Guidelines for Implementation: DASH-IF Interoperability Points

Living Document, 17 December 2018

This version:

https://dashif.org/guidelines/

Issue Tracking:

GitHub

Editors:

DASH Industry Forum

Table of Contents

1. Document editing notes§

Documentation: https://dashif.org/DocumentAuthoring/

Example document repository: https://dashif.org/DocumentAuthoring/

Live discussion in #document-authoring on Slack.

2. Chapter 1§

Placeholder text. This document will eventually contain IOP v5.

3. Content Protection and Security§

3.1. Introduction§

DASH-IF IOPs do not intend to specify a full end-to-end DRM system. However DASH-IF IOP provides a framework for multiple DRM systems to protect DASH content by adding proprietary information in predetermined locations in MPDs and DASH content that is encrypted with Common Encryption as defined in [MPEGCENC].

Common Encryption specifies several protection schemes and associated parameters. These can be applied by a scrambling system and used by key mapping methods part of different DRM systems, thanks to a common key identifier (`KID`). The same encrypted version of DASH content can be combined with different DRM systems that can store proprietary information for licenses and key retrieval in the Protection System Specific Header Box (`pssh`) and in ContentProtection elements. The DRM system is identified by a specific DRM SystemID.

The recommendations in this document reduce the encryption parameters and use of the encryption metadata to specific use cases for VOD and live content with key rotation.

3.2. HTTPS and DASH§

Transport security in HTTP-based delivery may be achieved by using HTTP over TLS (HTTPS) as specified in [RFC 8446]. HTTPS is a protocol for secure communication which is widely used on the Internet and also increasingly used for content streaming, mainly for the following purposes:

- Protecting the privacy of the exchanged data from eavesdropping by providing encryption of bidirectional communications between a client and a server, and
- Ensuring integrity of the exchanged data against man-in-the-middle attacks against tampering with and/or forging the contents of the communication.

As a MPD carries links to media resources, web browsers follow the W3C recommendation [mixed-content]. To ensure that HTTPS benefits are maintained once the MPD is delivered, it is recommended that if the MPD is delivered with HTTPS, then the media also be delivered with HTTPS.

In addition, DASH explicitly permits the use of https as a scheme and hence, HTTP over TLS as a transport protocol. When using HTTPS in DASH, one can for instance specify that all media segments are delivered over HTTPS, by declaring that all the `BaseURL`'s are HTTPS based, as follow:

```
<BaseURL>https://cdn1.example.com/</BaseURL>
<BaseURL>https://cdn2.example.com/</BaseURL>
```

One can also use HTTPS for retrieving other types of data carried with an `MPD` that are HTTP-URL based, such as, for example, DRM licenses specified within the `ContentProtection` descriptor:

```
<ContentProtection
  schemeIdUri="urn:uuid:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
  value="DRMNAME version"
  <drm:License>https://MoviesSP.example.com/protect?license=kljklsdfiowek</drm:License>
</ContentProtection>
```

It is recommended to adopt HTTPS for delivering DASH content. It must be noted nevertheless, that HTTPS does hurt proxies that attempt to intercept, cache and/or modify content between the client and the CDN that holds the delivery certs. Since the HTTPS traffic is opaque to these intermediate nodes, they can lose much of their intended functionality when faced with HTTPS traffic.

While using HTTPS in DASH provides good levels of trust and authenticity for data exchanged between DASH servers and clients connected over HTTPS, it should be pointed out that HTTPS only protects the transport link, but not the access to streaming content and the usage of streamed content. HTTPS itself does not imply user authentication and content authorization (or access control). This is especially the case that HTTPS provides no protection to any streamed content cached in a local buffer at a client for playback. HTTPS does not replace a DRM.

3.3. Content Encryption§

> Note: Add cenc and cbcs support

3.4. ISO BMFF Support for Common Encryption and DRM§

3.4.1. ISO BMFF Structure Overview§

The following table provides pointers to relevant information in the specifications to understand the standard DRM components and if the main description is located in the ISO base media file format [MPEG4] or the Common Encryption specification [MPEGCENC].

Box	*Full Name / Usage*	*Reference*
`moof`	movie fragment header - One `moof` box for each fragment, i.e. Media Segment/Subsegment.	[MPEG4] 8.32 + [MPEGDASH]
`moov`	movie header, container for metadata - One `moov` box per file.	[MPEG4] 8.1
`pssh`	Protection System Specific Header Box - Contains DRM specific data. `pssh` box version 1 (specified in Common Encryption 2nd edition) contains a list of `KID`s to allow removing duplicate `pssh` boxes when defragmenting a file by comparing their KIDs	[MPEGCENC] 8.1.1
`saio`	Sample Auxiliary Information Offsets Box - Contains the offset to the IVs & subsample encryption byte ranges.	[MPEG4] 8.7.9

`saiz`	Sample Auxiliary Information Sizes Box - Contains the size of the IVs & subsample encryption byte ranges.	[MPEG4] 8.7.8
`senc`	Sample Encryption Box - Contains Initialization Vectors; and subsample ranges for a Media Segment.	[MPEGCENC] 7.1
`schi`	Scheme Information Box - Container boxes used by that protection scheme type.	[MPEG4] 8.12.6 + [MPEGCENC] 4
`schm`	Scheme Type Box - Contains the encryption scheme, identified by a 4 character code, e.g. `cenc`	[MPEG4] 8.12.5 + [MPEGCENC] 4
`seig`	Cenc Sample Encryption Information Video Group Entry - A sample description containing KIDs describing sample groups in this segment, for key rotation.	[MPEGCENC] 6
`sbgp`	Sample to Group Box - lists a group of samples	[MPEG4] + [MPEGCENC] 5
`sgpd`	Sample Group Description Box - Describes properties of a sample group	[MPEG4] 8.9.3 + [MPEGCENC] 5
`sinf`	Protection Scheme Information Box - Signals that the stream is encrypted	[MPEG4] 8.12.1 + [MPEGCENC] 4
`stsd`	Sample description table (codec type, initialization parameters, stream layout, etc.)	[MPEG4] 8.16
`tenc`	Track Encryption Box - Contains default encryption parameters for the entire track, e.g. `default_KID`.	[MPEGCENC] 8.2.1

The ISO Media Format carries content protection information in different locations. Their hierarchy is explained in the informational chapter below, followed by a reference on where these elements are standardized.

The following shows the box hierarchy and composition for relevant boxes, when using common encryption:

- `moov/pssh` (zero or one per system ID)
- `moov/trak/mdia/minf/stbl/stsd/sinf/schm` (one, if encrypted)
- `moov/trak/mdia/minf/stbl/stsd/sinf/schi/tenc` (one, if encrypted)
- `moof/traf/saiz` (one, if encrypted)
- `moof/traf/saio` (one, if encrypted)
- `moof/traf/senc` (one, if encrypted)

for key rotation

- `moof/traf/sbgp` (one per sample group)
- `moof/traf/sgpd 'seig'` (sample group entry) (one per sample group)
- `moof/pssh` (zero or one per system ID)

The main DRM components are:

1. `tenc` parameters that specify encryption parameters and `default_KID` ([MPEGCENC] section 8.2). The `tenc` information is in the Initialization Segment. Any `KID`s in Movie Fragment sample group description boxes override the `tenc` parameter of the `default_KID`, as well as the `not encrypted` parameter. Keys referenced by `KID` in sample group descriptions must be available when samples are available for decryption, and may be

stored in a protection system specific header box (`pssh`) in each movie fragment box (`moof`). The `@default_KID` information may also appear in the MPD ([MPEGCENC] section 11).

2. `senc` parameters that may store initialization vectors and subsample encryption ranges. The `senc` box is stored in each track fragment box (`traf`) of an encrypted track ([MPEGCENC] section 7.1), and the stored parameters accessed using the sample auxiliary information offset box (`saio`) and the sample auxiliary information size box (`saiz`) ([MPEGDASH] sections 8.7.8 and 8.7.9).
3. `pssh` license acquisition data or keys for each DRM in a format that is Protection System Specific. `pssh` refers to the Protection System Specific Header box described in [MPEGCENC] section 8.1. `pssh` boxes may be stored in Initialization or Media Segments. It may also be present in a `cenc:pssh` element in the MPD ([MPEGDASH] section 5.8.4.1 and [MPEGCENC] section 11.2.1). `cenc:pssh` information in the MPD allows faster parsing, earlier access, identification of duplicate license requests, and addition of DRMs without content modification. `pssh` boxes in Initialization Segments are not recommended because they trigger a license request each time an Initialization Segment is processed in a Web browser for each Representation and bitrate switch.

   > Note: The duplication of the `pssh` information in the Initialization Segment may cause difficulties in playback with HTML5 - EME based players. I.e. content will fail unless players build complex DRM specific license handling.

4. Key rotation is mainly used to allow changes in entitlement for continuous live content. It is used as defined in [MPEGCENC] with the following requirements:

   - Sample To Group Box (`sbgp`) and Sample Group Description Box (`sgpd`) of type `seig` are used to indicate the `KID` applied to each sample, and changes to `KID`s over time (i.e. key rotation). ([MPEGDASH] section 8.9.4) `KID`s referenced by sample groups must have the keys corresponding to those `KID`s available when the samples in a Segment are available for decryption. Keys referenced by sample groups in a Segment may be stored in that Segment in Protection System Specific Header Boxes (`pssh`) stored in the Movie Fragment Box (`moof`). A version 1 `pssh` box may be used to list the `KID` values stored to enable removal of duplicate boxes if a file is defragmented.
   - Keys stored in Media Segment `pssh` boxes must be stored in the same DRM format for all users so that the same Media Segments can be shared by all users. User-specific information must be delivered out of band, as in a root license associated with the `default_KID`, which can be individualized for each DRM client, and control access to the shared `pssh` information stored in Media Segments, e.g. by encrypting the keys stored in Segment `pssh` boxes with a root key provided by the user-specific DRM root license. Common Encryption specifies `pssh` to enable key storage in movie fragments/Segments; but it does not preclude other methods of key delivery that satisfy KID indexing and availability requirements.
   - For details see Section §3.8.1 Periodic Re-Authorization.

3.4.2. ISO BMFF Content Protection Constraints§

- There SHALL be identical values of `default_KID` in the Track Encryption Box (`tenc`) of all Representation referenced by one Adaptation Set. Different Adaptation Sets may have equal or different values of `default_KID`.
- If a W3C Common `pssh` box [encrypted-media] is used with encrypted content, its list of `KID`s SHALL contain only the `default_KID` from the `tenc` box.
- `pssh` boxes SHOULD NOT be present in Initialization Segments, and `cenc:pssh` elements in ContentProtection Descriptors used instead. If `pssh` boxes are present in Initialization Segments, each Initialization Segment within one Adaptation Set SHALL contain an equivalent `pssh` box for each SystemID, i.e. license acquisition from any Representation is sufficient to allow switching between Representations within the Adaptation Set without acquiring a new license.

> Note: `pssh` boxes in Initialization Segments may result in playback failure during browser playback when a license request is initiated each time an Initialization Segment is processed, such as the start of each protected Representation, each track selection, and each bitrate switch. This content requires DASH clients that can parse the `pssh` box contents to determine the duplicate license requests and block them.

A `cenc:pssh` element is parsed at most once per Adaptation Set by a client's MPD parser, and the potential need for a new license request is identified by a new `@cenc:default_KID` value. In this case, only the DASH client initiates license requests, and may do so per Period, if `@cenc:default_KID` is a new value and the DRM system does not already have the key available for use.

3.5. DASH MPD Support for Common Encryption and DRM§

3.5.1. MPD Structure Overview§

The main DRM components is:

1. The `ContentProtection` descriptors in the `MPD` ([MPEGDASH] sections 5.3.7.2-Table 9, 5.8.4.1, and 5.8.5.2) that contains the URI for signaling of the use of Common Encryption or the specific DRM being used.

The `MPD` contains signaling of the content encryption and key management methods used to help the receiving client determine if it can possibly play back the content. The `MPD` elements to be used are the `ContentProtection` Descriptor elements. At least one `ContentProtection` Descriptor element SHALL be present in each `AdaptationSet` element describing encrypted content.

3.5.1.1. ContentProtection Descriptor for mp4protection Scheme§

A `ContentProtection` descriptor with the `@schemeIdUri` value equals to "urn:mpeg:dash:mp4protection:2011" signals that content is encrypted with the scheme indicated in the `@value` attribute. The file structure of content protection schemes is specified in [MPEG4] section 5.8.5.2, and the `@value` = `cenc` or `cbcs` for the Common Encryption scheme, as specified in [MPEGCENC]. Although the `ContentProtection` Descriptor for UUID Scheme described below is usually used for license acquisition, the `ContentProtection` Descriptor with `@schemeIdUri`="urn:mpeg:dash:mp4protection:2011" and with `@cenc:default_KID` may be sufficient to acquire a license or identify a previously acquired license that can be used to decrypt the Adaptation Set. It may also be sufficient to identify encrypted content in the MPD when combined with license acquisition information stored in `pssh` boxes in Initialization Segments.

A `ContentProtection` Descriptor for the mp4 Protection Scheme shall be used to identify the default KID, as specified by the `tenc` box, using the `@cenc:default_KID` attribute defined in [MPEGCENC] section 11.1. The value of the attribute is the `KID` expressed in UUID string notation.

```
<ContentProtection
  schemeIdUri="urn:mpeg:dash:mp4protection:2011"
  value="cenc"
  cenc:default_KID="34e5db32-8625-47cd-ba06-68fca0655a72"/>
```

When starting playback of any Adaptation Set, the client should interact with the DRM system to verify that the media key identified by the Adaptation Set's default KID is available and should not assume that a media key is available for decrypting content unless so signaled by the DRM system.

When the `default_KID` is present on each Adaptation Set, it allows a player to determine if a new license needs to be acquired for each Adaptation Set by comparing their `default_KIDs` with each other, and with the `default_KIDs` of stored licenses. A player can simply compare these KID strings and determine what unique licenses are necessary without interpreting license information specific to each DRM system.

3.5.1.2. ContentProtection Descriptor for UUID Scheme

A UUID `ContentProtection` descriptor in the MPD may indicate the availability of a particular DRM scheme for license acquisition. An example is provided below:

```
<ContentProtection
  schemeIdUri="urn:uuid:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
  value="DRMNAME version"/>
```

The `@schemeIdUri` uses a UUID URN with the UUID string equal to the registered SystemID for a particular DRM system. A list of known DRM SystemIDs can be found in the DASH-IF identifier repository.

This is specified in [MPEG4] section 5.8.5.2 and is referred to as “ContentProtection Descriptor for UUID Scheme” in the following.

3.5.1.3. Protection System Specific Header Box cenc:pssh element in MPD

A `pssh` box is defined by each DRM system for use with their registered SystemID, and the same box can be stored in the `MPD` within a `ContentProtection` Descriptor for UUID scheme using an extension element in the `cenc:` namespace. Examples are provided in [MPEGDASH-IMPGUIDE] and in [MPEGCENC] section 11.2.

Carrying `@cenc:default_KID` attribute and a `cenc:pssh` element in the `MPD` is useful to allow key identification, license evaluation, and license retrieval before live availability of initialization segments. This allows clients to spread license requests and avoid simultaneous requests from all viewers at the instant that an Initialization Segments containing license acquisition information in `pssh` becomes available. With `@cenc:default_KID`indicated in the mp4protection `ContentProtection` Descriptor on each Adaptation Set, clients can determine if that key and this presentation is not available to the viewer (e.g. without purchase or subscription), if the key is already downloaded, or which licenses the client SHOULD download before the `@availabilityStartTime` of the presentation based on the `default_KID` of each `AdaptationSet` element selected.

3.5.2. MPD Content Protections Constraints

For an encrypted Adaptation Set, `ContentProtection` Descriptors shall always be present in the `AdaptationSet` element and apply to all contained `Representations`.

A `ContentProtection` Descriptor for the mp4protection Scheme (`@schemeIdUri` equals to "urn:mpeg:dash:mp4protection:2011" and `@value=cenc` or `cbcs`) SHALL be present in the `AdaptationSet` element if content represented by the contained `Representations` is encrypted. This allows clients to recognize that content is encrypted with a common encryption scheme without the need to understand any DRM system specific element. This element SHALL contain the attribute `@cenc:default_KID`. The `tenc` box that specifies the encoded track encryption parameters shall be considered the source of truth for the default key ID value since it contains the `default_KID` field. The `@cenc:default_KID` attribute SHALL match the `tenc` default_KID value. This allows general purpose clients to identify the default KID from the `MPD` using a standard location and format without the need to understand any DRM specific element.

For each `ContentProtection` element with UUID Scheme, the `@cenc:pssh` element SHOULD be present. The base64 encoded contents of the element SHALL be equivalent to a `pssh` box including its header. The information SHOULD be sufficient to allow for license acquisition. The `@value` attribute SHOULD contain the DRM system name and version in a human readable form.

Note: A player such as DASH.js hosted by a browser may pass the contents of this `pssh` element through the Encrypted Media Extension (EME) API to the DRM system Content Decryption Module (CDM) with a SystemID equal to the element's UUID value. This allows clients to acquire a license using only information in the `MPD`, prior to downloading Segments.

Below is an example of the recommended format for a hypothetical acme DRM service:

```
<ContentProtection
  schemeIdUri="urn:uuid:d0ee2730-09b5-459f-8452-200e52b37567"
  value="Acme DRM 2.0">
  <cenc:pssh>
    YmFzZTY0IGVuY29kZWQgY29udGVudHMgb2YgkXBzc2iSIGJveCB3aXRoIHRoaXMgU3lzdGVtSUQ=
  </cenc:pssh>
</ContentProtection>
```

3.6. Mix ISOBMFF and MPD Content Protections Constraints§

In the case where the `pssh` box element is present in the `MPD` and in the Initialization Segment, the `pssh` box element in the `MPD` SHALL take precedence, because the parameters in the `MPD` will be processed first, are easier to update, and can be assumed to be up to date at the time the `MPD` is fetched.

Recommended scheduling of License and key delivery:

- Request licenses on initial processing of an `MPD` if `ContentProtection` Descriptors or Initialization Segments are available with license acquisition information. This is intended to avoid a large number of synchronized license requests at `MPD@availabilityStartTime`.
- Prefetch licenses for a new Period in advance of its presentation time to allow license download and processing time, and prevent interruption of continuous decryption and playback. Advanced requests will also help prevent a large number of synchronized license requests during a live presentation at `Period@startTime`.

The DRM system is signaled in the `MPD` and `pssh` boxes with a SystemID. A list of known DRMs can be found in the DASH identifier repository.

3.7. Client Interactions with DRM Systems§

The client interacts with one or more DRM systems during playback in order to control the decryption of content. Some of the most important interactions are:

1. Determining the availability of media keys.
2. Requesting the DRM system to acquire media keys.

In both of these interactions, the client and DRM system use the `default_KID` as an abstract mechanism to communicate information regarding the capability to decrypt adaptation sets that use a particular `default_KID`. A DRM system may also make use of other media keys in addition to the one signalled by `default_KID` (e.g. in key derivation or sample variant schemes) but this SHALL be transparent to the client, with only the `default_KID` being used in communications between the client and DRM system.

A client SHALL determine the required set of media keys based on the default KIDs signalled in the manifest for the adaptation sets selected for playback.

Upon determining that one or more required media keys signalled by default KIDs are not available the client SHOULD interact with the DRM system and request the missing media keys. The client MAY also request media keys that are known to be usable. Clients SHALL explicitly request all required media keys signaled by default KIDs and SHALL NOT assume that requesting one key from this set will implicitly make others available.

The client and/or DRM system MAY batch multiple key requests (and the respective responses) into a single transaction (for example, to reduce the chattiness of license acquisition traffic).

3.8. Additional Constraints for Specific Use Cases§

3.8.1. Periodic Re-Authorization§

This section explains different options and tradeoffs to enable change in keys (aka key rotation), considering different use cases, application scenarios, content encoding variants and signaling requirements.

3.8.1.1. Use Cases and Requirements§

The main use case in this context is to enable service changes at program boundaries, not to increase security of Common Encryption by preventing e.g. key factoring or key redistribution. In order to clarify this application, the term periodic re-authorization is used instead of the term key rotation.

In addition, this is one of the ways to implement counting of active streams as they are periodically requesting keys from a license server. The following use cases and requirements have been considered:

- Ability to force a client device to re-authorize to verify that it is still authorized for content consumption.
- Support for distribution models such as: Live content, PVR, PPV, VOD, SVOD, live to VOD, network DVR. This includes where live content is converted into another consumption license for e.g. catch up TV.
- Uninterrupted playback when keys are rotated.
    - Preventing of client storm: Requests from client should be distributed where possible to prevent spiking loads at isolated times.
    - Quick recovery: If the server or many client devices fail, the service should be able to resume quickly.
    - Player visibility into the key rotation signal
- Regional blackout: Device location may be taken into account to enable de-activation of content in a geographical area.
- Hybrid broadcast/unicast networks in which receivers operating in broadcast-only mode at least some of the time, i.e. unable to always download licenses on-demand through unicast.
- No required changes to the standard process and validity of `MPDs`.

3.8.1.2. Implementation Options§

This section describes approaches for periodic re-authorization; recommended because they best cover the use cases and allow interoperable implementation. Other approaches are possible and may be considered by individual implementers.

One of those is explicit signaling using e.g. esmg messages, using a custom key rotation signal to indicate future `KIDs`. To prevent the initial client storm to retrieve the first keys, before they are rotated, the initial `pssh` parameters SHOULD be included in the `MPD` as described in §3.5.2 MPD Content Protections Constraints.

3.8.1.2.1. Period Boundaries§

One possibility is to use a `Period` as minimum key duration interval and existing `MPD` level signaling for `KID`.

This is a simple implementation and a possible alternative but has limitations in the flexibility:

- The signal does not allow for early warning and time to switch the encryption keys and context.
- The logic of the periods is decided by content creation not DRM. Boundaries may not be suited and period may be longer than desired key interval.

3.8.1.2.2. Future Keys in pssh§

This approach considers the protection system to be responsible to manage notification and key retrieval that prevents a client storm. The `pssh` information is used for signaling in a content protected system proprietary form. No additional signaling mechanism is created and the DRM is managing key rotation by providing extra information in the Protection System Specific Header Box (`pssh`) ([MPEGDASH]). To prevent a client storm on key change boundaries the following implementation options can be considered. They are listed for informational purpose and do not affect the guidelines on content formatting.

Current and future keys or access information and validity times are provided in a proprietary format in the `pssh` (see example in figure below). The client can chose a random time to use the access information to request licenses so that requests are distributed over time.

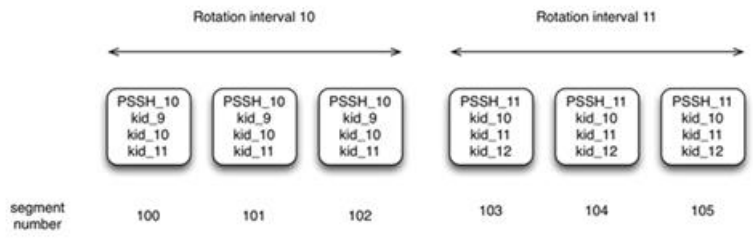

Figure 1 *`pssh` with version numbers and `KIDs`.*

3.8.1.2.3. Key Hierarchy§

The above approach also makes the protection system responsible to manage the key update and limits head end communication by using different types of licenses that established a hierarchy as follows:

- Entitlement Management License (EML) – A license a broadcaster can issue once to enforce some scope of content, such as a channel or library of shows (existing and future). It is cryptographically bound to one DRM domain associated with one user ID and, and enables access to ECLs and media keys associated with each show it authorizes.
- Entitlement Control License (ECL) – A license that contains a media key and can only be accessed by provisioned devices that have been authorized by installing the associated EML. ECLs may be delivered with the media in a broadcast distribution.

Changing media keys and ECLs per asset, forces re-authorization of each show by the DRM system which needs the media key.

When using any type of key hierarchy, the `@cenc:default_KID` value in the `ContentProtection` element, which is also encoded into the `tenc`, is the ID of the key which gives access to the content key(s). This is usually the key requested by the DRM client, and delivered in the EML.

3.8.1.3. Additional Constraints for Periodic Re-Authorization§

- Key rotation should not occur within individual segments, as their duration is typically short enough to enable the intended use cases.
- Each Movie Fragment SHOULD contain one `pssh` in each `moof` box per SystemID that contains sufficient

information for the DRM system with matching SystemID to obtain protected keys for this movie fragment, when combined with:

  - Information from `pssh` in `moov` or `cenc:pssh` in `MPD`.
  - `KID` associated with each sample from `seig` sample group description box.
  - Sample to group boxes that list all the samples that use a particular `KID`.
- The KID should be observable by the player by reading the `clear key_ids` in `pssh` definition v1.
- If the key is does not need to be retrieved, a `pssh` update may not result in a license request.
- If `KID` cannot be observed, the player may perform binary comparison of `pssh` segments to understand updates.

3.8.2. Low Latency§

> Note: To be added.

3.8.3. Encryption of Different Representations§

Representations contained in one Adaptation Set SHALL be protected by the same license for each protection system (“DRM”), and SHALL have the same value of `default_KID` in their `tenc` boxes in their Initialization Segments. This is to enable seamless switching within Adaptation Sets, which is generally not possible if a new DRM license needs to be authorized, client bound, generated, downloaded, and processed for each new Representation.

In the case of key rotation, if root licenses are used, the same requirement applies to the root licenses (one license per Adaptation Set for each DRM), and also means all Representations SHALL have the same value of `default_KID` in their `tenc` boxes in their Initialization Segments. The use of root and leaf licenses is optional and DRM specific, but leaf licenses are typically delivered in band to allow real time license acquisition, and do not require repeating client authentication, authorization, and rebuilding the security context with each key change in order to enable continuous playback without interruption cause be key acquisition or license processing.

In cases where SD and HD and UHD Representations are contained in one presentation, different license rights may be required for each quality level and may be sold separately. If different licenses are required for different quality levels, then it is necessary to create separate Adaptation Sets for each quality level, each with a different license and value of `default_KID`.

Representations that are equivalent resolution and bitrate but encrypted with different keys may be included in different Adaptation Sets. Seamless switching between UHD, HD and SD Representations is difficult because these quality levels typically use different decryption licenses and keys, use different DRM output rules (prohibit analog interfaces, require resolution down-scaling, require HDCP encryption on output, etc.), and use different decoding parameters for e.g. subsampling, codec, profile, bit depth, aspect ratios and color spaces.

If any Representation is encrypted in an Adaptation Set, then all must be encrypted using the same `default_KID` in the Track Encryption Box (`tenc`) to avoid realtime changes to the DRM licenses and security context. `KID` values may change over time (“key rotation”) as specified in Common Encryption and a particular DRM system.

For all Representations within an Adaptation Set with `@bitstreamSwitching`=”false” (default), the following parameters shall apply.

- `default_KID` in `tenc` shall be equal for all Representations

3.8.4. Encryption of Multiple Periods§

If a new license is needed and `@cenc:default_KID` is to be changed, it SHALL be at the beginning of a Period. A different file is indicated by a different `default_KID` signaled in the `tenc` box in the Initialization Segment.

A file associated with a single license may be continued over multiple Periods by being referenced by multiple Representations over multiple Periods (for instance, a program interspersed with ad Periods). A client can recognize the same `@cenc:default_KID` value and avoid having to download the same license again; but the DRM system may require a complete erase and rebuild of the security context, including all key material, samples in process, etc., between `Periods` with different licenses or no license (between protected and clear Periods).

3.8.5. Protection of Media Presentations that Include SD, HD and UHD Adaptation Sets§

Per DASH IF interop points, Representations with separate keys, licenses, and license policy are contained in different Adaptation Sets.

Adaptive bitrate switching can function automatically within an Adaptation Set without changing keys, licenses, robustness and output rules, etc.

A player may download licenses for multiple Adaptation Sets in a Group, and seamlessly switch between them if it is able. Seamless switching between Adaptation Sets is allowed, but not required. DASH may need to signal which Adaptation Sets are intended for seamless switching, i.e. have identical source content, same picture aspect ratio, same exact rescaled pixel registration, same sample description (e.g. `avc3`), same initialization behavior (`@bitstreamSwitching` = true/false), same Timescale and `@timescale`, and are mutually time-aligned.

The DASH-IF interop points are intended to make bitrate switching within an Adaptation Set simple and automatic, whether Representations are encrypted or not. Placement of Representations in different Adaptation Sets informs players that those Representations need to be initialized with different parameters, such as a different key and license. The full initialization process is repeated per Period. Adaptation Sets with `@bitstreamSwitching` = “true” only need to be initialized once per Period. Adaptation Sets with `@bitstreamSwitching` = “false” need to be partially re-initialized on each Representation switch (to change the SPS parameter sets referenced from NALs to those stored in in the containing track’s `avcC`), but most initialized parameters such as timescale, codec Profile/Level, display buffer size, colorspace, etc.; and licenses and the DRM system do not need to be changed.

Fetching and resetting keys and licenses during adaptive switching requires processing Initialization Segments with different `tenc` `default_KID`, protection scheme and possibly `pssh` boxes. That may not be seamless, especially in browser playback where the decoders are only aware of player switching when an Initialization Segment flows through the MSE buffer and a needKey() event is raised via EME.

Note that switching between Adaptation Sets with different Media Profiles could be restricted by key and license policy, e.g. the user only purchased SD rights, the player only has analog output and HD content requires a protected digital output, UHD content requires hardware protected DRM, etc.

Implementations that seamlessly switch between Representations with different keys and policies generally require a standardized presentation ID or content ID system that associates multiple keys and licenses to that ID and presentation, then downloads only the keys/licenses authorized for that user and device (e.g. SD or HD+SD). The player must then install those licenses and use player logic to select only Representations in an Adaptation Set for which a license is installed and output controls, display configuration, etc. allow playback (e.g. only Representations keyed for an installed SD license). Players and license servers without this pre-configuration protocol and adaptive switching logic will encounter key/license requests in the process of adaptive switching, and may find output blocked by different license policies, user rights, etc.

3.8.6. Use of W3C Clear Key with DASH§

When using Clear Key with DASH [encrypted-media], Clear Key management availability is signaled in the MPD with a `ContentProtection` element that has the following format.

The Clear Key `ContentProtection` element attributes take the following values:

- The UUID e2719d58-a985-b3c9-781a-b030af78d30e is used for the `@schemeIdUri` attribute.
- The `@value` attribute is equal to the string “ClearKey1.0”

The following element MAY be added under the `ContentProtection` element:

- `Laurl` element that contains the URL for a Clear Key license server allowing to receive a Clear Key license in the format defined in [encrypted-media] section 9.1.4. It has the attribute `@Lic_type` that is a string describing the license type served by this license server. Possible value is “EME-1.0” when the license served by the Clear Key license server is in the format defined in [encrypted-media] section 9.1.4.

The name space for the `Laurl` element is `http://dashif.org/guidelines/clearKey`

An example of a Clear Key `ContentProtection` element is as follows

```
<xs:schema xmlns:ck=http://dashif.org/guidelines/clearKey>
<ContentProtection
  schemeIdUri="urn:uuid:1077efec-c0b2-4d02-ace3-3c1e52e2fb4b"
  value="ClearKey1.0">
  <ck:Laurl Lic_type="EME-1.0">
    https://clearKeyServer.foocompany.com
  </ck:Laurl>
</ContentProtection>
```

W3C also specifies the use of the SystemID=”1077efec-c0b2-4d02-ace3-3c1e52e2fb4b” in [eme-initdata-cenc] section 4 to indicate that tracks are encrypted with Common Encryption [MPEGCENC], and list the `KID` of keys used to encrypt the track in a version 1 `pssh` box with that SystemID. However, the presence of this Common `pssh` box does not indicate whether keys are managed by DRM systems or Clear Key management specified in this section. Browsers are expected to provide decryption in the case where Clear Key management is used, and a DRM system where a DRM key management system is used.

Therefore, clients SHALL NOT use the signalling of SystemID 1077efec-c0b2-4d02-ace3-3c1e52e2fb4b as an indication that the Clear Key mechanism is to be used.

W3C specifies that in order to activate the Clear Key mechanism, the client must provide Clear Key initialization data to the browser. The Clear Key initialization data consists of a listing of the default KIDs required to decrypt the content.

The MPD SHALL NOT contain Clear Key initialization data. Instead, clients SHALL construct Clear Key initialization data at runtime, based on the default KIDs signaled in the MPD using `ContentProtection` elements with the urn:mpeg:dash:mp4protection:2011 scheme.

When requesting a Clear Key license to the license server, it is recommended to use a secure connection as described in Section §3.2 HTTPS and DASH.

When used with a license type equal to “EME-1.0”:

- The GET request for the license includes in the body the JSON license request format defined in [encrypted-media] section 9.1.3. The license request MAY also include additional authentication elements such as access token, device or user ID.
- The response from the license server includes in the body the Clear Key license in the format defined in [encrypted-media] section 9.1.4 if the device is entitled to receive the Content Keys.

Clear Key licenses SHALL NOT be used to manage a key and `KID` that is also used by a DRM system. The use of an unprotected DRM key risks the security of DRM systems using that key, and violates the terms of use of most DRM systems.

Conformance§

Conformance requirements are expressed with a combination of descriptive assertions and RFC 2119 terminology. The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in the normative parts of this document are to be interpreted as described in RFC 2119. However, for readability, these words do not appear in all uppercase letters in this specification.

All of the text of this specification is normative except sections explicitly marked as non-normative, examples, and notes. [RFC2119]

Examples in this specification are introduced with the words “for example” or are set apart from the normative text with `class="example"`, like this:

> EXAMPLE 1
> This is an example of an informative example.

Informative notes begin with the word “Note” and are set apart from the normative text with `class="note"`, like this:

> Note, this is an informative note.

References§

Normative References§

[EME-INITDATA-CENC]
David Dorwin; et al. "cenc" Initialization Data Format. 15 September 2016. NOTE. URL: https://www.w3.org/TR/eme-initdata-cenc/

[ENCRYPTED-MEDIA]
David Dorwin; et al. Encrypted Media Extensions. 18 September 2017. REC. URL: https://www.w3.org/TR/encrypted-media/

[MPEG4]
ISO/IEC 14496-12: ISO base media file format. ISO/IEC.

[MPEGCENC]
ISO/IEC 23001-7:2016 Preview Information technology -- MPEG systems technologies -- Part 7: Common encryption in ISO base media file format files. February 2016. URL: https://www.iso.org/standard/68042.html

[MPEGDASH]
ISO/IEC 23009-1:2014 Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats. URL: http://standards.iso.org/ittf/PubliclyAvailableStandards/c065274_ISO_IEC_23009-1_2014.zip

[MPEGDASH-IMPGUIDE]
ISO/IEC 23009-3:2014 Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 3: Implementation Guidelines. URL: http://standards.iso.org

[RFC2119]
S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. March 1997. Best Current Practice. URL: https://tools.ietf.org/html/rfc2119

[RFC8446]
E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. August 2018. Proposed Standard. URL: https://tools.ietf.org/html/rfc8446

Informative References§

[MIXED-CONTENT]

Mike West. Mixed Content. 2 August 2016. CR. URL: https://www.w3.org/TR/mixed-content/

↑

→