INTERNATIONAL STANDARD

ISO/IEC 23001-7

Third edition
2016-02-15

AMENDMENT 1
2019-09

Information technology — MPEG systems technologies —

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

AMENDMENT 1: AES-CBC-128 and key rotation

Technologies de l'information — Technologies des systèmes MPEG —

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

AMENDEMENT 1: AES-CBC-128 et rotation des clés

Reference number
ISO/IEC 23001-7:2016/Amd.1:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

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).

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) or the IEC list of patent declarations received (see http://patents.iec.ch).

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.

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*.

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

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.

Information technology — MPEG systems technologies —

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

AMENDMENT 1: AES-CBC-128 and key rotation

Clause 2

Add the following new normative references:

ISO/IEC 23008-2, Information technology — *High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding*

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

3.1

Insert a new 3.1.8 and renumber current 3.1.8 as 3.1.9:

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

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

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

4.2

Add the following item to the list:

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

Clause 6

In paragraph 4, replace:

```
{
   unsigned int(8)      reserved = 0;
   unsigned int(4)      crypt_byte_block;
   unsigned int(4)      skip_byte_block;
   unsigned int(8)      isProtected;
   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;
   }
}
```

with

```
{
   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;
      }
   }
}
```

Clause 6

Add a new item to the list in paragraph 5 (insert before semantics of `isProtected`):

- `multi_key_flag` indicates that the multiple key 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.

Add a new item to the list in paragraph 5 (insert after semantics of `isProtected`):

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

7.1

Replace the sample auxiliary information in paragraph 3 with:

```
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(8) 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]

   }

}
```

Following the sample auxiliary information, add the following at the end of the 'where' list (after semantics of `BytesOfProtectedData`):

`multi_IV_count` indicates the number of entries in the initialization vector loop;

`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 track 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 track sample 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;

`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.

7.2.2

Replace the content of 7.2.2 with the following:

```
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 (flags & 0x000002) {

            unsigned int(16) subsample_count;

            {

               unsigned int(16) BytesOfClearData;

               unsigned int(32) BytesOfProtectedData;

            } [subsample_count ]

         }
      } else if (version==1) {
         unsigned int(16) multi_IV_count;
         for (i=1; i <= multi _IV_count; i++) {
            unsigned int(8) 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]
      }
   }[ sample_count ]
}
```

7.2.3

Add the following to the list of semantics:

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

8.1.1

Replace

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

with

Container: Movie (‘`moov`’) or Movie Fragment (‘`moof`’) or ‘meta’ if no Movie (‘`moov`’)

8.2

Add the following new subclauses after 8.2:

8.3 Item encryption box

8.3.1 Definition

Box Type: `‘ienc’`

Container: Item Properties Box

Mandatory (per item): Yes for protected items using schemes defined in this document

Quantity (per item): At most one associated with any item

Quantity: Zero or one

Items as defined in ISO/IEC 14496-12 may be protected using the schemes defined in this document. In this case, such items shall have an associated `ItemEncryptionBox` property and an associated auxiliary item with an `aux_info_type` matching the protection scheme used, as defined in 8.4. The payload of that auxiliary item shall be exactly one `CencSampleAuxiliaryDataFormat` as defined in 7.1.

The `ItemEncryptionBox` contains default values for the `isProtected` flag, `Per_Sample_IV_Size`, and `KID` for the item. In the case where pattern-based encryption is in effect, it supplies the pattern and when Constant IVs are in use, it supplies the Constant IV. These values are used as the encryption parameters for the item in this meta box. For files with only one key for all items, this property box allows the basic encryption parameters to be specified once for all items instead of being repeated per item.

Items sharing this property are always protected; consequently, the default value for `isProtected` field (see 9.1) is 1.

If the value `Per_Sample_IV_Size` is 0, then the `constant_IV_size` for all items that use these settings SHALL be present. A Constant IV SHALL NOT be used with counter-mode encryption.

`ItemEncryptionBox` properties shall be marked as essential in `ItemPropertyAssociation`.

NOTE The version field of the `ItemEncryptionBox` is set to a value greater than zero when the pattern encryption defined in 9.6 is used and to zero otherwise.

8.3.2 Syntax

```
aligned(8) class ItemEncryptionBox extends ItemFullProperty(‘ienc’, version, flags=0)

{

      unsigned int(8) reserved = 0;

      if (version==0) {

            unsigned int(8) reserved = 0;

      } else { // version is 1 or greater

            unsigned int(4) crypt_byte_block;

            unsigned int(4) skip_byte_block;

      }

      unsigned int(8) num_keys;

      for (i=1; i<= num_keys; 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;

         }

      }

}
```