.3	10/17/2017	Added Widevine PSSH.
0.4	11/1/2017	Restructured V1 and V2 formats, various edits and re-arrangements
0.5	11/28/2017	Updated SAMPLE-AES-CENC tag to SAMPLE-AES-CTR
0.6	12/1/2017	Additional corrections for SAMPLE-AES-CTR
0.7	12/13/2017	Removed overlap of V1/V2 signaling. Various corrections.
0.8	1/4/2018	Correction - content_id must be base64-decoded before copying to protobuf.
0.9	1/11/2018	Made V2 optional for legacy TS
1.0	4/17/2018	Updates to platform support for CMAF

Overview

The purpose of this document is to specify a set of extensions to the base HTTP (HTTP Live Streaming) specification [1] to support the Widevine DRM system.

There are two versions of Widevine DRM signaling for HLS:

Version 1 is the original working design applicable to Chromecast and the Android in-app library. This will continue to be needed for Widevine-protected content delivery to those devices for some time, but will be phased out. It is meant for signaling for HLS using MPEG-2 TS and Sample AES.

Version 2 specifies the new extensions which are meant to be used for HLS in CMAF and be compatible and consistent with Widevine DRM signaling for MPEG-DASH.

References

- [1] HTTP Live Streaming Specification

HLS with CMAF support (V2)

This version of Widevine DRM signaling is required going forward for CMAF content. It may also be present for MPEG-2 TS presentations for forward compatibility.

Expected media format:

- Playlist: m3u8
- Container formats:
    - CMAF-branded MP4
    - ***Optional:*** MPEG-2 TS HLS

Platform	Supported
Android P	Coming Soon Changes are around manifest parsing. Encryption scheme is already supported.
Chrome Browser	Summer 2018 (M68)
EME-based browsers with Widevine CDM	post-Chrome release
Shaka Player	Supported * depends on browser
Chromecast	Q2 2018
NexPlayer	Supported

Goal

- Add a new Widevine EXT-X-KEY tag in HLS playlists, that is similar to elements and attributes in DASH MPD.
- Additions that do not break spec-compliant players.

Non-Goals

- V2 specification has no intent to deprecate the V1 format, which still ought to be used for HLS with MPEG-2 transport streams.

New Format

EXT-X-KEY tag is used in HLS to signal encrypted media in the playlist. HLS mentions that "Two or more EXT-X-KEY tags with different KEYFORMAT attributes MAY apply to the same Media Segment if they ultimately produce the same decryption key". So adding a new KEYFORMAT is spec compliant and existing Widevine HLS players that already handle Android format may ignore the new format.

Attributes

- KEYFORMATVERSION="1"
    - This proposal defines key format version 1.
    - Version 1 defines the following attributes.
- KEYFORMAT="`urn:uuid:edef8ba9-79d6-4ace-a3c8-27dcd51d21ed`"
    - The UUID is the Widevine UUID from DASH IF IOP. The same exact string is used in MPD with Widevine encrypted streams.
    - Note that the in section 6.3.6, the spec mentions that "*A client MUST ignore any EXT-X-KEY tag with an unsupported or unrecognized KEYFORMAT attribute, to allow for cross-device addressability.*" Which means a spec-compliant player should not prevent playback solely due to unrecognizing this key format.
- URI="`data:text/plain;base64,<base64 encoded PSSH box>`"
- METHOD indicates the encryption cipher used when encrypting the content.
    - SAMPLE-AES signals that the content is encrypted using 'cbcs'.
    - SAMPLE-AES-CTR signals that the content is encrypted using one of the AES-CTR protections schemes, namely 'cenc'.

Attributes mapped to DASH MPD

- KEYFORMAT
    - `ContentProtection` element's `schemeIdUri` attribute.
- URI
    - The content of `cenc:pssh` element.
- KEYID
    - 16-byte hexadecimal string (`0x--------------------------------`) encoding the key ID which has the same role as the default_kid in MPEG DASH. That is, the key which is retrieved from a DRM server to decrypt the content, which may or may not be the same as the actual content key ID. If using a hierarchical key scheme, this would be the "root" key.

Example HLS Playlist with V2 Signaling

```
#EXTM3U
#EXT-X-VERSION:6
#EXT-X-TARGETDURATION:2
#EXT-X-PLAYLIST-TYPE:VOD
#EXT-X-MAP:URI=”init_segment.mp4”
#EXTINF:1.001,
output_video-1.mp4
#EXT-X-DISCONTINUITY
#EXT-X-KEY:METHOD=SAMPLE-AES,URI=”data:text/plain;base64,AAAAPXBzc2gA
AAAA7e+LqXnWSs6jyCfc1R0h7QAAAB0aDXdpZGV2aW5lX3Rlc3QiDHRlc3QgY29udGVud
A==”,KEYID=0x112233445566778899001122334455,KEYFORMAT=”urn:uuid:edef8
ba9-79d6-4ace-a3c8-27dcd51d21ed”,KEYFORMATVERSION=”1”
#EXTINF:1.001,
output_video-2.mp4
#EXTINF:0.734,
output_video-3.mp4
#EXT-X-ENDLIST
```

Appendix A: Widevine PSSH

The Widevine PSSH is encoded in a version 0 ProtectionSystemSpecificHeader ('pssh') with system ID `edef8ba9-79d6-4ace-a3c8-27dcd51d21ed`. The Data field contains a serialized protocol buffer defined as follows:

```
// Each SubLicense message represents a single content key. These keys can be
// added to Widevine CENC initialization data to support both content grouping
// and key rotation.
message SubLicense {
  // Required. The key_id of a SUB_SESSION_KEY received in the master license.
  // SUB_SESSION_KEY is defined in the Widevine License Protocol.
  optional bytes sub_session_key_id = 1;

  // Required. The key_msg contains the bytes of a serialized SignedMessage
  // proto. Internally the message field will contain a serialized KeyContainer
  // holding a single content key.
  optional bytes key_msg = 2;
}

// Container for keys which are wrapped using an entitlement key from a master
// license.
message WrappedKey {
  // ID of the wrapped key. Required.
  optional bytes key_id = 1;
  // ID of wrapping key. Required.
  optional bytes wrapping_key_id = 2;
  // IV used to wrap the key. Required.
  optional bytes wrapping_iv = 3;
  // Encrypted entitled key. Wrapped with the entitlement key and IV, using
  // AES-256-CBC with PKCS#7 padding. Required.
  optional bytes wrapped_key = 4;
}

message WidevinePsshData {
  // Key IDentifier(s). This field is mutually exclusive with content_id, below.
Only
  // One or the other, but at least one must be present.
  repeated bytes key_ids = 2;

  optional string provider = 3 [deprecated = true];

  // A content identifier, specified by content provider.
  // This field is mutually exclusive with key_ids, above. Only
```

```
  // One or the other, but at least one must be present.
  optional bytes content_id = 4;

  // Crypto period index, for media using key rotation.
  optional uint32 crypto_period_index = 7;

  // Protection scheme identifying the encryption algorithm. The protection
  // scheme is represented as a uint32 value. The uint32 contains 4 bytes each
  // representing a single ascii character in one of the 4CC protection scheme
  // values. To be soon deprecated in favor of signaling from content.
  // 'cenc' - (AES-CTR) protection_scheme = 0x63656E63,
  // 'cbc1' - (AES-CBC) protection_scheme = 0x63626331,
  // 'cens' - (AES-CTR pattern encryption) protection_scheme = 0x63656E73,
  // 'cbcs' - (AES-CBC pattern encryption) protection_scheme = 0x63626373.
  optional uint32 protection_scheme = 9 [default = 0];

  // Optional. For media using key rotation, this represents the duration
  // of each crypto period in seconds.
  optional uint32 crypto_period_seconds = 10;

  // Optional licenses containing additional keys for purposes of content key
  // rotation for devices with OEMCrypto 13 or older. Ought to be present only
  // in media segments, and never in the media initialization data.
  repeated SubLicense sub_licenses = 11;

  // IDs of the groups to which the content belongs. A group is a set of
  // content IDs. A particular piece of content may belong to multiple groups.
  repeated bytes group_ids = 12;

  // Copy/copies of the content key used to decrypt the media stream in which
  // the PSSH box is embedded, each wrapped with a different entitlement key.
  // May be repeated if using group entitlement keys. Optional, used for content
  // key rotation for devices with OEMCrypto 14 or newer. Ought to be present
  // only in media segments, and never in the media initialization data.
  repeated WrappedKey entitled_keys = 13;

  // Key sequence for Widevine-managed keys. Optional.
  optional uint32 key_sequence = 14;

}
```

Appendix B: Legacy Widevine HLS (V1)

This type of Widevine DRM signaling is meant for legacy devices, consuming media streams as MPEG-2 TS.

Expected media format for V1:

- Playlist: m3u8
- Container format: MPEG-TS

Platform	Supported
Android N (7.x)	Yes
Android O (8.x)	Yes
Chromecast	Yes
NexPlayer SDK	Yes

Media playlist information

HLS playlists contains EXT-X-KEY tags that specify how encrypted media segments may be decrypted. An attribute list is associated with each tag. The attribute list applies to the media segments seen until the next EXT-X-KEY tag with the same KEYFORMAT attribute. A summary of the attributes, relevant to this integration, is listed below. A more complete description can be found in the HLS specification. One or more EXT-X-KEY tags may be present in a media playlist, one for each DRM vendor.

- METHOD (**required**): SAMPLE-AES
- URI (**required**): A quoted string containing a "data" URL (RFC 2397) whose data item is a json formatted version of the CENC init data. The "data" URL shall specify a media-type text/plain and use the base64 option.
- IV (optional): hexadecimal sequence that specifies a 128-bit unsigned integer Initialization vector. If absent the media sequence number is used as IV when decrypting a media segment.
- KEYFORMAT (**required**): quoted string that specifies how the key is represented. As multiple EXT-X-KEY tags may be present in a media playlist, this attribute is used by DRM vendors to disambiguate attribute lists meant for them. Widevine use the CENC Key System “com.widevine” to identify its attribute list.
- KEYFORMATVERSIONS (**optional**): quoted strings containing one or more positive integers separated by the “/” character. If not present, its value is considered to be “1”.

The optionality of the parameters above refers to whether they are needed in an attribute list for the Widevine CDM to work correctly. This is separate from requirements that the HLS specification imposes. Here is an example of what an EXT-X-KEY attribute list might look like for Widevine

```
EXT-X-KEY: METHOD=SAMPLE-AES, \
URI="data:text/plain;base64,eyAKICAgInByb3ZpZGVyIjoibWxiYW1oYm8iLAogICAiY29udG
VudF9pZCI6Ik1qQXhOVjlVWldGeWN3PT0iLAogICAia2V5X2lkcyI6CiAgIFsKICAgICAgIjM3MWUx
MzVlMWE5ODVkNzVkMTk4YTdmNDEwMjBkYzIzIgogICBdCn0=",
 \
IV=0x6df49213a781e338628d0e9c812d328e, \
KEYFORMAT="com.widevine", \
KEYFORMATVERSIONS="1"
```

The init data specified in the URI will be a json formatted version of the WidevineCencHeader protobuf. Here is an example,

```
{
   "content_id":"MjAxNV9UZWFycw==",
   "key_ids":
   [
      "371e135e1a985d75d198a7f41020dc23"
   ]
}
```

The init data json string will consist of

- provider: json string. Encoded with ASCII characters only.
- content_id: json string. Base64 encoded Content ID (as specified by RFC 4648). This value shall be consistent with the Content ID specified to the license server.
- key_ids: json array, whose individual values are a hexadecimal sequence, a string of 32 characters drawn from the set [0-9a-fA-F].

The data from the init data json string will be used to populate the fields of a serialized WidevineCencHeader proto. This protobuf will be included in the license request.

  - The provider value shall be copied as is into the protobuf field.
  - The content_id value shall be base64-decoded first then copied into the protobuf field.
  - The Key ID should be converted, from a 32 character hex ASCII string to a 16 byte value. For KEYFORMATVERSION "1", key_ids shall specify a single 16 byte key ID. If multiple key IDs are present, the first will be used to decrypt the media segments that follow, until the next media playlist is encountered.

Example HLS Playlist with V1 Signaling

```
#EXTM3U
#EXT-X-TARGETDURATION:9
#EXT-X-VERSION:5
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-PLAYLIST-TYPE:VOD
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-KEY:METHOD=SAMPLE-AES,KEYFORMAT="com.widevine",KEYFORMATVERSIO
NS="1",URI="data:text/plain;base64,ewogICAicHJvdmlkZXIiOiAiY2FzdCIsCi
AgICJjb250ZW50X2lkIjogIlltbG5ZblZqYTJKMWJtNTUiLAogICAia2V5X2lkcyI6IFs
KICAgICAgIjliNzU5MDQwMzIxYTQwOGE1Yzc3NjhiNDUxMTI4N2E2IgogICBdCn0=",IV
=0x75537a79fa41abc7b598ea72aba0c26f
#EXTINF:9.00898,
0.ts
#EXTINF:9.00901,
1.ts
```