INTERNATIONAL STANDARD

ISO/IEC 23001-7

Second edition
2015-04-01

Information technology — MPEG systems technologies —

Part 7:
Common encryption in ISO base media file format files

Technologies de l'information — Technologies des systèmes MPEG —

Partie 7: Cryptage commun des fichiers au format de fichier de médias de la base ISO

Reference number
ISO/IEC 23001-7:2015(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT), see the following URL: Foreword — Supplementary information.

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This second edition cancels and replaces the first edition which has been technically revised.

ISO/IEC 23001 consists of the following parts, under the general title *Information technology — MPEG systems technologies*:

- *Part 1: Binary MPEG format for XML*
- *Part 2: Fragment Request Units*
- *Part 3: XML IPMP messages*
- *Part 4: Codec configuration representation*
- *Part 5: Bitstream Syntax Description Language (BSDL)*
- *Part 7: Common encryption in ISO base media file format files*
- *Part 8: Coding-independent code points*
- *Part 9: Common encryption of MPEG-2 transport streams*

The following parts are under preparation:

- *Part 10: Carriage of timed metadata metrics of media in ISO base media file format*
- *Part 11: Green metadata*

Introduction

The common encryption protection scheme specifies standard encryption and key mapping methods that can be utilized to enable decryption of the same file using different digital rights management (DRM) and key management systems. The schemes operates by defining a common format for the encryption related metadata necessary to decrypt the protected streams, yet leaves the details of rights mappings, key acquisition and storage, DRM compliance rules, etc., up to the DRM system or systems supporting the common encryption scheme. For instance, DRM systems supporting the 'cenc' protection scheme must support identifying the decryption key via 'cenc' key identifier (KID) but how the DRM system locates the identified decryption key is left to a DRM-specific method. DRM specific information such as licenses or rights and license/rights acquisition information can be stored in an ISO Base Media file using a Protection System Specific Header box ('pssh'). Each instance of this box stored in the file corresponds to one applicable DRM system. DRM licenses/rights need not be stored in the file in order to look up a key using KID values stored in the file and decrypt media samples using the encryption parameters stored in each track. The second edition of this part of ISO/IEC 23001 also describes XML representation of common encryption parameters in MPEG DASH Media Presentation Description Documents.

Information technology — MPEG systems technologies —

Part 7: Common encryption in ISO base media file format files

1 Scope

This part of ISO/IEC 23001 specifies a common encryption format for use in any file format based on ISO/IEC 14496-12.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 14496-12, *Information technology — Coding of audio-visual objects — Part 12: ISO base media file format*

ISO/IEC 14496-15, *Information technology — Coding of audio-visual objects — Part 15: Carriage of network abstraction layer (NAL) unit structured video in ISO base media file format*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

NOTE Words used as defined terms and normative terms (SHALL, SHOULD, MAY) are written in upper case to distinguish them from the same word intending its dictionary definition.

3.1
ISO Base Media File
file conforming to the file format described in ISO/IEC 14496-12 in which the techniques in ISO/IEC 23001-7 can be used

3.2
network abstraction layer
NAL
NAL syntax element specified by a network abstraction layer specification such as AVC or HEVC

3.3
NAL unit
syntax structure containing an indication of the type of data to follow and bytes containing that data in the form of an RBSP interspersed as necessary with emulation prevention bytes

3.4
NAL structured video
video sample description format specified by ISO/IEC 14496-15

4 Abbreviated terms

For the purposes of this International Standard, the following abbreviated terms apply.

AES	Advanced Encryption Standard as specified in Federal Information Processing Standards Publication 197, FIPS-197
AES-CTR	AES Counter Mode as specified in *Recommendation of Block Cipher Modes of Operation*, NIST, NIST Special Publication 800-38A
AES-CBC	AES Cipher-Block Chaining Mode as specified in *Recommendation of Block Cipher Modes of Operation*, NIST, NIST Special Publication 800-38A
AVC	Advanced Video Coding as specified in ISO/IEC 14496-10
HEVC	High Efficiency Video Coding as specified in ISO/IEC 23008-2
NAL	Network Abstraction Layer, as specified in ISO/IEC 14496-10 and ISO/IEC 23008-2
URN	Unique Resource Name
UUID	Universally Unique Identifier

5 Scheme Signaling

Scheme signaling SHALL conform to ISO/IEC 14496-12. As defined in ISO/IEC 14496-12, the sample entry is transformed and a Protection Scheme Information Box (`‘sinf’`) is added to the standard sample entry in the Sample Description Box to denote that a stream is encrypted. The Protection Scheme Information Box SHALL contain a Scheme Type Box (`‘schm’`) so that the scheme is identifiable. The Scheme Type Box SHALL have the following additional constraints:

— The `scheme_type` field is set to a value of `‘cenc’` (Common Encryption). As an optional alternative, AES-CBC may be used in which case the scheme_type field is set to the value `cbc1’.

— The `scheme_version` field is set to 0x00010000 (Major version 1, Minor version 0).

The Protection Scheme Information Box SHALL also contain a Scheme Information Box (‘`schi’`). The Scheme Information Box SHALL have the following additional constraint:

— The Scheme Information Box SHALL contain a Track Encryption Box (‘`tenc’`), describing the default encryption parameters for the track.

6 Overview of Encryption Metadata

The encryption metadata defined by the ‘cenc’ Common Encryption Scheme can be categorized as follows:

— Protection System Specific Data – this data is opaque to the ‘cenc’ Common Encryption Scheme. This gives protection systems a place to store their own data using a common mechanism. This data is contained in the `ProtectionSystemSpecificHeaderBox` described in 9.1.

— Common encryption information for a media track – this includes default values for the key identifier (KID), initialization vector size, and encryption flag. This data is contained in the `TrackEncryptionBox` described in section 9.1.

— Common encryption information for groups of media samples – this includes overrides to the track level defaults for key identifier (KID), initialization vector size, and encryption flag. This allows groups of samples within the track to use different keys, a mix of clear and encrypted content, etc. This data is contained in a `SampleGroupDescriptionBox` (`‘sgpd’`) that is referenced by a `SampleToGroupBox` (`‘sbgp’`). See 7 for further details.

— Encryption information for individual media samples – this includes initialization vectors and, if required, sub sample encryption data. This data is sample auxiliary information, referenced by using a SampleAuxiliaryInformationSizesBox ('saiz') and a SampleAuxiliaryInformationOffsetsBox ('saio'). See 8 for further details.

7 Encryption Parameters shared by groups of samples

Each sample in a protected track SHALL be associated with an IsEncrypted flag, IV_Size, and KID. This can be accomplished by (a) relying on the default values in the TrackEncryptionBox (see 9.2, or (b) specifying the parameters by sample group, or (c) using a combination of these two techniques. Encryption parameters defined at sample group level override the corresponding default parameter values defined in the TrackEncryptionBox.

When specifying the parameters by sample group, the SampleToGroupBox in the sample table or track fragment specifies which samples use which sample group description from the SampleGroupDescriptionBox. The format of the sample group description is based on the handler type for the track.

Tracks with a handler type of 'vide' SHALL use the CencSampleEncryptionInformationVideoGroupEntry sample group description structure, which has the following syntax:

```
aligned(8) class CencSampleEncryptionInformationVideoGroupEntry
    extends VisualSampleGroupEntry( 'seig' )
{
    unsigned int(24)       IsEncrypted;
    unsigned int(8)        IV_size;
    unsigned int(8)[16]    KID;
}
```

Similarly, tracks with a handler type of 'soun' SHALL use the CencSampleEncryptionInformationAudioGroupEntry sample group description structure, which has the following syntax:

```
aligned(8) class CencSampleEncryptionInformationAudioGroupEntry
    extends AudioSampleGroupEntry( 'seig' )
{
    unsigned int(24)       IsEncrypted;
    unsigned int(8)        IV_size;
    unsigned int(8)[16]    KID;
}
```

NOTE Sample Group Entries with identical structure should be defined if protection of other media types is needed.

These structures use a common semantic for their fields as follows:

IsEncrypted is the flag which indicates the encryption state of the samples in the sample group. See the IsEncrypted field in 10.2 for further details.

IV_size is the Initialization Vector size in bytes for samples in the sample group. See the IV_size field in 10.2 for further details.

KID is the key identifier used for samples in the sample group. See the KID field in 10.2 for further details.

In order to facilitate the addition of future optional fields, clients SHALL ignore additional bytes after the fields defined in the CencSampleEncryption group entry structures.

8 Common Encryption Sample Auxiliary Information

Each encrypted sample in a protected track SHALL have an Initialization Vector associated with it. Further, each encrypted sample in protected NAL structured video tracks SHALL conform to ISO/IEC 14496-15 and SHALL use the subsample encryption scheme specified in 10.6.2, which requires subsample encryption data. Both initialization vectors and subsample encryption data are provided as Sample Auxiliary Information with aux_info_type equal to 'cenc' and aux_info_type_parameter equal to 0. For tracks protected using the 'cenc' scheme, the default value for aux_info_type is equal to 'cenc' and the default value for the aux_info_type_parameter is 0 so content MAY be created omitting these optional fields. Storage of sample auxiliary information SHALL conform to ISO/IEC 14496-12.

The format of the sample auxiliary information for samples with this type SHALL be:

```
aligned(8) class CencSampleAuxiliaryDataFormat
{
   unsigned int(IV_size*8) InitializationVector;
   if ( sample_info_size > IV_size )
   {
      unsigned int(16) subsample_count;
      {
        unsigned int(16) BytesOfClearData;
        unsigned int(32) BytesOfEncryptedData;
      } [ subsample_count ]
   }
}
```

Where:

InitializationVector is the initialization vector for the sample. See the InitializationVector field in 10.2 for further details.

subsample_count is the count of subsamples for this sample. See the subsample_count field in 10.2 for further details.

BytesOfClearData is the number of bytes of clear data in this subsample. See the BytesofClearData field in 10.2 for further details.

BytesOfEncryptedData is the number of bytes of encrypted data in this subsample. See the BytesofEncryptedData field in 10.2 for further details.

If sub-sample encryption is not used (sample_info_size equals IV_size), then the entire sample is encrypted (see 10.5 for further details). In this case, all auxiliary information will have the same size and hence the default_sample_info_size of the SampleAuxiliaryInformationSizes box will be equal to the IV_Size of the initialization vectors.

Note, however, that even if subsample encryption is used, the size of the sample auxiliary information MAY be the same for all of the samples (if all of the samples have the same number of subsamples) and the default_sample_info_size MAY be used.

8.1 Sample Encryption Information Box for Storage of Sample Auxiliary Information

An optional storage location for Sample Auxiliary Information is the Sample Encryption Information Box ('senc') specified below.

8.1.1 Sample Encryption Box ('senc')

Box Type 'senc'

Container Track Fragment Box ('traf') or Track Box ('trak')

Mandatory No

Quantity Zero or one

The Sample Encryption Box contains sample auxiliary information, including the initialization vector for each sample, and clear and encrypted byte ranges of partially encrypted video samples (“subsample encryption”). It MAY be used when samples in a track or track fragment are encrypted. Storage of ‘senc’ in a Track Fragment Box makes the necessary Sample Auxiliary Information accessible within the movie fragment for all contained samples in order to make each track fragment independently decryptable; for instance, when movie fragments are delivered as DASH Media Segments.

8.1.2 Syntax

```
aligned(8) class SampleEncryptionBox
   extends FullBox(‘senc’, version=0, flags=0)
{
   unsigned int(32)  sample_count;
   {
      unsigned int(IV_size*8)  InitializationVector;
      if (flags & 0x000002)
      {
         unsigned int(16)  subsample_count;
         {
            unsigned int(16)  BytesOfClearData;
            unsigned int(32)  BytesOfEncryptedData;
         } [ subsample_count ]
      }
   }[ sample_count ]
}
```

8.1.2.1 Semantics

- flags is inherited from the FullBox structure. The SampleEncryptionBox currently supports the following bit values:

  - 0x2 – UseSubSampleEncryption

  - If the UseSubSampleEncryption flag is set, then the track fragment that contains this Sample Encryption Box SHALL use the sub-sample encryption as described in Section 9.6. When this flag is set, sub-sample mapping data follows each InitilizationVector. The sub-sample mapping data consists of the number of sub-samples for each sample, followed by an array of values describing the number of bytes of clear data and the number of bytes of encrypted data for each sub-sample.

- sample_count is the number of encrypted samples in the containing track or track fragment. This value SHALL be either zero (0) or the total number of samples in the track or track fragment.

- InitializationVector SHALL conform to the definition specified in Section 10.2. Only one IV_size SHALL be used within a file, or IV_size SHALL be zero when a sample is unencrypted. Selection of InitializationVector values SHOULD follow the recommendations of Section 10.3.

- subsample_count SHALL conform to the definition specified in Section 10.2.

- BytesOfClearData SHALL conform to the definition specified in Section 10.2.

- BytesOfEncryptedData SHALL conform to the definition specified in Section 10.2.

9 Box Definitions

9.1 Protection System Specific Header Box

9.1.1 Definition

Box Type: `pssh’

Container: Movie (‘moov’) or Movie Fragment (‘moof’)

Mandatory: No

Quantity: Zero or more

This box contains information needed by a Content Protection System to play back the content. The data format is specified by the system identified by the ‘pssh’ parameter SystemID, and is considered opaque for the purposes of this specification. The collection of Protection System Specific Header boxes from the initial movie box, together with those in a movie fragment, SHALL provide all the required Content Protection System information to decode that fragment.

The data encapsulated in the Data field MAY be read by the identified Content Protection System client to enable decryption key acquisition and decryption of media data. For license/rights-based systems, the header information MAY include data such as the URL of license server(s) or rights issuer(s) used, embedded licenses/rights, embedded keys(s), and/or other protection system specific metadata.

A single file MAY be constructed to be playable by multiple key and digital rights management (DRM) systems, by including Protection System Specific Header boxes for each system supported. In order to find all of the Protection System Specific data that is relevant to a sample in the presentation readers SHALL:

- Examine all Protection System Specific Header boxes in the Movie Box and in the Movie Fragment Box associated with the sample (but not those in other Movie Fragment Boxes).
- Match the SystemID field in this box to the SystemID(s) of the DRM System(s) they support
- Match the KID associated with the sample (either from the default_KID field of the Track Encryption Box or the KID field of the appropriate sample group description entry) with one of the KID values in the Protection System Specific Header Box. Boxes without a list of applicable KID values, or with an empty list, SHALL be considered to apply to all KIDs in the file or movie fragment.

Protection System Specific Header data SHALL be associated with a sample based on a matching KID value in the ‘pssh’ and sample group description or default ‘tenc’ describing the sample. If a sample or set of samples is moved due to file defragmentation or refragmentation or removed by editing, then the associated Protection System Specific Header boxes for the remaining samples SHALL be stored following the above requirements.

9.1.2 Syntax

```
aligned(8) class ProtectionSystemSpecificHeaderBox extends FullBox(‘pssh’, version, flags=0)
{
   unsigned int(8)[16]   SystemID;
   if (version > 0)
   {
      unsigned int(32)    KID_count;
      {
         unsigned int(8)[16]  KID;
      } [KID_count];
   }
   unsigned int(32)    DataSize;
   unsigned int(8)[DataSize] Data;
}
```

9.1.3 Semantics

SystemID specifies a UUID that uniquely identifies the content protection system that this header belongs to.

KID_count specifies the number of KID entries in the following table. The value MAY be zero.

KID identifies a key identifier that the Data field applies to. If not set, then the Data array SHALL apply to all KIDs in the movie or movie fragment containing this box.

DataSize specifies the size in bytes of the Data member.

Data holds the content protection system specific data.

9.2 Track Encryption Box

9.2.1 Definition

Box Type: `tenc’

Container: Scheme Information Box (‘schi’)

Mandatory: No (Yes, for encrypted tracks)

Quantity: Zero or one

The TrackEncryptionBox contains default values for the IsEncrypted flag, IV_size, and KID for the entire track These values are used as the encryption parameters for the samples in this track unless over-ridden by the sample group description associated with a group of samples. For files with only one key per track, this box allows the basic encryption parameters to be specified once per track instead of being repeated per sample.

9.2.2 Syntax

```
aligned(8) class TrackEncryptionBox extends FullBox(‘tenc’, version=0, flags=0)
{
   unsigned int(24)      default_IsEncrypted;
   unsigned int(8)       default_IV_size;
   unsigned int(8)[16]   default_KID;
}
```

9.2.3 Semantics

default_IsEncypted is the encryption flag which indicates the default encryption state of the samples in the track. See the IsEncrypted field in 10.2 for further details.

default_IV_size is the default Initialization Vector size in bytes. See the IV_size field in 10.2 for further details.

default_KID is the default key identifier used for samples in this track. See the KID field in 10.2 for further details.

10 Encryption of Media Data

10.1 Encryption Schemes

Media data using the ‘cenc’ Protection Scheme SHALL use the *Advanced Encryption Standard*, Federal Information Processing Standards Publication 197, FIPS-197 published by the United States National Institute of Standards and Technology (NIST) using 128-bit keys in Counter Mode (AES-CTR), as specified in *Recommendation of Block Cipher Modes of Operation*, NIST, NIST Special Publication 800-38A. The ‘cenc’ scheme defines two elementary stream encryption formats, full sample encryption and subsample encryption. Full sample encryption is where the entire sample is encrypted as a single encryption unit whereas subsample encryption is where the sample is broken into smaller units each containing a clear area and an encrypted area. Encrypted NAL structured video tracks SHALL follow the subsample encryption scheme specified in 10.6.2 which defines an encryption scheme to allow access to NAL units and unencrypted NAL unit headers in an encrypted stream of NAL structured video samples specified in ISO/IEC 14496-15. All other types of tracks SHALL follow the scheme specified in 10.5, which defines a sample-based encryption scheme.

10.2 Field semantics

Within the sample groups and sample auxiliary information used by the common encryption scheme, the fields have the following semantics:

IsEncrypted is the identifier of the encryption state of the samples in the track or group of samples. This flag takes the following values:

0x0: Not encrypted

0x1: Encrypted (as signalled by the scheme_type field of the scheme type box ‘schm’, e.g. for ‘cenc’ this is AES-CTR)

0x000002 – 0xFFFFFF: Reserved

IV_size is the size in bytes of the InitializationVector field. Supported values:

0 – If the IsEncrypted flag is 0x0 (Not Encrypted).

8 – Specifies 64-bit initialization vectors.

16 – Specifies 128-bit initialization vectors.

KID is a key identifier that uniquely identifies the key needed to decrypt the associated samples within the scope of an application so that KID is sufficient to identify a separately stored license containing the key that was used to encrypt the content. This allows the identification of multiple encryption keys per file or track. Unencrypted samples in an encrypted track SHALL be identified by IsEncrypted flag of 0x0, an IV_size of 0x0, and a KID value of 0x0. It is strongly recommended to use UUIDs [1] as KIDs in order to satisfy the uniqueness requirement above across all applications.

InitializationVector specifies the initialization vector (IV) needed for decryption of a sample. For an IsEncrypted flag of 0x0, no initialization vectors are needed and the auxiliary information SHOULD have0 a size of 0, i.e. not be present.

For an IsEncrypted flag of 0x1

if the IV_size field is 16 then InitializationVector specifies the entire 128-bit IV value

If the IV_size field is 8, then its value is copied to bytes 0 to 7 of the Initialization Vector and bytes 8 to 15 of the Initialization Vector are set to zero. The IV_size field SHALL NOT be 0 when the IsEncrypted flag is 0x1.

For an IsEncrypted flag of 0x1 where the scheme_type field of the scheme type box is ‘cenc’ (i.e. AES-CTR), counter values SHALL be unique per KID. If an IV_size of 8 is used, then the InitializationVector values for a given KID SHALL be unique for each sample in all tracks and samples SHALL be less than 2^{64} blocks in length. If an IV_size of 16 is used, then initialization vectors SHALL have large enough numeric differences to prevent duplicate counter values for any encrypted block using the same KID.

subsample_count specifies the number of subsample encryption entries present for this sample. If present this field SHALL be greater than 0.

BytesOfClearData specifies the number of bytes of clear data at the beginning of this subsample encryption entry. (Note: this value may be zero if no clear bytes exist for this entry.)

BytesOfEncryptedData specifies the number of bytes of encrypted data following the clear data. (Note: this value may be zero if no encrypted bytes exist for this entry.)

The subsample encryption entries SHALL not include an entry with a zero value in both theBytesOfClearData field and in the BytesOfEncryptedData field unless the sample is zero bytes in length. The total length of allBytesOfClearData and BytesOfEncryptedData for a sample SHALL equal the length of the sample. Further, it is recommended that the subsample encryption entries be as compactly represented as possible. For example, instead of two entries with {15 clear, 0 encrypted}, {17 clear, 500 encrypted} use one entry of {32 clear, 500 encrypted}

10.3 Initialization Vectors

The initialization vector (IV) values for each sample are located in the Sample Auxiliary Information associated with the encrypted samples. See 10.2 for details on how initialization vectors are formed and stored.

It is recommended that applications applying encryption randomly generate the initialization vector for the first sample in the track using a cryptographically sound random number generator.

— For 64-bit (8-byte) IV_Sizes, initialization vectors for subsequent samples can be created by incrementing the initialization vector of the previous sample. Using a random starting value introduces entropy into the initialization vector values and incrementing for each sample processed ensures that each IV value is unique. The 64-bit initialization vector SHOULD be allowed to roll over from the maximum value (`0xFFFFFFFFFFFFFFFF`) to the minimum value (`0x0`) if the random starting position is close to the maximum value.

— For 128-bit (16-byte) IV_Sizes, initialization vectors for subsequent samples SHOULD be created by adding the block count of the previous sample to the initialization vector of the previous sample. Using a random starting value introduces entropy into the initialization vector values and incrementing by the block count of the previous sample ensures that each IV value is unique. Even though the block counter portion of the counter (bytes 8 to 15) is treated as an unsigned 64-bit number by the client as described in 10.4, it is recommended that the initialization vector is treated as a 128-bit number when calculating the next initialization vector from the previous one.

10.4 Counter Operation

AES-CTR mode is a block cipher that can encrypt arbitrary length data without need for padding. It operates by encrypting a counter block with the AES algorithm and then XOR-ing the output of AES with the data to encrypt or decrypt. The counter block used is constructed as described in 10.2. Of the 16 byte counter block, bytes 8 to 15 (i.e. the least significant bytes) are used as a simple 64 bit unsigned integer that is incremented by one for each subsequent block of sample data processed and is kept in network byte order. Note that if this integer reaches the maximum value (`0xFFFFFFFFFFFFFFFF`) in the case where a 128-bit (16-byte) IV_size is used, then incrementing it resets the block counter to zero (bytes 8 to 15) without affecting the other 64 bits of the counter (i.e. bytes 0 to 7).

10.5 Full Sample Encryption

In full sample encryption, the entire sample is encrypted. Figure 1 shows sample-based encryption using AES-CTR mode.

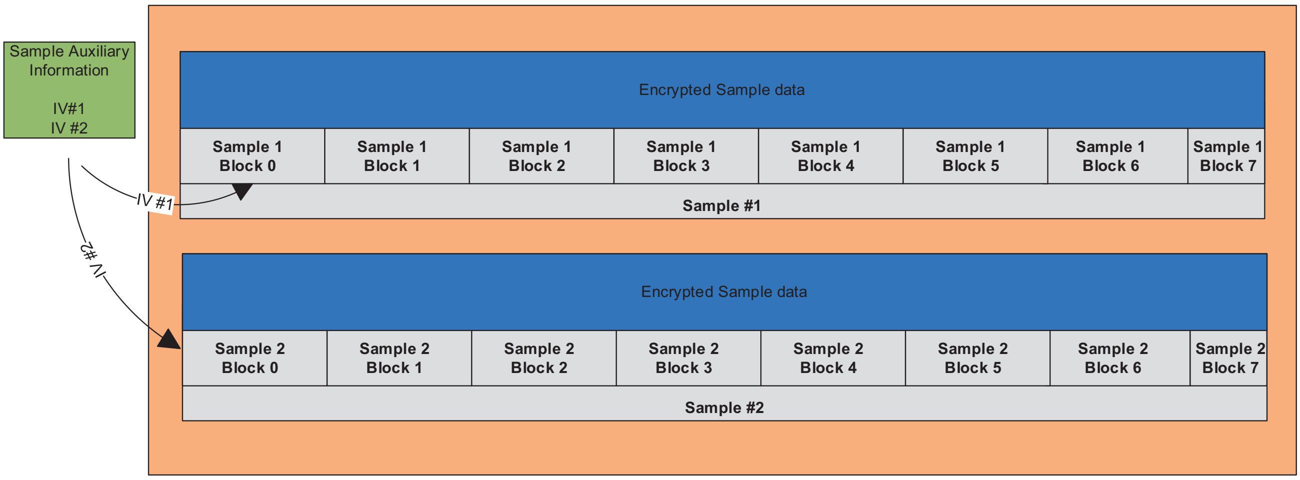

Figure 1 — Sample-based encryption for AES-CTR

Note that AES-CTR mode is a block cipher mode that acts like a stream cipher. Cipher blocks are shown to illustrate the underlying blocks used in generating the stream cipher (Block 7 is shown as smaller than 16 bytes to illustrate that CTR mode can encrypt blocks smaller than 16 bytes without adding padding that changes the file size).

10.6 Subsample Encryption

10.6.1 Definition

In subsample encryption, each sample is divided into one or more contiguous subsamples. Each subsample has an unencrypted part followed by an encrypted part, only one of which MAY be zero bytes in length, but usually both are non-zero values. The total length of all of the subsamples (`BytesOfClearData` + `BytesOfEncryptedData` for all subsamples that make up a sample) SHALL be equal to the size of the sample itself.

The encrypted regions of a sample are treated as a logically contiguous chain of 16 byte cipher blocks, even when they are separated by `BytesOfClearData`. In other words, the block counter is not incremented between subsamples, but only at the end of 16 byte blocks, or the start of the next sample by its initialization vector. Subsamples MAY be end-aligned in video slice NALs by appropriately sizing `BytesOfClearData` and `BytesOfEncryptedData` to extend to the last byte of video data.

Figure 2 shows Subsample based encryption using AES-CTR.

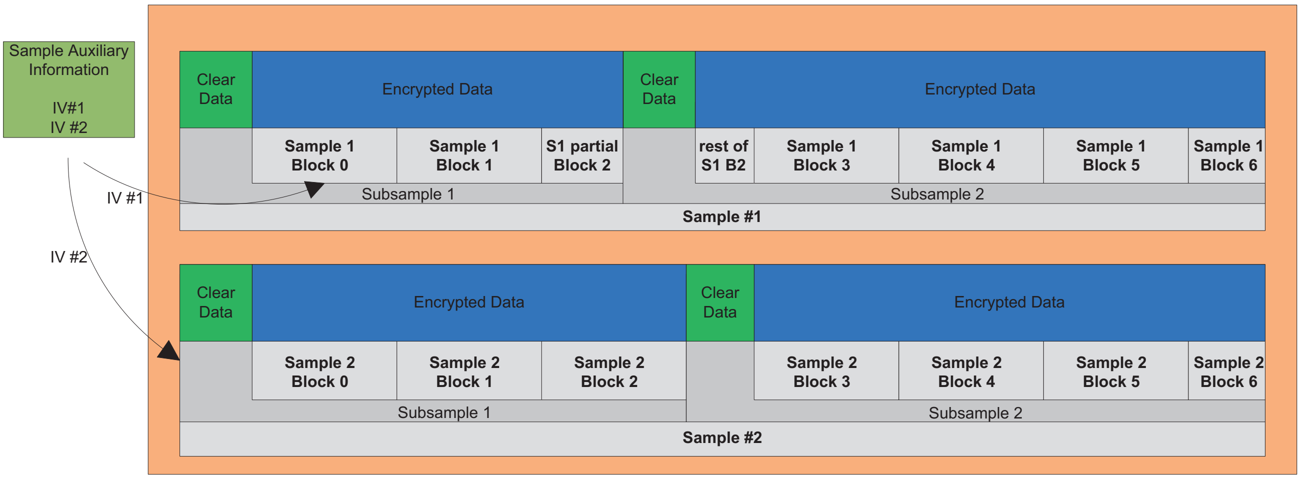

Figure 2 — Subsample-based encryption scheme for AES-CTR with IVs shown

Note that AES-CTR mode is a block cipher mode that acts like a stream cipher. Cipher blocks are shown to illustrate the underlying blocks used in generating the stream cipher. Block 6 in both Sample #1 and Sample #2 are shown as less than 16 byte blocks to illustrate that CTR mode can encrypt partial blocks without padding to 16 bytes and changing file size. Also note that Block 2 of Sample #1 is used to encrypt the end of the first subsample and the beginning of the second subsample as one logically continuous block using one counter value.

10.6.2 Encryption of NAL Structured Video Tracks

Encrypted NAL structured video tracks SHALL use subsample encryption as specified in the following Subclauses.

10.6.2.1 Structure of NAL video tracks

NAL structured video specifications defines NAL unit syntax elements that can be sequenced to form elementary streams, and access units that can be decoded to images. ISO/IEC 14496-15 specifies how NAL structured video is stored in ISO Base Media files, and how each access unit is stored as a sample in a track. Each sample is composed of multiple NAL units, and each NAL unit is separated by a Length field stating the length of the NAL unit. For example, Figure 3 shows NAL structured video samples distributed over multiple NAL units.

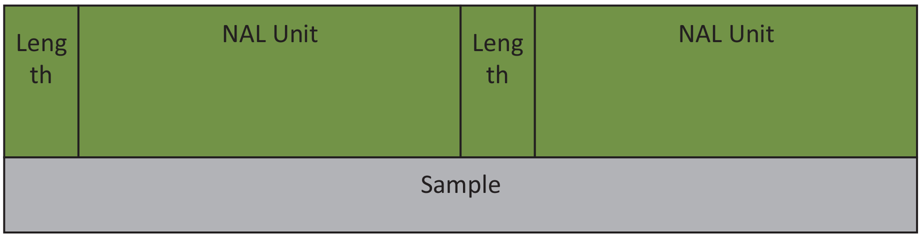

Figure 3 — NAL Structured Video sample distributed over multiple NAL units

Secure video processors typically do not make data from the video stream that has been decrypted available to applications in order to protect decrypted video, so display applications that need to access information stored in video slice headers or SEI NAL units, such as caption and framing information, will not be able to access that data if it is encrypted.

Not all decoders are designed to deal with 'avc1' formatted streams with size headers. Some decoders are designed to handle a different AVC elementary stream format: for example, ISO/IEC 14496-10, Annex B byte stream format with startcode delimited NAL Units. It may be necessary to reformat the elementary stream in order to transmit the data using a network protocol like RTP that packetizes NAL Units.

Full sample encryption prevents stream reformatting and information access prior to decrypting the samples. If NAL headers or complete NALs other than video slice data are left unencrypted, an application can convert 'avc1' bit-streams to Annex B byte streams by replacing unencrypted NAL size headers with start codes and inserting PPS/SPS NAL units to form *sequence headers*. Encryption of only the video slice data allows applications to access information in SEI NALs as well as picture information in slice NAL headers.

10.6.2.2 Subsample Encryption Applied to NAL Structured Video

For NAL structured video samples, each NAL unit SHALL be spanned by one or more subsamples. The slice data in a video NAL MAY be spanned by multiple subsamples to create multiple clear and encrypted ranges, or to span slice data that is larger than a single subsample.

— For AVC video using 'avc1' sample description stream format, the NAL length field and the `nal_unit_type` field (the first byte after the length) of each NAL unit SHALL be unencrypted, and only video data in slice NALs SHOULD be encrypted. Note that the length field is a variable length field. It can be 1, 2, or 4 bytes long and is specified in the Sample Entry for the track as the `lengthSizeMinusOne` field in the `AVCDecoderConfigurationRecord`

- For other NAL structure video stream formats, only video slice data and other data carried in SEI messages that require protection SHALL be encrypted. For avoidance of doubt, video NAL slice, size and type headers SHALL be unencrypted. Other NAL types SHALL be unencrypted, except for SEI data requiring protection, e.g. protected caption data.
- Partial video encryption MAY be implemented with multiple subsamples per video NAL that indicate multiple clear and encrypted byte ranges per video slice. There MAY be multiple subsamples per NAL, and MAY be multiple NALs per subsample, e.g. when multiple unencrypted NALs are included in one clear byte range for efficient representation.

Figure 4 illustrates Subsample Encryption Applied to NAL structured video slice NALs using AES-CTR. The figure details the IVs used, the areas of clear data, the areas of encrypted data, as well as the NAL unit and sample boundaries.

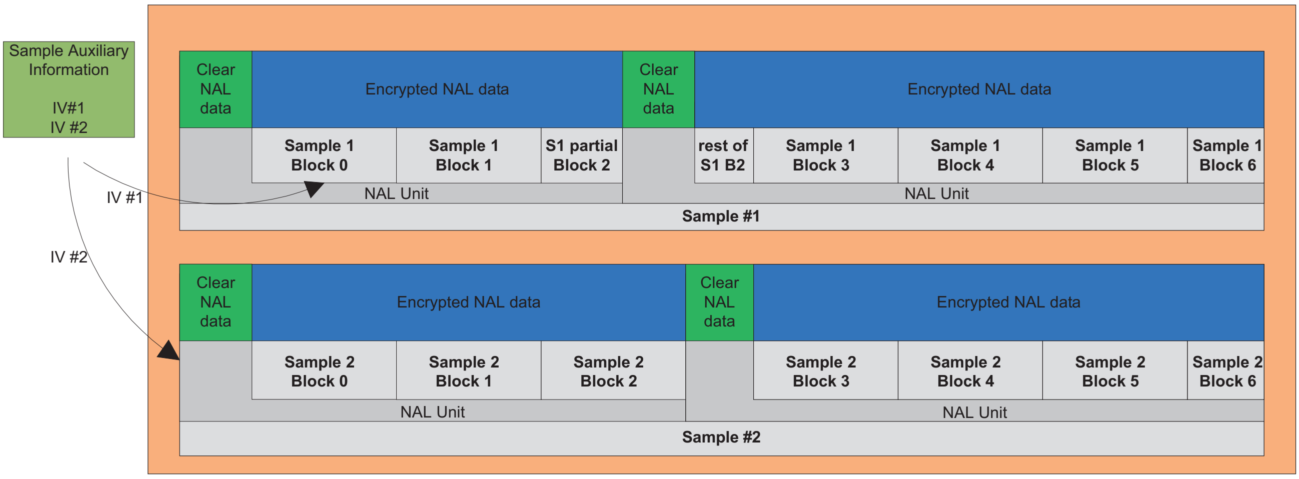

Figure 4 — Subsample Encryption Applied to NAL Structured Video using AES-CTR

Note that AES-CTR mode is a block cipher mode that acts like a stream cipher. Cipher blocks are shown to illustrate the underlying blocks used in generating the stream cipher. The last block (block 6) in both Sample #1 and Sample #2 are less than 16 bytes to illustrate that CTR mode allows encryption of partial blocks without padding to 16 bytes and changing file size. Also note that Block 2 of Sample #1 is used to encrypt the end of the first NAL unit and the beginning of the second NAL unit as a logically continuous cipher block with one counter value. This example shows subsamples that match the size of each NAL unit, but that is not a constraint of this specification.

11 AES 128-bit Cipher Block Chaining (CBC-128) Encryption of Media Data

11.1 Introduction to AES 128-bit Cipher-Block Chaining (CBC-128) Mode

Media data using ‘cbc1’ Protection Scheme uses the Advanced Encryption Standard specified by AES [FIPS197] using 128-bit keys in Cipher-block chaining mode (AES-CBC-128), as specified in Block Cipher Modes [NIST 800-38A], with IVs stored as described in sections 7 and 10.2. Encrypted NAL Structured Video Tracks SHALL follow the scheme outlined in 11.2.3, which defines a NAL unit based encryption scheme to allow access to NAL units and unencrypted NAL unit headers in an encrypted stream of NAL Structured Video. All other types of tracks must follow the scheme outlined in 10.2.4, which defines a simple sample-based encryption scheme.

NOTE Support for ‘cbc1’ scheme is not mandatory in the common encryption mechanism, however implementations that process the ‘cbc1’ scheme are also required to process the ‘cenc’ scheme so that files using the ‘cenc’ scheme MAY be processed on all implementations of this standard.

11.2 AES-CBC-128 Mode

The scheme_type field of the scheme Type Box (`schm') SHALL be set to 'cbc1' to signal AES-CBC-128 Mode. The AES-CBC-128 mode SHALL follow the same mechanisms as defined in sections 4 to 9.3 except for Initialization Vector creation, 10.5 and 10.6.2, but using the 'cbc1' rather than 'cenc', and with additional constraints as detailed in 11.2.1 to 11.2.5.

11.2.1 Field Semantics for AES-CBC-128 Mode

`IV_size` (as defined in 10.2) SHALL be 16 (which specifies 128-bit initialization vectors).

11.2.2 Creation of Initialization Vectors (Informative)

There are no constraints on the values used for initialization vectors when applying encryption. However, security may be improved if the first initialization vector used for encryption is randomly selected and no duplicate values are used with the same KID value. Decryption efficiency may be improved if subsequent initialization vectors use the value of the last cipher block at the end of the previous sample so that multiple samples may be decrypted as a continuous chain.

11.2.3 AES-CBC-128 Mode Encryption of NAL Structured Video Tracks

AES-CBC-128 encryption of NAL Structured Video Tracks follow the principles set out in 9.6.2.2 using partial encryption as signalled by the common encryption sample auxiliary information described in 7. The size of clear data (BytesOfClearData) at the beginning of each NAL Unit SHALL be set such that the size of encrypted NAL data (BytesOfEncryptedData) be an integral number of 16 bytes blocks terminating at the end of each subsample. Figure 5 below shows AES-CBC-128 handing of NAL Structured Video tracks.

NOTE There are no clear partial blocks at the end of the NAL Unit Payload as shown in Figure 5.

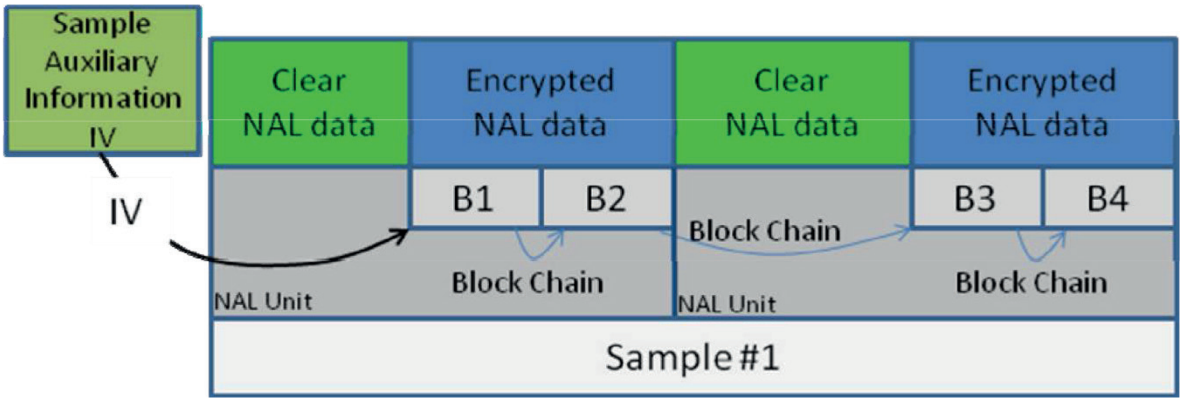

Figure 5 — Subsample Encryption Applied to NAL Structured Video using AES-CBC-128

11.2.4 Full Encryption in AES-CBC-128 Mode

For full encryption in AES-CBC-128 Mode, residual block (i.e. when the last block in the chain is less than 16 bytes) SHALL be left in the clear as shown in Figure 6 below. If a sample size is smaller than 16 bytes, then the sample SHALL be treated as a solitary block and left in the clear,

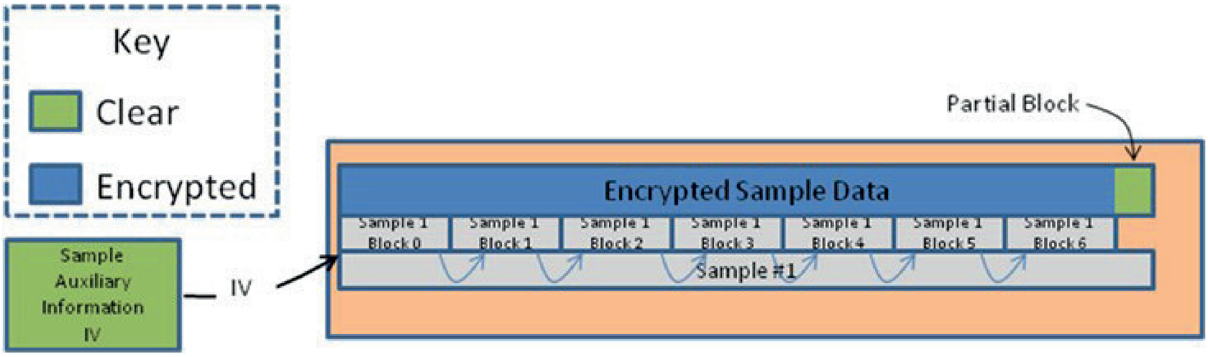

Figure 6 — Sample-based Encryption for AES-CBC-128

12 XML Representation of Common Encryption Parameters

In some cases, such as MPEG Dynamic Adaptive Streaming over HTTP (ISO/IEC 23009-1 DASH), it is useful to express the default_KID field from the Track Encryption Box (‘tenc’) and Protection System Specific Header Box (‘pssh’) in an XML manifest document accessible prior to availability of media. Then a media player application may read the XML default_KID value to determine if that key has been acquired, and may acquire a license using information in a cenc:pssh element in advance of media availability. To encourage consistency, an attribute and element to express the Common Encryption default_KID and pssh are specified in XML below. XML documents that allow extension attributes and elements SHOULD use the specified namespace, attribute, and element for consistency.

12.1 Definition of the XML cenc:default_KID attribute and cenc:pssh element

The cenc:default_KID attribute and cenc:pssh element SHALL be defined within the “urn:mpeg:cenc:2013” namespace by the following schema:

```
<xs:schema
    targetNamespace=”urn:mpeg:cenc:2013”
    attributeFormDefault=”unqualified”
    elementFormDefault=”qualified”
    xmlns:xs=”http://www.w3.org/2001/XMLSchema”
    xmlns:cenc=”urn:mpeg:cenc:2013”>

    <!-- KID is a 128-bit integer written in canonical UUID notation -->
    <xs:simpleType name=”KeyIdType”>
        <xs:restriction base=”xs:string”>
            <xs:pattern value=”[A-Fa-f0-9]{8}-[A-Fa-f0-9]{4}-[A-Fa-f0-9]{4}-[A-Fa-f0-9]
{4}-[A-Fa-f0-9]{12}”/>
        </xs:restriction>
    </xs:simpleType>

    <!-- space-delimited list of KIDs -->
    <xs:simpleType name=”KeyIdListType”>
        <xs:list itemType=”KeyIdType” />
    </xs:simpleType>

    <!-- attribute used within the DASH mp4protection descriptor -->
    <xs:attribute name=”default_KID” type=”KeyIdListType”/>

    <!-- element used within system specific UUID ContentProtection descriptors -->
    <xs:element name=”pssh” type=”xs:base64Binary”/>

</xs:schema>
```

Documents SHOULD use namespace prefix: ‶cenc:″.

`default_KID` is a string in UUID format [1]. Any 128-bit number may be written using this hyphenated hexadecimal notation (even if it is not generated as a UUID), though use of mathematically unique UUIDs throughout the system is highly recommended to prevent number collisions between independent content producers.

`cenc:pssh` is a base64 encoded `‘pssh’` box with `SystemID` matching the `SystemID` of the containing Content Protection Descriptor element.

Note Frequently used SystemID identifier values indexed to protection systems and their specifications may be found on the DASH Industry Forum web site: http://dashif.org/identifiers

12.2 Use of the cenc:default_KID attribute and cenc:pssh element in DASH ContentProtection Descriptor Elements

The MPEG DASH standard specifies Content Protection Descriptors for use in Media Presentation Descripton (MPD) XML documents [3]. The XML syntax of the DASH `ContentProtection Descriptor` element is specified in section 5.8 of DASH. The `Descriptor` complex type allows the addition of an attribute and/or element in a declared namespace different from the DASH namespace. This extension mechanism may be used to add the cenc:default_KID attribute and `cenc:pssh` element defined above to store information that is defined in this Common Encryption standard for storage in ISO Media files also in an MPD.

12.2.1 Addition of cenc:default_KID attributes in DASH ContentProtection Descriptors

The `default_KID` (the default Key Identifier field stored in the Track Encryption Box `‘tenc’`) identifies the default key used to encrypt samples in an encrypted ISO Base Media track. It may be used with the DASH specified `“mpeg:dash:mp4protection:2011”` content protection scheme to identify the protection scheme and `default_KID` used in an ISO Media file. The attribute `@value` is specified to contain the four character code of the ISO Media Scheme Type Box (`‘schm’`). If a Common Encryption scheme such as `‘cenc’` is contained in the @value attribute, the encryption scheme can be decrypted by any number of DRM key management systems that have access to the media key(s) and support that decryption scheme.

The `default_KID` field in `‘tenc’` is a big endian array of 16 bytes, and is defined above to be stored in the `cenc:default_KID` attribute in DASH `ContentProtection` Descriptor elements as a UUID string.[1]

When a `ContentProtection` descriptor refers to several tracks, and these use different default Key Identifiers in different `‘tenc’` boxes, the `cenc:default_KID` attribute SHALL store a space-delimited list of those different `default_KID` values.

The following is an example of `cenc:default_KID` contained in a `ContentProtection` Descriptor element:

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

NOTE For global uniqueness, a UUID [1] SHOULD be used for each unique `KID`/key value pair to prevent duplicate IDs for different keys by independent publishers. Publishers may use the same key value and `KID` in more than one track or file according to their rights management intentions.

With unique `KIDs`, a license request using the `cenc:default_KID` attribute value is sufficient to identify a DRM license containing the encryption key(s) used to encrypt the media; and that license can enable decryption and playback of the Components, Representations, or Adaptation Sets that the `ContentProtection` Descriptor element and `default_KID` describe.

Common Encrypted content described in an MPD SHALL include an `mp4protection` Content Protection Descriptor. A `cenc:default_KID` attribute SHOULD be contained in the `mp4protection` Content Protection Descriptor to identify the `default_KID` in the content, and need not be duplicated in each UUID Content Protection descriptor specific to each protection system.

12.2.2 Addition of the cenc:pssh element in protection system specific UUID Content Protection Descriptors

DASH ContentProtection Descriptor elements in an MPD may use a urn:uuid schemeIdUri to identify a specific DRM system using the SystemID value used in the Protection System Specific Header Box (‘pssh’) defined in this specification. Each DRM system may also specify additional elements and attributes for its scheme and SystemID that can be used for license acquisition or other functions of the identified DRM system.

In addition to containing a Content Protection Descriptor with "urn:mpeg:dash:mp4protection:2011" to notify a DASH player that the content is encrypted, it is also recommended that an MPD include a ContentProtection Descriptor with schemeIdUri of urn:uuid for each SystemID that can provide a DRM license, and include sufficient information in the descriptor to enable license acquisition. License acquisition can be enabled by adding a cenc:pssh element to each urn:uuid scheme descriptor so that a player can find the same license acquisition information it would find in a ‘pssh’ box. An MPD will normally be processed before Media Segments are downloaded, so license acquisition information in an MPD will normally take precedence over information stored in ‘pssh’ boxes. Note that the cenc:pssh element contains a complete ‘pssh’ box, not just the contents of the box, so that parsing will be identical from the MPD or file.

12.2.3 Example showing two Content Protection Descriptors included in an MPD

The first Content Protection Descriptor with schemeIdUri of urn:mpeg:dash:mp4protection:2011 indicates that the ‘cenc’ scheme (Common Encryption CTR mode) was used to encrypt the referenced media, and the cenc:default_KID.

The second Content Protection Descriptor with schemeIdUri of urn:uuid type, indicates a hypothetical DRM system "Acme" with a SystemID represented as a UUID string, and a cenc:pssh element containing a base64 encoded ‘pssh’ box with that SystemID to provide license acquisition information.

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

<ContentProtection schemeIdUri=”urn:uuid:d0ee2730-09b5-459f-8452-200e52b37567”
        value=”Acme 2.0”>

        <!-- base64 encoded ‘pssh’ box with this Acme SystemID -->
        <cenc:pssh>
           YmFzZTY0IGVuY29kZWQgY29udGVudHMgb2YgkXB
           zc2iSIGJveCB3aXRoIHRoaXMgU3lzdGVtSUQ=
        </cenc:pssh>
</ContentProtection>
```

A single mpeg:dash:mp4protection:2011 Content Protection Descriptor may be sufficient for license acquisition if license acquisition information is provided in the media (e.g. in ‘pssh’ boxes), in a player application, or by an Internet service that can resolve the cenc:default_KID value to a license. Alternatively, the publisher of a DASH MPD may provide all the license acquisition information in MPD Content Protection Descriptors so that a DASH player may immediately acquire a license for a DRM system the player supports on receipt of the MPD by using the Content Protection Descriptor containing a cenc:pssh element for that DRM system.

Bibliography

[1] ITU-T Rec. X.667 (09/2004) | ISO/IEC 9834-8:2005, Information technology — Open Systems Interconnection — Procedures for the operation of OSI Registration Authorities: Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 Object Identifier components

[2] ITU-T Rec.H.264 | ISO/IEC 14496-10, *Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding*

[3] *Advanced Encryption Standard*, Federal Information Processing Standards Publication 197, FIPS-197, http://www.nist.gov/

[4] *Recommendation of Block Cipher Modes of Operation*, NIST, NIST Special Publication 800-38A, http://www.nist.gov/

[5] IETF RFC 3406, *Uniform Resource Names (URN) Namespace Definition Mechanisms*, October 2002

[6] IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*, January 2005

[7] IETF RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace,* July 2005

[8] ISO/IEC 23009-1, *Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats*

[9] ISO/IEC 23008-2, *Information technology — D Coding of audio-visual objects — D Part 2: High Efficiency Video Coding (HEVC)*

ICS 35.040

Price based on 17 pages