ISO/IEC JTC 1/SC 29 N

Date: 2011-07-225

ISO/IEC CD 23001-7 2nd Edition

ISO/IEC JTC 1/SC 29/WG 11

Secretariat:

Information technolog — MPEG systems technologies — Part 7: Common encryption in ISO base media file format files

Élément introductif — Élément central — Partie 7: Titre de la partie

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: International Standard
Document subtype:
Document stage: (30) Committee
Document language: E

STD Version 2.1c2

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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.

ISO/IEC 23001-7 was prepared by Joint Technical Committee 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 (ISO/IEC 23001-7:2012), which has been technically revised.

ISO/IEC 23001 consists of the following parts, under the general title *Information technolog — 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*

Introduction

The Common Encryption ('cenc') protection scheme specifies standard encryption and key mapping methods that can be utilized by one or more digital rights and key management systems [digital-rights management (DRM systems)] to enable decryption of the same file using different DRM systems. The scheme 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 'cenc' 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'), using one for each DRM system applied. 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.

Information technolog — 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, The ISO Base Media File Format.

2 Normative references

The following referenced documents are indispensable for the application of this document. 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 NAL unit structured video in the ISO Base Media File Format*

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

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

3 Definitions

3.1 Terms and definitions

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

3.1.1
ISO Base Media File
name of a file conforming to the file format described in ISO/IEC 14496-12 in which the techniques in ISO/IEC 23001-7 may be used

NOTE Adapted from ISO/IEC 14496-12, definition 3.1.8.

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

NAL Unit AES Counter Mode as specified in *Recommendation of Block Cipher Modes of Operation*, NIST, NIST Special Publication 800-38A

NAL Structured Video A video sample description format specified by ISO/IEC 14496-15 – Carriage of NAL unit structured video in the ISO Base Media File format

4 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 has 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 shall be 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 has the following additional constraint:

- The encryption metadata defined by schemes conforming to this standard can be categorized as follows:

5 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 8.1.
- 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 0.
- 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 6 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 7 for further details.

6 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 0), or (b) specifying the parameters by sample group, or (c) using a combination of these two techniques.

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 VisualSampleGroupDescriptionEntry( '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 AudioSampleGroupDescriptionEntry( 'seig' )
{
   unsigned int(24)     IsEncrypted;
   unsigned int(8)      IV_size;
   unsigned int(8)[16]  KID;
}
```

NOTE Groups 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 9.2 for further details.
`IV_size` is the Initialization Vector size in bytes for samples in the sample group. See the `IV_size` field in 9.2 for further details.
`KID` is the key identifier used for samples in the sample group. See the `KID` field in 9.2 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.

7 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 9.6.2, 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 9.2 for further details.

`subsample_count` is the count of subsamples for this sample. See the `subsample_count` field in 9.2 for further details.

`BytesOfClearData` is the number of bytes of clear data in this subsample. See the `BytesofClearData` field in 9.2 for further details.

`BytesOfEncryptedData` is the number of bytes of encrypted data in this subsample. See the `BytesofEncryptedData` field in 9.2 for further details.

If sub-sample encryption is not used (sample_info_size equals IV_size), then the entire sample is encrypted (see 9.5 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` used.

8 Box Definitions

8.1 Protection System Specific Header Box

8.1.1 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.

NOTE Multiple Protection System Specific Header boxes may be associated with a given KID and `SystemID`.For storage efficiency, Protection System Specific Header boxes containing the same KID(s) and SystemID should not be duplicated in a movie fragment or movie box resulting from defragmentation or refragmentation.

8.1.2 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;
}
```

8.1.1.1 8.1.3 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.

`DataSize` specifies the size in bytes of the Data member.

`Data` holds the content protection system specific data.

8.2 Track Encryption Box

8.2.1 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.

8.2.2 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;
}
```

8.2.3 Semantics

default_IsEncypted is the encryption flag which indicates the default encryption state of the samples in the track. See the IsEncrypted field in 9.2 for further details.
default_IV_size is the default Initialization Vector size in bytes. See the IV_size field in 9.2 for further details.
default_KID is the default key identifier used for samples in this track. See the KID field in 9.2 for further details.

9 Encryption of Media Data

9.1 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 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 9.6.2 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 9.5, which defines a simple sample-based encryption scheme.

9.2 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. This allows the identification of multiple encryption keys per file or track. Unencrypted samples in an

encrypted track shall be identified by having an `IsEncrypted` flag of 0x0, an `IV_size` of 0x0, and a `KID` value of 0x0.

`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 have 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 InitializationVector 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 the `BytesOfClearData` field and in the `BytesOfEncryptedData` field unless the sample is zero bytes in length. The total length of all

`BytesOfClearData` 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}

9.3 Initialization Vectors

The initialization vector (IV) values for each sample are located in the Sample Auxiliary Information associated with the encrypted samples. See 9.2 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 9.4, it is recommended that the initialization vector is treated as a 128-bit number when calculating the next initialization vector from the previous one.

9.4 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 9.2. 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).

9.5 Full Sample Encryption

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

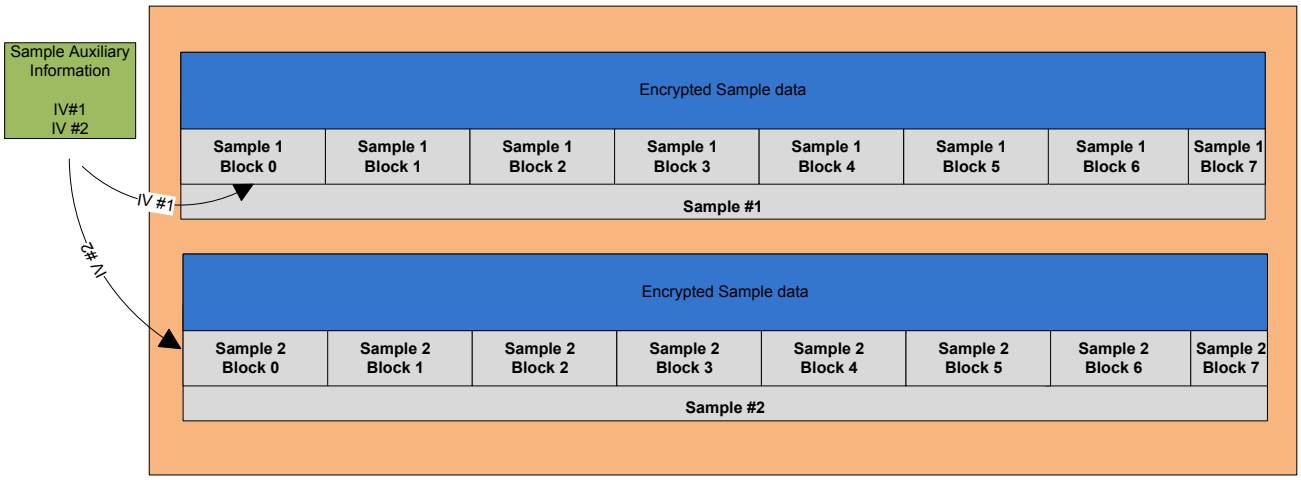

Figure 1 — Sample-based encryption for AES-CTR

Note that AES-CTR mode is a block cipher mode that acts like a stream cipher. Blocks are shown to illustrate the underlying cipher blocks used in generating the stream cipher (this is why Block 7 is shown as only partially used, as the unused bytes of the stream cipher are discarded during the encryption process).

9.6 Subsample Encryption

9.6.1 Definition

In subsample encryption, the sample is divided into one or more subsamples. Each subsample may have an unencrypted part followed by an encrypted part. The total length of all of the subsamples (BytesOfClearData + BytesOfEncryptedData for each subsample) 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 block, even though they are broken up by areas of clear data. In other words, the block counter is not arbitrarily incremented between NAL units. Figure 2 shows Subsample based encryption using AES-CTR.

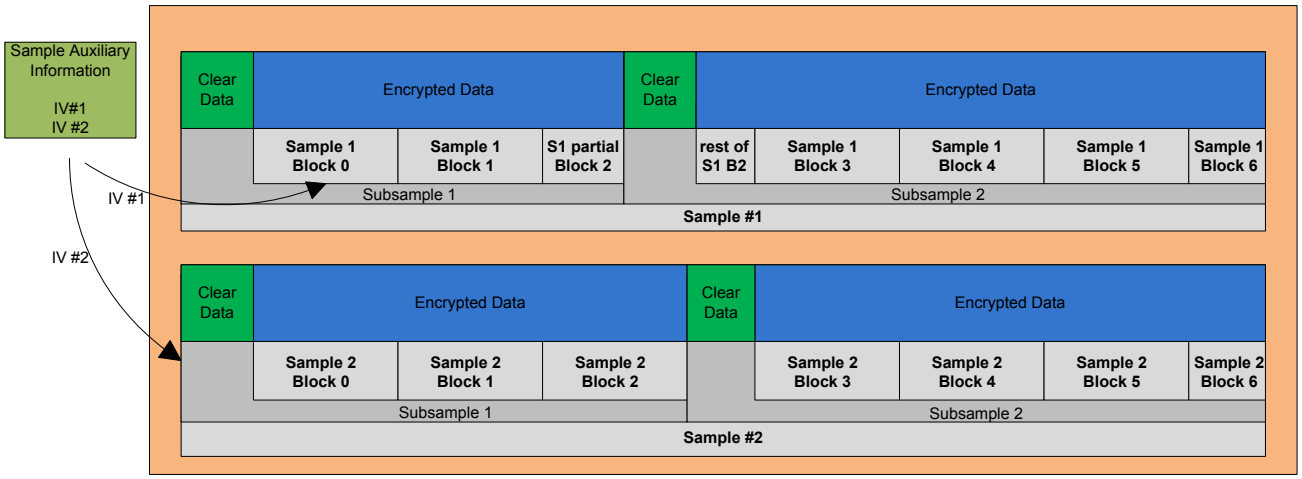

Figure 2 — Subsample-based encryption scheme for AES-CTR with IVs shown

Note that AES-CTR mode is a block cipher mode that acts like a stream cipher. Blocks are shown to illustrate the underlying blocks used in generating the stream cipher. This is why Block 6 in both Sample #1 and Sample #2 are not shown as full 16 byte blocks, the unused bytes of the stream cipher are discarded during the encryption process. 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.

9.6.2 Encryption of NAL Structured Video Tracks

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

9.6.2.1 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 structure 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 for 'avc1' and 'avc2' sample descriptions 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.

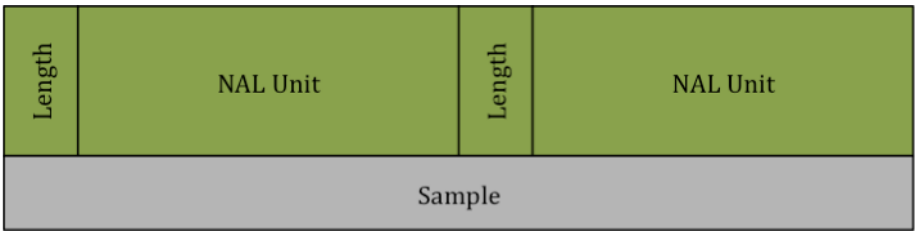

Figure 3 — NALStructured Video sample distributed over multiple NAL units

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 with startcode delimited NAL Units. Further, 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 without first decrypting the samples to access NAL Units or their headers.

In this case, 'avc1' type bit-stream can be converted to Annex B byte stream format by adding start codes and PPS/SPS NAL units as *sequence headers*. To facilitate stream reformatting before decryption, it is required that at least the NAL length field and the `nal_unit_type` field (the first byte after the length) of each NAL

unit is left unencrypted, and recommended that entire slice NAL headers be unencrypted. In addition, it should be noted 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`
- There are multiple NAL units per sample, requiring multiple pieces of clear and encrypted data per sample.
- NAL units that do not contain video slice data need not be encrypted, and should not be encrypted if they contain information that must be accessed prior to decryption, such as caption information contained in SEI NAL units.

9.6.2.2 Subsample Encryption Applied to NAL Structured Video

For NAL structured video samples, each NAL unit shall be treated as a subsample. Further, the BytesOfClearData value for each subsample shall be large enough that slice NAL headers remain unencrypted. NAL units that do not contain slice data may remain unencrypted.

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.

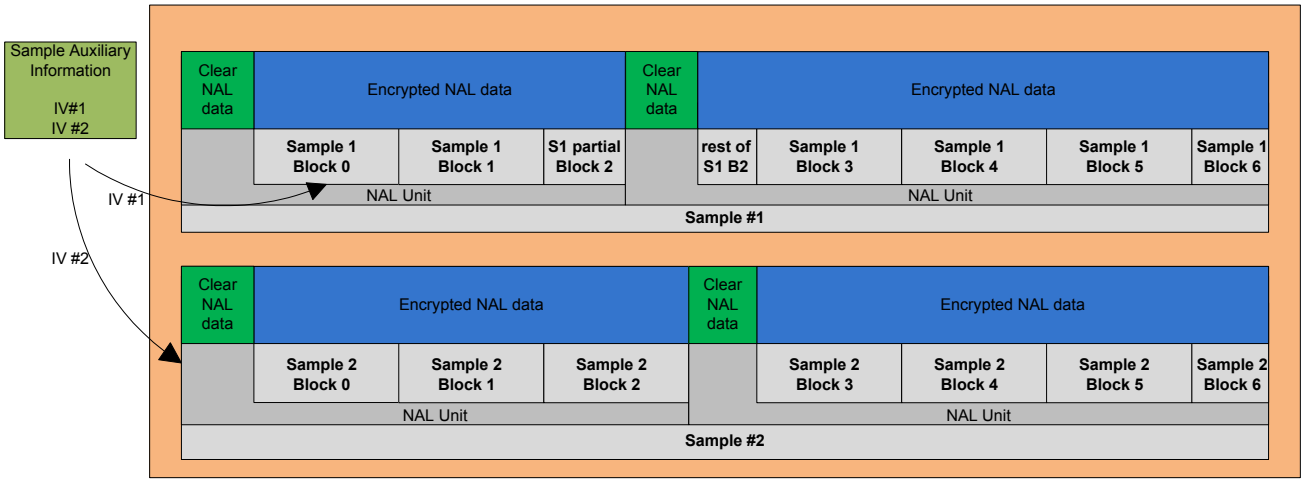

Figure 4 — Subsample Encryption Applied to NAL Structured Video using AES-CTR

Note that AES-CTR mode is a block cipher mode that acts like a stream cipher. Blocks are shown to illustrate the underlying blocks used in generating the stream cipher. This is why Block 6 in both Sample #1 and Sample #2 are not shown as full 16 byte blocks, the unused bytes of the stream cipher are discarded during the encryption process. Also note that Block 2 of Sample #1 is used to encrypt the end of the first NAL and the beginning of the second NAL.

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

10.1 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 6 and 9.2. Encrypted NAL Structured Video Tracks shall follow the scheme outlined in10.2.4, 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 10.2.5, 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.

10.2 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 Clauses 4 to 9.3 except for Initialization Vector creation, 9.5 and 9.6.2, but using the 'cbc1' rather than 'cenc', and with additional constraints as detailed in 10.2.1 to 10.2.5.

10.2.1 Field Semantics for AES-CBC-128 Mode

`IV_size` (as defined in 9.2) shall be 16 which specifies 128-bit initialization vectors).

10.2.2 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.

10.2.3 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 9.6.1 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.

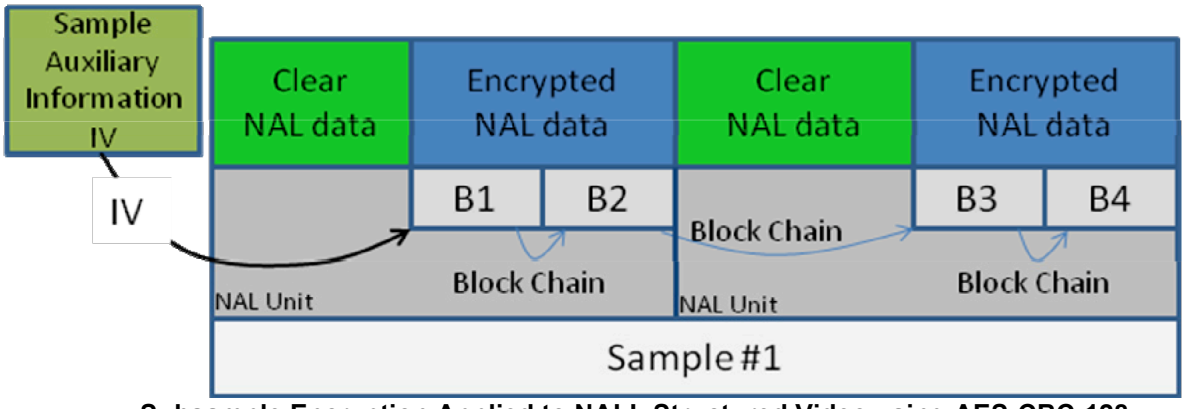

Figure 5 — Subsample Encryption Applied to NALL Structured Video using AES-CBC-128

10.2.4 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 chaining is less than 16 bytes) shall be left in the clear as shown in figures 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,

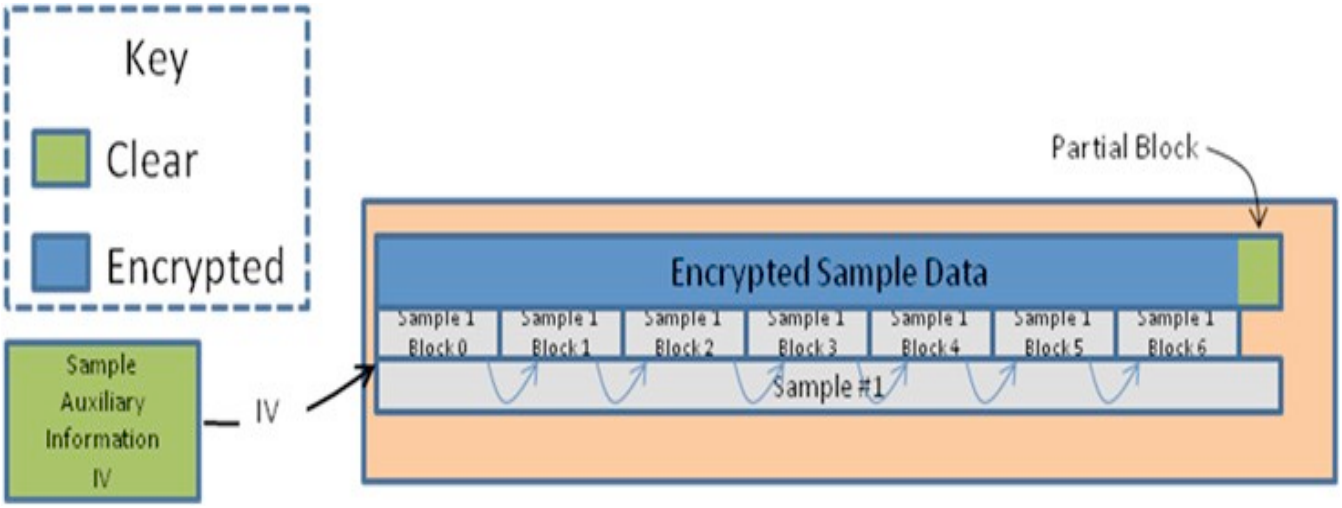

Figure 6 — Sample-based Encryption for AES-CBC-128

11 XML Representation of Common Encryption Parameters

In some cases, such as MPEG DASH streaming [3], it is useful to express the default_KID field in the Track Encryption Box ('tenc') of an ISO Media track in an XML document accessible prior to availability of media. Then a media player application may read the XML KID value to acquire a license with that key in advance. To encourage consistency, an attribute to express the Common Encryption default_KID value in XML is specified below. XML documents that allow extension attributes should use the specified namespace and attribute for consistency.

11.1 Definition of the default_KID XML attribute

The cenc:default_KID attribute shall be declared in the following namespace:

```
xmlns:cenc="urn:mpeg:cenc:2013"
```

and use the following attribute schema syntax:

```
<xs:attribute name="default_KID" type="xs:string"/>
```

resulting in an attribute instance of the form:

```
cenc:default_KID=”string”
```

Where “string” should be a UUID formatted according to RFC-4122 section 3 that is equal to a UUID byte array stored in the default_KID field of the 'tenc' box.

11.2 Use of the cenc:default_KID attribute in DASH MPD ContentProtection Descriptor Elements

The syntax of the DASH ContentProtection Descriptor element is specified in DASH [3] by the schema excerpts below. The Descriptor complex type allows the addition of an attribute in a declared namespace different from the DASH namespace. This extension mechanism may be used to add the cenc:default_KID attribute defined above.

```
<!– DASH ContentProtection Descriptor -->
<xs:sequence>
    <xs:element name="ContentProtection" type="DescriptorType" minOccurs="0"maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>

<!– DASH Descriptor Complex Type -->
<xs:complexType name="DescriptorType">
   <xs:sequence>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
    <xs:attribute name="schemeIdUri" type="xs:anyURI" use="required"/>
    <xs:attribute name="value" type="xs:string"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

11.2.1 Application to ContentProtection Descriptor with urn:mpeg:dash:mp4protection:2011

The default_KID (default Key Identifier in ‘tenc’) is a property of an encrypted ISO Base Media track. It may be added to the DASH specified “mp4protection” content protection scheme identifier for ISO Media files using ‘cenc’ encryption scheme and accessed by any number of DRM key management systems.

The default_KID parameter in an ISO Media Track Encryption Box (‘tenc’) is a big endian array of 16 bytes, so a UUID is stored as a byte array as specified by RFC-4122 section 4.1.2. It shall be represented in the cenc:default_KID attribute in DASH ContentProtection Descriptor elements in the UUID string format specified in RFC-4122 section 3 (also big endian).

For example:

```
<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 name collisions between 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:defaultKID attribute value is sufficient to identify a DRM license containing that key that will enable playback of the Components, Representations, Adaptation Sets, or Periods that the ContentProtection Descriptor element and default_KID describe.

11.2.2 Application to DRM specific ContentProtection Descriptors

DASH ContentProtection Descriptor elements may use a schemeIdUri identifying a DRM system by using the SystemID parameter as specified in the Protection System Specific Header Box (‘pssh’) defined in this specification. The cenc:default_KID attribute may also be used in these SystemID-specific ContentProtection Descriptors, as specified by the DRM system identified by the SystemID value. Each DRM system may also specify additional elements and attributes that may be used for license acquisition or other functions of the identified DRM system.

For Example: A DASH MPD may include the following Content Protection Descriptors to indicate that the 'cenc' scheme (Common Encryption) was used to encrypt the referenced media, and provide license acquisition information for one (or more) DRM system(s) with the indicated SystemID. This particular DRM Content Protection Descriptor uses the SystemID UUID format defined in DASH section 5.8.5.2 [3], and defines an extension element in its namespace (mspr:) that contains the contents of a 'pssh' box with that SystemID. It also uses the cenc:default_KID attribute defined in this specification:

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

<ContentProtection schemeIdUri="urn:uuid:9a04f079-9840-4286-ab92-
e65be0885f95" value="2.0" cenc:default_KID="34e5db32-8625-47cd-ba06-
68fca0655a72"/>
     <mspr:prheader>
     <!—base64 encoded contents of 'pssh' box with this SystemID>
     </mspr:prheader>
</ContentProtection>
```

Note that a single 'cenc' Content Protection Descriptor may be sufficient if license acquisition information is provided in the media (e.g. a 'pssh' box), in a player application, or by an Internet service that can resolve the KID to a license. However, the publisher of a DASH MPD may wish to provide all the license acquisition information in the MPD so DASH players may immediately acquire a license on receipt of the MPD for any DRM system they support by reading Content Protection Descriptors for each DRM system that can provide a license.

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] ISO/IEC 14496-10, Information technology — *Coding of audio-visual objects — Part 10: Advanced Video Coding*

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

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