

Media Over QUIC
Internet-Draft
Intended status: Informational
Expires: 5 December 2026

W. Law
Akamai
3 June 2026

CMSF- a CMAF compliant implementation of MOQT Streaming Format
draft-ietf-moq-cmsf-01

Abstract

This document updates MOQT Streaming Format by defining optional syntax and semantics for carrying CMAF-packaged media.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at <https://moq-wg.github.io/cmsf/draft-wilaw-moq-cmsf.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-ietf-moq-cmsf/>.

Discussion of this document takes place on the Media Over QUIC Working Group mailing list (<mailto:moq@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/moq/>. Subscribe at <https://www.ietf.org/mailman/listinfo/moq/>.

Source for this draft and an issue tracker can be found at <https://github.com/moq-wg/cmsf>.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 5 December 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. MSF Extension	3
3. CMAF Packaging	3
3.1. Initialization headers	3
3.2. Switching sets and tracks	3
3.3. Object Packaging	4
3.4. Group Packaging	4
3.5. Catalog description	4
3.5.1. CMAF packaging type	4
3.5.2. Max SAP starting types	5
3.6. Event Timelines	5
3.6.1. SAP Type timeline	5
3.6.2. SAP-type timeline track example	6
4. Content Protection	7
4.1. Content Protection catalog fields	8
4.1.1. Content Protections	8
4.1.2. Content Protection Reference IDs	11
4.2. Initialization data for protected tracks	11
4.3. ClearKey content protection	11
5. Catalog Examples	12
5.1. Simulcast video tracks - 3 alternate video qualities along with audio	12
5.2. DRM-protected video with audio	14
5.3. ClearKey-protected video	16
6. Conventions and Definitions	18
7. Security Considerations	18
8. IANA Considerations	18
9. References	18
9.1. Normative References	18
9.2. Informative References	19
Acknowledgments	19
Author's Address	19

1. Introduction

CMAF compliant MOQT Streaming Format (CMSF) is a media format designed to deliver CMAF [CMAF] and LOC [LOC] compliant media content over MOQ Transport (MOQT) [MoQTransport]. CMSF extends MSF and retains all the scope, capabilities and features of MSF including the catalog format, timeline, ABR switching and LOC support. CMSF is targeted at real-time and interactive levels of live latency, as well as VOD content.

This document describes version 1 of the CMSF streaming format.

2. MSF Extension

All of the specifications, requirements, and terminology defined in [MSF] apply to implementations of this extension unless explicitly noted otherwise in this document.

3. CMAF Packaging

3.1. Initialization headers

A CMAF header is a sequence of CMAF constrained ISO BMFF boxes that do not reference any media samples, but are associated with a CMAF track and are necessary for initializing the decoding of the subsequent CMAF fragments.

The header for a given MOQT Track MUST be added to the catalog by generating an initDataList entry with a "type" of "inline", a "data" value of the header encoded using [BASE64] and an "id" which is unique across the initDataList items. That "id" value is then inserted as the value of an "initRef" property in all tracks which use that header for initialization.

3.2. Switching sets and tracks

This specification defines a direct mapping between CMAF Tracks ([CMAF] Sect 3.2.1) and MOQT tracks ([MoQTransport] Sect 2.3).

CMAF switching sets are a set of one or more CMAF tracks (3.2.1), where each track is an alternative encoding of the same source content and are constrained to enable seamless track switching (3.3.9).

Each CMAF track in a switching set MUST be transmitted as a separate MOQT Track. The catalog entry for each of these tracks in the switching set MUST carry a Alternate group (altGroup) key with a common value.

The MOQT Group numbers within these switching set tracks MUST be media time-aligned. Mandating the track being media time-aligned requires that the presentation time of the first media sample contained within the first MOQT Object of each MOQT Group is identical.

3.3. Object Packaging

The payload of each Object is subject to the following requirements:

- * MUST contain at least one Movie Fragment Box (moof) followed by a Media Data Box (mdat). This is equivalent to requiring that each Object hold at least one CMAF Chunk. The Media Fragment Box (moof) MUST contain a Movie Fragment Header Box (mfhd) and Track Box (trak) with a Track ID (track_ID) matching a Track Box in the initialization fragment.
- * MAY contain multiple successive CMAF Chunks.
- * MUST contain a single track.

3.4. Group Packaging

Each MOQT Group

- * MUST begin with an Object containing a stream access point (SAP) type 1 or 2.
- * MUST contain one or more contiguous independently coded sequences of media samples.
- * The Group boundary MUST align with a CMAF Fragment boundary. CMAF Fragments and CMAF Chunks MUST NOT span Groups.

3.5. Catalog description

3.5.1. CMAF packaging type

This specification extends the allowed packaging values defined in [MSF] to include one new entry, as defined in Table 1 below:

Name	Value	Reference
CMAF	cmf	This RFC

Table 1

Every Track entry in a CMSF catalog carrying CMAF-packaged media data MUST declare a "packaging" type value of "cmaf".

3.5.2. Max SAP starting types

This specification adds two track-level catalog fields, as defined in Table 2 below:

Field	Name	Definition
Max Group SAP starting type	maxGrpSapStartingType	Section 3.5.2.1
Max Object SAP starting type	maxObjSapStartingType	Section 3.5.2.2

Table 2

3.5.2.1. Max Group SAP starting type

Location: T Required: Optional JSON Type: Number

A number indicating the maximum SAP type the MOQT Groups in the track start with.

3.5.2.2. Max Object SAP starting type

Location: T Required: Optional JSON Type: Number

A number indicating the maximum SAP type the MOQT Objects in the track start with.

3.6. Event Timelines

3.6.1. SAP Type timeline

CMSF defines a special instance of an Event Timeline track, termed the SAP Type timeline track. Its purpose is to convey information about the distribution of Stream Access Point types and their associated Earliest Presentation Times.

In the catalog, the SAP-type timeline track MUST include a 'packaging' value of 'eventtimeline' and MUST include an 'eventType' value of 'org.ietf.moq.cmsf.sap'.

In the SAP Type timeline JSON payload:

- * The index reference MUST be '1' for Location
- * The data field is a JSON Array containing two integers. The first integer defines SAP type with an allowed value of 0,1,2 or 3. The value 0 indicates that the Object does not start with an ISOBMFF stream access point. The value equal to 1, 2, or 3 indicates that the Object begins with a stream access point of SAP type 1, 2, or 3, respectively. When the Object is the first Object in the Group, the value MUST be equal to 1 or 2. The second integer defines the earliest media presentation timestamp, rounded to the nearest millisecond, of all media samples in the Object defined by the Location of that record.

3.6.2. SAP-type timeline track example

This shows an example of 30-fps HEVC-encoded content, in which each 4s Group begins with SAP-type 2 (i.e., the first picture in the Group is an IDR picture, while there may be one or more pictures in the Group following the IDR picture in decoding order but preceding it in output order). After 2 seconds in each Group, there is a SAP-type 3, i.e., a CRA picture, which is associated with one or more Random Access Skipped Leading (RASL) pictures. A small buffer of frames (10 frames at 30 fps) is skipped/discarded (RASL pictures) when the streaming session starts from the SAP-type 3 location. In this example, the EPT is the presentation time of the first picture after the RASL pictures in decoding order; all pictures after the RASL pictures can be fully correctly decoded and are thus presentable when the streaming session starts from the SAP-type 3 location. Note that if the streaming session starts from the start of the Group, then these RASL pictures can be fully correctly decoded and are thus presentable.

```
[
  {
    "l": [0,0],
    "data": [2,0]
  },
  {
    "l": [0,60],
    "data": [3,2100]
  },
  {
    "l": [1,0],
    "data": [2,4000]
  },
  {
    "l": [1,60],
    "data": [3,6100]
  }
]
```

4. Content Protection

CMSF supports content protection using Common Encryption [CENC] for CMAF-packaged media. This enables interoperability with existing DRM ecosystems by reusing the signaling model established by DASH [DASH] and the DASH-IF Encryption and Content Protection (ECCP) guidelines [DASHIF-ECCP].

Content protection in CMSF differs from the encryption scheme defined in [MSF] Section 4.3. While MSF defines an end-to-end encryption mechanism using MoQ Secure Objects for LOC-packaged content, CMSF uses ISO Common Encryption [CENC] applied at the CMAF media layer. In CMSF, the media samples within CMAF chunks are encrypted as specified by [CENC], and the DRM signaling is carried in the catalog rather than in per-object headers.

A key advantage of CENC-based content protection is that decryption can be delegated to a Content Decryption Module (CDM) operating at a deeper system level, potentially in hardware or a trusted execution environment. This enables robust content protection through commercial DRM systems such as Widevine, PlayReady, and FairPlay Streaming, where the decryption keys and decrypted media are handled within a secure pipeline that is not accessible to the application layer. This hardware-level protection is a requirement for high-value content distribution and is not achievable with application-layer encryption schemes alone.

4.1. Content Protection catalog fields

Content protection information is signaled in the catalog at two levels: a root-level contentProtections array containing DRM system descriptions, and track-level references to those descriptions.

4.1.1. Content Protections

Location: R Required: Optional JSON Type: Array

A JSON array of Content Protection objects at the root level of the catalog. Each object describes a single DRM system configuration. Content protection information MUST NOT be duplicated at the track level; all tracks reference the root-level entries.

Each Content Protection object contains the following fields:

4.1.1.1. Reference ID

Location: CP Required: Required JSON Type: String

A unique identifier for this content protection entry. Track entries reference this value via the contentProtectionRefIDs field.

4.1.1.2. Default KIDs

Location: CP Required: Required JSON Type: Array of Strings

An array of default Key IDs (KIDs) expressed as UUID strings in the format "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx". These correspond to the default_KID values defined in [CENC] Section 5. When multiple KIDs are present, each identifies a key used to encrypt a different set of samples (e.g., separate keys for different track types).

4.1.1.3. Scheme

Location: CP Required: Required JSON Type: String

The Common Encryption protection scheme as defined in [CENC]. Allowed values are:

Name	Value	Description
CENC	cenc	AES-CTR full sample and sub-sample encryption
CBCS	cbcs	AES-CBC pattern-based encryption with subsample

Table 3

The RECOMMENDED scheme for CMSF is "cbcs" as it provides better compatibility with hardware decoders and aligns with DASH-IF ECCP recommendations [DASHIF-ECCP].

4.1.1.4. DRM System

Location: CP Required: Required JSON Type: Object

An object describing the DRM system associated with this content protection entry. It contains the following fields:

4.1.1.4.1. System ID

Location: DS Required: Required JSON Type: String

The DRM System ID expressed as a UUID string. Well-known system IDs include:

DRM System	System ID
Widevine	edef8ba9-79d6-4ace-a3c8-27dcd51d21ed
PlayReady	9a04f079-9840-4286-ab92-e65be0885f95
FairPlay	94ce86fb-07ff-4f43-adb8-93d2fa968ca2
ClearKey	1077efec-c0b2-4d02-ace3-3c1e52e2fb4b

Table 4

The ClearKey system ID follows the W3C EME specification and corresponds to the DASH-IF ECCP "Explicit Clear Key Content Protection" mechanism [DASHIF-ECCP].

4.1.1.4.2. License URL

Location: DS Required: Optional JSON Type: Object

An object containing the URL of the license acquisition service. The object has the following fields:

- * url (String, Required): The URL of the license server.
- * type (String, Optional): The license protocol type (e.g., "EME-1.0").

4.1.1.4.3. Certificate URL

Location: DS Required: Optional JSON Type: Object

An object containing the URL of the DRM certificate service. This field is REQUIRED for DRM systems that require a server certificate (e.g., FairPlay Streaming). The object has the following fields:

- * url (String, Required): The URL of the certificate server.
- * type (String, Optional): The MIME type of the certificate resource (e.g., "application/pkcs7-mime", "application/x-x509-ca-cert"). This corresponds to the certType attribute defined in [DASHIF-ECCP].

4.1.1.4.4. Authorization URL

Location: DS Required: Optional JSON Type: Object

An object containing the URL of an authorization service. The object has the following fields:

- * url (String, Required): The URL of the authorization server.
- * type (String, Optional): The authorization protocol type.

4.1.1.4.5. PSSH

Location: DS Required: Optional JSON Type: String

A Base64-encoded [BASE64] Protection System Specific Header (PSSH) box as defined in [CENC] Section 8.1. The PSSH box contains DRM system-specific initialization data needed by the client to acquire a license. This field SHOULD be present for DRM systems that require PSSH data (e.g., Widevine, PlayReady).

4.1.1.4.6. Robustness

Location: DS Required: Optional JSON Type: String

A string indicating the minimum robustness level required by the DRM system. The interpretation of this value is DRM system-specific.

4.1.2. Content Protection Reference IDs

Location: T Required: Optional JSON Type: Array of Strings

An array of Reference ID strings (see Section 4.1.1.1) identifying which content protection entries from the root-level contentProtections array apply to this track. When this field is present, the track content is encrypted using Common Encryption [CENC] and the subscriber MUST acquire appropriate licenses before decryption.

When this field is absent, the track content is not protected by Common Encryption.

4.2. Initialization data for protected tracks

For protected CMAF tracks, the initialization data (carried in the catalog initDataList array (as defined in Section 3.1) MUST include the Protection Scheme Information Box ('sinf') containing the Scheme Type Box ('schm') and Scheme Information Box ('schi') with the Track Encryption Box ('tenc') as specified in [CENC] Section 6. This enables the subscriber to determine the encryption parameters (default_isProtected, default_Per_Sample_IV_Size, default_KID) from the initialization segment.

4.3. ClearKey content protection

For testing and development scenarios, CMSF supports ClearKey content protection using the W3C EME ClearKey mechanism. ClearKey uses the Common System ID "1077efec-c0b2-4d02-ace3-3cle52e2fb4b" and follows the DASH-IF ECCP Explicit Clear Key Content Protection (ECCP) model [DASHIF-ECCP].

When using ClearKey:

- * The systemID MUST be set to "1077efec-c0b2-4d02-ace3-3cle52e2fb4b".
- * The laURL field SHOULD contain the URL of a ClearKey license server that implements the EME ClearKey protocol.

- * The optional pssh field SHOULD contain a Base64-encoded PSSH box with version 1, the Common System ID, and the KID(s) in the KID list.

5. Catalog Examples

The following section provides non-normative JSON examples of various catalogs compliant with this draft.

5.1. Simulcast video tracks - 3 alternate video qualities along with audio

This example shows catalog for a media producer capable of sending 3 time-aligned video tracks for high definition, low definition and medium definition video qualities, along with an audio track.

```
{
  "version": "1",
  "generatedAt": 1746104606044,
  "tracks": [
    {
      "name": "hd",
      "renderGroup": 1,
      "packaging": "cmf",
      "isLive": true,
      "targetLatency": 2000,
      "initRef": "init-hd",
      "role": "video",
      "codec": "avc1.640028",
      "width": 1920,
      "height": 1080,
      "bitrate": 5000000,
      "framerate": 30,
      "altGroup": 1
    },
    {
      "name": "md",
      "renderGroup": 1,
      "packaging": "cmf",
      "isLive": true,
      "targetLatency": 2000,
      "initRef": "init-md",
      "role": "video",
      "codec": "avc1.64001e",
      "width": 720,
      "height": 640,
      "bitrate": 3000000,
      "framerate": 30,

```

```
    "altGroup":1
  },
  {
    "name": "sd",
    "renderGroup": 1,
    "packaging": "cmf",
    "isLive": true,
    "targetLatency": 2000,
    "initRef": "init-sd",
    "role": "video",
    "codec": "avc1.64000d",
    "width": 192,
    "height": 144,
    "bitrate": 500000,
    "framerate": 30,
    "altGroup": 1
  },
  {
    "name": "audio",
    "renderGroup": 1,
    "packaging": "cmf",
    "isLive": true,
    "targetLatency": 2000,
    "initRef": "init-audio",
    "role": "audio",
    "codec": "mp4a.40.5",
    "samplerate": 48000,
    "channelConfig": "2",
    "bitrate": 67071
  }
],
"initDataList": [
  {
    "id": "init-hd",
    "type": "inline",
    "data": "AAAAHGZ0eXBjbWYyAAAAAGNtZjJpc282bXA0MQ..."
  },
  {
    "id": "init-md",
    "type": "inline",
    "data": "AAAAHGZ0eXBjbWYyAAAAAGNtZjJpc282bXA0MQ..."
  },
  {
    "id": "init-sd",
    "type": "inline",
    "data": "AAAAHGZ0eXBjbWYyAAAAAGNtZjJpc282bXA0MQ..."
  }
]
```

```

        "id": "init-audio",
        "type": "inline",
        "data": "AAAAHGZ0eXBjbWYyAAAAAGNtZjJpc282bXA0MQ..."
    }
  ]
}

```

5.2. DRM-protected video with audio

This example shows a catalog for a single DRM-protected video track and an unprotected audio track. The video track is encrypted using CBCS and references three content protection entries: Widevine, PlayReady, and FairPlay.

```

{
  "version": "1",
  "generatedAt": 1746104606044,
  "contentProtections": [
    {
      "refID": "1",
      "defaultKID": [
        "01234567-89ab-cdef-0123-456789abcdef"
      ],
      "scheme": "cbcs",
      "drmSystem": {
        "systemID": "edef8ba9-79d6-4ace-a3c8-27dcd51d21ed",
        "laURL": {
          "url": "https://widevine-license.example.com/proxy"
        },
        "pssh": "AAAAP3Bzc2gAAAAA7e+LqXnWSs6jy..."
      }
    },
    {
      "refID": "2",
      "defaultKID": [
        "01234567-89ab-cdef-0123-456789abcdef"
      ],
      "scheme": "cbcs",
      "drmSystem": {
        "systemID": "9a04f079-9840-4286-ab92-e65be0885f95",
        "laURL": {
          "url": "https://playready-license.example.com/auth"
        },
        "pssh": "AAACvnBzc2gAAAAAmgTweZhAQoar..."
      }
    },
    {
      "refID": "3",

```

```
"defaultKID": [
  "01234567-89ab-cdef-0123-456789abcdef"
],
"scheme": "cbcs",
"drmSystem": {
  "systemID": "94ce86fb-07ff-4f43-adb8-93d2fa968ca2",
  "laURL": {
    "url": "https://fps-license.example.com/api/licenses"
  },
  "certURL": {
    "url": "https://fps-license.example.com/cert"
  }
}
},
],
"tracks": [
  {
    "name": "video_protected",
    "packaging": "cmaf",
    "isLive": true,
    "buffers": {"target": 1500},
    "role": "video",
    "renderGroup": 1,
    "altGroup": 1,
    "initRef": "1",
    "codec": "avc3.4D401F",
    "framerate": 25,
    "bitrate": 581905,
    "width": 1280,
    "height": 720,
    "contentProtectionRefIDs": ["1", "2", "3"]
  },
  {
    "name": "audio",
    "packaging": "cmaf",
    "isLive": true,
    "buffers": {"target": 1500},
    "role": "audio",
    "renderGroup": 1,
    "initRef": "2",
    "codec": "mp4a.40.5",
    "samplerate": 48000,
    "channelConfig": "2",
    "bitrate": 67071
  }
],
"initDataList": [
  {
```

```
    "id": "1",
    "type": "inline",
    "data": "AAAAHGZ0eXBjbWYyAAAAAGNtZjJpc282bXA0MQAA..."
  },
  {
    "id": "2",
    "type": "inline",
    "data": "AAAAHGZ0eXBjbWYyAAAAAGNtZjJpc282bXA0MQAA..."
  }
]
```

5.3. ClearKey-protected video

This example shows a catalog using ClearKey content protection following the ECCP model, suitable for testing and development.


```
{
  "version": "1",
  "generatedAt": 1746104606044,
  "contentProtections": [
    {
      "refID": "1",
      "defaultKID": [
        "01234567-89ab-cdef-0123-456789abcdef"
      ],
      "scheme": "cenc",
      "drmSystem": {
        "systemID": "1077efec-c0b2-4d02-ace3-3c1e52e2fb4b",
        "laURL": {
          "url": "https://clearkey-server.example.com/clearkey",
          "type": "EME-1.0"
        },
        "pssh": "AAAANHBzc2gBAAAAEHfv7MCyTQKs4..."
      }
    }
  ],
  "tracks": [
    {
      "name": "video",
      "packaging": "cmf",
      "isLive": true,
      "buffers": {"target": 1500},
      "role": "video",
      "renderGroup": 1,
      "initRef": "init-video",
      "codec": "avc1.640028",
      "framerate": 30,
      "bitrate": 5000000,
      "width": 1920,
      "height": 1080,
      "contentProtectionRefIDs": ["1"]
    }
  ],
  "initDataList": [
    {
      "id": "init-video",
      "type": "inline",
      "data": "AAAAHGZ0eXBjbWYyAAAAAGNtZjJpc282bXA0MQAA..."
    }
  ]
}
```

6. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

7. Security Considerations

TODO Security

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

- [BASE64] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <<https://www.rfc-editor.org/rfc/rfc4648>>.
- [CENC] Standardization, I. O. for., "Information technology — MPEG systems technologies — Part 7: Common encryption in ISO base media file format files", ISO/IEC 23001-7:2024, August 2024.
- [CMAF] Standardization, I. O. for., "Information technology — Multimedia application format (MPEG-A) — Part 19: Common media application format (CMAF) for segmented media", October 2021.
- [LOC] Zanaty, M., Nandakumar, S., and P. Thatcher, "Low Overhead Media Container", Work in Progress, Internet-Draft, draft-ietf-moq-loc-02, 15 March 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-moq-loc-02>>.
- [MoQTransport] Nandakumar, S., Vasiliev, V., Swett, I., and A. Frindell, "Media over QUIC Transport", Work in Progress, Internet-Draft, draft-ietf-moq-transport-18, 12 May 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-moq-transport-18>>.

- [MSF] Law, W. and S. Nandakumar, "MOQT Streaming Format", Work in Progress, Internet-Draft, draft-ietf-moq-msf-01, 2 June 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-moq-msf-01>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.

9.2. Informative References

- [DASH] Standardization, I. O. for., "Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats.", ISO/IEC 23009-1:2022, August 2022.
- [DASHIF-ECCP] Forum, D. I., "DASH-IF Implementation Guidelines: Encryption and Content Protection (ECCP)", 2023, <<https://dashif.org/docs/DASH-IF-ECCP-v1.0.0.pdf>>.

Acknowledgments

TODO acknowledge.

Author's Address

Will Law
Akamai
Email: wilaw@akamai.com