FINAL
DRAFT

INTERNATIONAL
STANDARD

**ISO/IEC
FDIS
23001-7**

ISO/IEC JTC **1**/SC **29**

Secretariat: **JISC**

Voting begins on:
2023-05-10

Voting terminates on:
2023-07-05

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 FDIS 23001-7:2023(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: 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.

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 or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This fourth edition cancels and replaces the third edition (ISO/IEC 23001-7:2016), which has been technically revised. It also incorporates the Amendment ISO/IEC 23001-7:2016/Amd 1:2019.

The main changes are as follows:

Addition of:

— item encryption, which allows image items to use protection schemes defined for media tracks,

— support for multiple keys and IVs per protected sample,

— `'sve1'` sensitive encryption scheme, a codec-specific encryption scheme for which the encrypted bitstream remains a valid decodable bitstream,

— improved selective encryption using sample groups

A list of all parts in the ISO/IEC 23001 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

Common Encryption specifies encryption and key mapping methods that enable decryption of the same file using different Digital Rights Management (DRM) and key management systems. It defines encryption algorithms and encryption related metadata necessary to decrypt the protected streams, yet it leaves the details of rights mappings, key acquisition and storage, DRM content protection compliance rules, etc., up to the DRM system or systems. For instance, DRM systems necessarily support identifying the decryption key via stored key identifiers (KIDs), but how each DRM system protects and locates the KID identified decryption key is left to a DRM-specific method.

DRM specific information such as licenses, rights, and license acquisition information can be stored in an ISO Base Media file using a `ProtectionSystemSpecificHeaderBox`. Each instance of this box stored in the file corresponds to one applicable DRM system identified by a well-known `SystemID`. DRM licenses or license acquisition information need not be stored in the file in order to look up a separately delivered key using a `KID` stored in the file and decrypt media samples using the encryption parameters stored in each track.

The second edition of this document added XML representations of Common Encryption parameters for delivery in XML documents, such as an MPEG DASH Media Presentation Description Documents (MPD). The second edition also defined the `'cbc1'` protection scheme using AES-CBC mode encryption.

The third edition added `'cbcs'` and `'cens'` protection schemes for pattern encryption, which encrypt only a fraction of the data blocks within each video subsample protected. Pattern encryption reduces the computational power required by devices to decrypt video tracks.

The additions in this fourth edition are listed in the Foreword.

Information technology — MPEG systems technologies —

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

1 Scope

This document specifies common encryption formats for use in any file format based on ISO/IEC 14496-12. File, item, track, and track fragment metadata is specified to enable multiple digital rights and key management systems (DRMs) to access the same common encrypted file or stream. This document does not define a DRM system.

The AES-128 symmetric block cipher is used to encrypt elementary stream data contained in media samples. Both AES counter mode (CTR) and Cipher Block Chaining (CBC) are specified in separate protection schemes. Partial encryption using a pattern of encrypted and clear blocks is also specified in separate protection schemes. The identification of encryption keys, initialization vector storage and processing is specified for each scheme.

Subsample encryption is specified for NAL structured video, such as AVC and HEVC, to enable normal processing and editing of video elementary streams prior to decryption.

An XML representation is specified for important common encryption information so that it can be included in XML files as standard elements and attributes to enable interoperable license and key management prior to media file download.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

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

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 the ISO base media file format*

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

ISO/IEC 23008-12, *Information technology — High efficiency coding and media delivery in heterogeneous — Part 12: Image File Format (HEIF)*

IETF RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace*

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

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

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

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

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1.1
block
16-byte extent of sample data that may be encrypted or decrypted by AES-128 block cipher

Note 1 to entry: This is commonly known as a cipher block.

3.1.2
CENC SAI
sample auxiliary information associated with a sample and containing cryptographic information such as initialization vector or subsample information

Note 1 to entry: The sample auxiliary information is defined in ISO/IEC 14496-12, and is not part of the sample data.

3.1.3
constant IV
initialization vector specified in a sample entry or sample group description that applies to all samples and subsamples under that sample entry or mapped to that sample group

3.1.4
initialization vector
8 or 16-byte value used in combination with a key and a block to create the first cipher block in a chain, and derive subsequent cipher blocks in a cipher block chain

3.1.5
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

3.1.6
NAL structured video
video streams composed of NAL Units

Note 1 to entry: The carriage of NAL Units is specified in ISO/IEC 14496-15

3.1.7
protection scheme
encryption algorithm and information identified by the `scheme_type` in a `SchemeTypeBox` in a `ProtectionSchemeInfoBox`

3.1.8
sample
media sample when the protection applies to media tracks, or the payload of an item when the protection applies to items

Note 1 to entry: Media sample as defined in ISO/IEC 14496-12.

Note 2 to entry: Payload of an item as defined in ISO/IEC 14496-12.

3.1.9
selective encryption
change in the `isProtected` value of samples associated with the same sample description entry

Note 1 to entry: This is achieved using `CencSampleEncryptionInformationGroupEntry` sample groups.

3.1.10
subsample
byte range within a sample consisting of an unprotected part immediately followed by a protected part

3.2 Abbreviated terms

AES	Advanced Encryption Standard
AES-CTR	AES Counter
AES-CBC	AES Cipher-Block Chaining
AVC	Advanced Video Coding as specified in ISO/IEC 14496-10
CENC	Common ENCryption
DRM	Digital Rights Management
HEVC	High Efficiency Video Coding as specified in ISO/IEC 23008-2
IV	Initialization vector
NAL	Network Abstraction Layer, as specified in ISO/IEC 14496-10 and ISO/IEC 23008-2
UUID	Universally Unique Identifier

4 Protection schemes

4.1 Scheme type signalling

Scheme signalling shall conform to ISO/IEC 14496-12. For media tracks, as defined in ISO/IEC 14496-12, the sample entry is transformed and a `ProtectionSchemeInfoBox` is added to the standard sample entry in the `SampleDescriptionBox` to denote that a stream is protected. The `ProtectionSchemeInfoBox` shall contain a `SchemeTypeBox` so that the scheme is identifiable. The `SchemeTypeBox` shall obey the following additional constraints:

— The `scheme_type` field shall be set to a value equal to a four-character code defined in Clause 10.

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

The `ProtectionSchemeInfoBox` shall also contain a `SchemeInformationBox`. For media tracks, the `SchemeInformationBox` shall contain a `TrackEncryptionBox`, describing the default encryption parameters for the track.

The schemes identify general classes of algorithms used to encrypt data. Implementations should not rely solely on `scheme_type` and `scheme_version` to determine if they can process a file and should also take into account:

— parameters associated with the scheme (e.g. the pattern in case of pattern encryption, or the size of initialization vectors),

— use of `CencSampleEncryptionInformationGroupEntry` and the associated parameters (e.g. change in `isProtected`, change in number and/or values of keys, change in size of initialization vectors),

— value of the field `aux_info_type_parameter` associated with CENC SAI,

— versions and flags of the `SampleEncryptionBox` box if present,

— versions of the `ProtectionSystemSpecificHeaderBox` and `TrackEncryptionBox`,

— support for, and values of versions and flags, of `ItemEncryptionBox` and `ItemAuxiliaryInformationBox`.

This document does not define brands nor profiles to restrict or recommend combinations of these parameters. Derived specifications may restrict some of these aspects.

4.2 Common encryption scheme types

Five protection schemes are specified in this edition of Common Encryption. Each scheme uses syntax and algorithms specified in Clause 5 to Clause 9, as constrained in Clause 10. They are the following:

a) '**cenc**' – AES-CTR mode full sample and video NAL subsample encryption; see 10.1.

b) '**cbc1**' – AES-CBC mode full sample and video NAL subsample encryption; see 10.2.

c) '**cens**' – AES-CTR mode partial video NAL pattern encryption; see 10.3.

d) '**cbcs**' – AES-CBC mode partial video NAL pattern encryption; see 10.4.

e) '**sve1**' – AES-CTR content sensitive encryption, as defined in Annex A.

5 Overview of encryption metadata

The encryption metadata defined by Common Encryption can be categorized as follows:

— Protection system specific data – this data is opaque to Common Encryption. This gives protection systems (i.e. key and DRM 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 and vector size, protection pattern, and protection flag. This data is contained in the `TrackEncryptionBox` described in 8.2 or in the `ItemEncryptionBox` described in 8.3.

— Common encryption information for groups of media samples – this includes overrides to the track level defaults defined above. This allows groups of samples within the track to use different keys, a mix of clear and protected content, share a constant IV (for some schemes), etc. This data is contained in a `SampleGroupDescriptionBox` that is referenced by a `SampleToGroupBox`. See Clause 6 for further details.

— CENC SAI, containing cryptographic information for individual media samples such as initialization vectors and subsample encryption data. CENC SAI data is sample auxiliary information as defined in ISO/IEC 14496-12. CENC SAI may reference bytes in a `SampleEncryptionBox`. See Clause 7 for further details.

6 Encryption parameters shared by groups of samples

Each sample in a protected track shall be associated with an `isProtected` flag, optional subsample information and, for each key involved in the sample protection, a `Per_Sample_IV_Size`, `KID`, and an optional `constant_IV`. This can be accomplished by using the default values in the `TrackEncryptionBox` (see 8.2), and optionally by specifying parameters by sample group. Encryption parameters specified in a sample group override the corresponding default parameter values for the samples in that group defined in the `TrackEncryptionBox`. Samples not mapped to any sample group use the default parameters established in the `TrackEncryptionBox`.

When specifying the parameters by sample group, samples are mapped using the `SampleToGroupBox` to sample group descriptions in the `SampleGroupDescriptionBox` of type `CencSampleEncryptionInformationGroupEntry` as defined below.

The syntax of `CencSampleEncryptionInformationGroupEntry` is the same for all track types (i.e., is independent from the handler type of the track).

For fragmented files, it may be necessary to store both the mappings and descriptions in each track fragment to make them accessible for decryption of the samples they describe, e.g. when movie fragments are separately stored and delivered.

```
aligned(8) class CencSampleEncryptionInformationGroupEntry
   extends SampleGroupEntry( 'seig' )
{
   unsigned int(1)      multi_key_flag;
   unsigned int(7)      reserved = 0;
   unsigned int(4)      crypt_byte_block;
   unsigned int(4)      skip_byte_block;
   unsigned int(8)      isProtected;
   if (multi_key_flag == 1) {
      unsigned int(16)      key_count;
   } else {
      key_count = 1;
   }
   for (i=1; I <= key_count; i++) {
      unsigned int(8)      Per_Sample_IV_Size;
      unsigned int(8)[16]   KID;
      if (Per_Sample_IV_Size == 0) {
         unsigned int(8)   constant_IV_size;
         unsigned int(8)[constant_IV_size]   constant_IV;
      }
   }
}
```


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

`multi_key_flag` indicates that the multiple keys version of the sample group description is used. If this flag is set, multiple keys will be described for this sample group description entry; otherwise, a single key is described for this sample group description entry.

`isProtected` is the flag which indicates the encryption state of the samples in the sample group. See the `isProtected` field in subclause 9.1 for further details.

`key_count` indicates the number of keys that may apply to a sample associated to this sample group description entry. It is not required for a sample associated with this sample group description entry to use all the keys described.

`Per_Sample_IV_Size` is the initialization vector size in bytes for samples in the sample group. See the `Per_Sample_IV_Size` field in subclause 9.1 for further details.

`KID` is the key identifier used for samples in the sample group. See the `KID` field in subclause 9.1 for further details.

`constant_IV_size` is the size of a possible initialization vector used for all samples associated with this group (when per-sample initialization vectors are not used).

`constant_IV`, if present, is the initialization vector used for all samples associated with this group. See the `constant_IV` field in subclause 9.1 for further details.

`crypt_byte_block` specifies the count of the encrypted blocks in the protection pattern, where each block is of size 16-bytes. See subclause 9.1 for further details.

`skip_byte_block` specifies the count of the unencrypted blocks in the protection pattern. See subclause 9.1 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

7.1 Definition

Each protected sample in a protected track shall have initialization vector information associated with it. Both initialization vector and subsample encryption information may be given in a CENC SAI referenced by `SampleAuxiliaryInformationSizesBox` and `SampleAuxiliaryInformationOffsetBox`, as defined in ISO/IEC 14496-12, with `aux_info_type` equal to the scheme and `aux_info_type_parameter` equal to 0 or 1.

For example, for tracks protected using the `'cenc'` scheme, the default value for `aux_info_type` is `'cenc'` and the default value for the `aux_info_type_parameter` is 0, so content should be created omitting these optional fields.

The format of the CENC SAI for `aux_info_type_parameter` equal to 0 or 1 shall be:

```
aligned(8) class CencSampleAuxiliaryDataFormat
{
   if (aux_info_type_parameter==0) {
      unsigned int(Per_Sample_IV_Size*8) InitializationVector;
      if (sample_info_size > Per_Sample_IV_Size ) {
         unsigned int(16) subsample_count;
         {
            unsigned int(16) BytesOfClearData;
            unsigned int(32) BytesOfProtectedData;
         } [subsample_count ]
      }
   } else if (aux_info_type_parameter == 1) {
      unsigned int(16) multi_IV_count;
      for (i=1; i <= multi _IV_count; i++) {
         unsigned int(16) multi_subindex_IV;
         unsigned int(Per_Sample_IV_Size*8) IV;
      }
      unsigned int(32) subsample_count;
      {
         unsigned int(16) multi_subindex;
         unsigned int(16) BytesOfClearData;
         unsigned int(32) BytesOfProtectedData;
      } [subsample_count]
   }
}
```


Where:

`sample_info_size` is the size of the CENC SAI for this sample.

`InitializationVector` is the initialization vector for the sample, unless a `constant_IV` is present in the `TrackEncryptionBox`. See the `InitializationVector` field in 9.1 for further details.

`subsample_count` is the count of subsamples for this sample. See the `subsample_count` field in 9.1 for further details.

`BytesOfClearData` is the number of bytes of clear data in this subsample. See the `BytesOfClearData` field in 9.1 for further details.

`BytesOfProtectedData` is the number of bytes of protected data in this subsample. See the `BytesOfProtectedData` field in 9.1 for further details.

`multi_IV_count` indicates the number of entries in the initialization vector loop; this value may be zero when constant initialization vectors are used.

`multi_subindex_IV` indicates the index of the associated key entry, where value one is the first entry, in the associated list; if this data is read for the processing of a media sample, the associated list is the `'seig'` sample group description entry associated with this sample; otherwise (this data is read for the processing of an item), the associated list is the list of key definitions in the `'ienc'` item property of this item. The associated key entry shall have a `Per_Sample_IV_Size` different from 0, i.e. key entries using constant IV shall not be present in this loop. if this data is read for the processing of a media sample (i.e. not an item) and `aux_info_type_parameter` is set to 1, the associated `'seig'` sample group description entry shall have the `multi_key_flag` set to 1; Within a CENC SAI, there shall not be two `multi_subindex_IV` with the same value.

`IV` indicates the initialization vector to be used for the first block of protected data for the associated key entry.

`multi_subindex` indicates the index of the associated key entry, where value one is the first entry, in the associated list (see `multi_subindex_IV`) for the following run of encrypted data.

If subsample encryption is not used (the size of the CENC SAI equals `Per_Sample_IV_Size`), then the entire sample is protected (see 9.4 for further details). In this case, for a media track, all CENC SAI will have the same size and hence the `default_sample_info_size` of the `SampleAuxiliaryInformationSizesBox` will be equal to the `Per_Sample_IV_Size` of the initialization vector. If `Per_Sample_IV_Size` is also zero (because constant IVs are in use) then the CENC SAI is then empty and should be omitted.

NOTE Even if subsample encryption is used, the size of the CENC SAI can 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` can then be used.

7.2 Sample encryption information box for storage of sample auxiliary information

7.2.1 Sample encryption box — Definition

Box Type: `'senc'`

Container: `TrackFragmentBox` or `TrackBox`

Mandatory: No

Quantity: Zero or one

The `SampleEncryptionBox` provides an optional storage location for CENC SAI of samples in a track or track fragment.

The `SampleEncryptionBox` may be used when samples in a track or track fragment are protected. Storage of `SampleEncryptionBox` in a `TrackFragmentBox` makes the necessary CENC SAI 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.

When version 0 of `SampleEncryptionBox` is used, `sample_count` shall be equal to the number of samples in the track or track fragment. Consequently, version 0 shall not be used when selective encryption is in use.

When version other than 0 of `SampleEncryptionBox` is used, the `SampleEncryptionBox` only contains CENC SAI for samples having their `isProtected` flag different from 0x00, either through default or through an explicit `CencSampleEncryptionInformationGroupEntry` sample to group mapping. The CENC SAI entries are listed in the same order as samples in the track or track fragment. For example, the first entry will describe the CENC SAI of the first protected sample in the track or track fragment, regardless of the number of unprotected samples before this protected sample. Consequently, for version other than 0 of `SampleEncryptionBox`, there is no CENC SAI for a sample with `isProtected` different from 0x00, and the corresponding `SampleAuxiliaryInformationSizesBox` entry shall be 0.

NOTE This means that for version other than 0, the index of CENC SAI into this box for a given sample depends on the number of previous samples with non-zero `isProtected`; retrieving this information through the `SampleAuxiliaryInformationSizesBox` and `SampleAuxiliaryInformationOffsetsBox` can be easier.

Derived specifications may further restrict the content of the `SampleEncryptionBox`, for example by enforcing that all samples in a track fragment are either protected or unprotected.

The following flags are defined for `SampleEncryptionBox`:

`senc_use_subsamples`: flag mask is 0x000002. This flag shall not be set if the version is other than 0.

The variable `UseSubSampleEncryption` is set as follows:

— if the version of the `SampleEncryptionBox` is 0 and the flag `senc_use_subsamples` is set, `UseSubSampleEncryption` is set to 1,

— otherwise, if the version of the `SampleEncryptionBox` is not 0 and the sample description entry associated with the sample uses a protection scheme mandating usage of subsamples for the described media type, `UseSubSampleEncryption` is set to 1,

— otherwise, `UseSubSampleEncryption` is set to 0.

7.2.2 Syntax

```
aligned(8) class SampleEncryptionBox extends FullBox('senc', version, flags)
{
   unsigned int(32)  sample_count;
   {
      if (version==0) {
         unsigned int(Per_Sample_IV_Size*8) InitializationVector;
         if (UseSubSampleEncryption) {
            unsigned int(16) subsample_count;
            {
               unsigned int(16) BytesOfClearData;
               unsigned int(32) BytesOfProtectedData;
            } [subsample_count ]
         }
      } else if ((version==1) && isProtected){
         unsigned int(16) multi_IV_count;
         for (i=1; i <= multi _IV_count; i++) {
            unsigned int(16) multi_subindex_IV;
            unsigned int(Per_Sample_IV_Size*8) IV;
         }
         unsigned int(32) subsample_count;
         {
            unsigned int(16) multi_subindex;
            unsigned int(16) BytesOfClearData;
            unsigned int(32) BytesOfProtectedData;
         } [subsample_count]
      } else if ((version==2) && isProtected) {
         unsigned int(Per_Sample_IV_Size*8) InitializationVector;
         if (UseSubSampleEncryption) {
            unsigned int(16) subsample_count;
            {
               unsigned int(16) BytesOfClearData;
               unsigned int(32) BytesOfProtectedData;
            } [subsample_count ]
         }
      }
   }[ sample_count ]
}
```

7.2.3 Semantics

`sample_count` is the number of CENC SAI coded in the `SampleEncryptionBox`. For version 0, it shall be either 0 or the number of samples in the track or track fragment where the `SampleEncryptionBox` is contained. For versions other than 0, it shall be the number of protected samples in the track or track fragment where the `SampleEncryptionBox` is contained.

`InitializationVector` shall conform to the definition specified in subclause 9.2. Only one `Per_Sample_IV_Size` shall be used within a track, or `Per_Sample_IV_Size` shall be zero when a sample is unencrypted or a constant IV is in use. Selection of `InitializationVector` values should follow the recommendations of subclause 9.2.

`subsample_count` shall conform to the definition specified in subclause 9.1.

`BytesOfClearData` shall conform to the definition specified in subclause 9.1.

`BytesOfProtectedData` shall conform to the definition specified in subclause 9.1.

`multi_IV_count`, `multi_subindex_IV`, `IV` and `multi_subindex` shall conform to the definition specified in Clause 7.

8 Box definitions

8.1 Protection system specific header box

8.1.1 Definition

Box Type: `'pssh'`

Container: `MovieBox`, `MovieFragmentBox` or `MetaBox` if no `MovieBox` and no `MovieFragmentBox`

Mandatory: No

Quantity: Zero or more

The `ProtectionSystemSpecificHeaderBox` contains information needed by a content protection system to play back the content. The data format is specified by the system identified by `SystemID`, and is considered opaque for the purposes of this document. For fragmented tracks, the collection of `ProtectionSystemSpecificHeaderBox`es from the initial `MovieBox`, 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 DRM systems, by including `ProtectionSystemSpecificHeaderBox`es 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:

— For media tracks, examine all `ProtectionSystemSpecificHeaderBox`es in the `MovieBox` and in the `MovieFragmentBox` associated with the sample (but not those in other `MovieFragmentBox`es). For image items, examine all `ProtectionSystemSpecificHeaderBox`es in the `MetaBox` or in the `MovieBox`

— 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 `TrackEncryptionBox` or `ItemEncryptionBox` or the `KID` field of the appropriate sample group description entry) with one of the `KID` values in the `ProtectionSystemSpecificHeaderBox`. Boxes without a list of applicable `KID` values, or with an empty list, shall be considered to apply to all `KID`s in the file or movie fragment.

The data in a `ProtectionSystemSpecificHeaderBox` is associated with samples based on a matching `KID` value in the `ProtectionSystemSpecificHeaderBox` and sample group description or default `TrackEncryptionBox` or `ItemEncryptionBox` 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 `ProtectionSystemSpecificHeaderBox`es for the remaining samples shall be stored following the above requirements.