INTERNATIONAL STANDARD

ISO/IEC 23000-21

First edition
2019-07

Information technology — Multimedia application format (MPEG-A) —

Part 21: Visual identity management application format

Technologies de l'information — Format pour application multimédia (MPEG-A) —

Partie 21: Format pour application de gestion d'identité visuelle

Reference number
ISO/IEC 23000-21:2019(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

Contents

Page

Foreword ..... **iv**
Introduction ..... **v**
1 Scope **1**
2 Normative references **1**
3 Terms and definitions **1**
4 Abbreviated terms **2**
5 System for identity privacy management **2**
5.1 General framework 2
5.2 Applying privacy protection in ISO/IEC 21000-22 4
5.2.1 General description 4
5.2.2 User description 5
5.2.3 Context description 9
5.2.4 Service description 12
6 Content sensitive encryption **14**
6.1 Overview of content sensitive encryption 14
6.2 Content sensitive encryption for Rec. ITU-T H.264 | ISO/IEC 14496-10 15
6.2.1 General 15
6.2.2 Content sensitive encryption with CAVLC entropic coding 15
6.2.3 Content sensitive encryption with CABAC entropic coding 16
6.3 Content sensitive encryption for HEVC 17
6.4 Content sensitive encryption for region encryption 17
6.4.1 General 17
6.4.2 AVC 17
6.4.3 HEVC 20
7 Support for protected streams at system level **21**
7.1 Signalization of protected stream 21
7.2 Signal of multiple access in protected stream 22
7.3 Signal of content sensitive encryption 27
7.3.1 Definition of content sensitive encryption 27
7.3.2 Content sensitive encryption applied to a video NAL unit 28
7.3.3 `'svel'` AES-CTR sensitive encryption scheme 28
Annex A (normative) **Content sensitive encryption scheme** **30**
Bibliography **39**

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 23000 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.

Introduction

The main goal of the ISO/IEC 23000 series (also known as "MPEG-A") is to facilitate the swift development of innovative, standards-based multimedia services and applications by selecting and combining readily tested and verified tools taken from the MPEG body of standards.

Visual identity management is designed to enable users to control and manage privacy protection by defining a new framework and tools. It also provides to industry a coherent and consistent approach to manage privacy protection in order to be implemented in a variety of scenarios, applications or systems.

The main objective of preserving privacy protection is to enable security and confidentiality in the multimedia content chain. Many usages of image/video communication services, social networking and video sharing platforms have led to an increasing interest to protect users' privacy.

Traditionally, multimedia data security is achieved by cryptography solutions, which deal with encryption of data. This approach is called Naive Encryption Algorithm (NEA) and it treats the video bitstream as text data without paying attention to the structure of the compressed video. To this end, MPEG common encryption has been standardized in order to support encryption and key mapping methods for file format in ISO/IEC 23001-7 and for transport streaming in ISO/IEC 23001-9[3]. Consequently, bitstreams encrypted by those documents are decodable only after a correct decryption process even when only parts of the video are encrypted. Nevertheless, none of these formats allow signalling the encryption of a part of the picture (region), or indicating to the decoder that the encrypted bitstream can be partially decoded.

Moreover, all the access control is provided and performed globally without taking into account the image/video content and context. To restore citizens' confidence in online data collection practices, submitted media should be encrypted to protect privacy and only viewed with limited access that the user chooses: group of people, purpose of sharing, time, date, metadata, etc.

In order to provide privacy protection over processing and sharing of multimedia content, a flexible, effective and scalable mechanism is required to provide users a way to express their control desires in a form that can be processed and monitored systematically, consistently and persistently throughout the lifecycle of the multimedia content. There is currently no standardized format to represent privacy description information (PDI), hindering the interoperability between secured systems.

Information technology — Multimedia application format (MPEG-A) —

Part 21:
Visual identity management application format

1 Scope

This document specifies the standard representation of the set of signalling and data used in the process of preserving privacy for storage sharing image/video.

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.

Rec. ITU-T H.264 | ISO/IEC 14496-10:—[1)], *Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding*

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 23001-7:2016, *Information technology — MPEG systems technologies — Part 7: Common encryption in ISO base media file format files*

Rec. ITU-T H.265 | ISO/IEC 23008-2:—[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*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 14496-10, ISO/IEC 23008-2, ISO/IEC 23008-12 and the following apply.

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

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

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

3.1
CSE
content sensitive encryption
selective encryption
image or video content protection scheme that can encrypt only a subset of the compressed bitstream data, preserving format compliant

1) Under preparation. Stage at the time of publication: ISO/IEC/DIS 14496-10:2018.

2) Under preparation. Stage at the time of publication: ISO/IEC/DIS 23008-2:2018.

3.2
privacy description management
entity capable of managing the access control of different regions and associated with different authorizations

3.3
ROI
region of interest
sample or subset of an image within a media, identified for a particular purpose

4 Abbreviated terms

AVC	advanced video coding (as specified by Rec. ITU-T H.264 | ISO/IEC 14496-10)
CABAC	context-adaptive binary arithmetic coding (as specified in Rec. ITU-T H.264 | ISO/IEC 14496-10 and in Rec. ITU-T H.265 | ISO/IEC 23008-2)
CAVLC	context-adaptive variable-length coding (as specified in Rec. ITU-T H.264 | ISO/IEC 14496-10)
CTU	coding transform unit (as specified by Rec. ITU-T H.265 | ISO/IEC 23008-2)
HEVC	high efficiency video coding (as specified by Rec. ITU-T H.265 | ISO/IEC 23008-2)
MB	Macroblock unit (as specified by Rec. ITU-T H.264 | ISO/IEC 14496-10)
MPEG 21 UD	MPEG 21 user description (as specified by ISO/IEC 21000-22)
NAL	network abstraction layer (as specified in Rec. ITU-T H.264 | ISO/IEC 14496-10 and in Rec. ITU-T H.265 | ISO/IEC 23008-2)
SEI	supplemental enhancement information (as specified in Rec. ITU-T H.264 | ISO/IEC 14496-10 and in Rec. ITU-T H.265 | ISO/IEC 23008-2)

5 System for identity privacy management

5.1 General framework

To protect privacy content, stored and/or shared media should be encrypted by the service's user and should only be viewed with a well-defined limited access (group of people, purpose of sharing, time, date, metadata, etc.). Consequently, some particular regions of the video (e.g. human faces, text data) can only be seen by the authorized users, regardless of who captures and shares the video. Additionally, since multiple regions sometimes need different protections (e.g. multiple faces shown in the video), there is also a need to manage different control access (i.e. different key identifiers) within the same video bitstream, potentially for each frame, as shown in Figure 1.

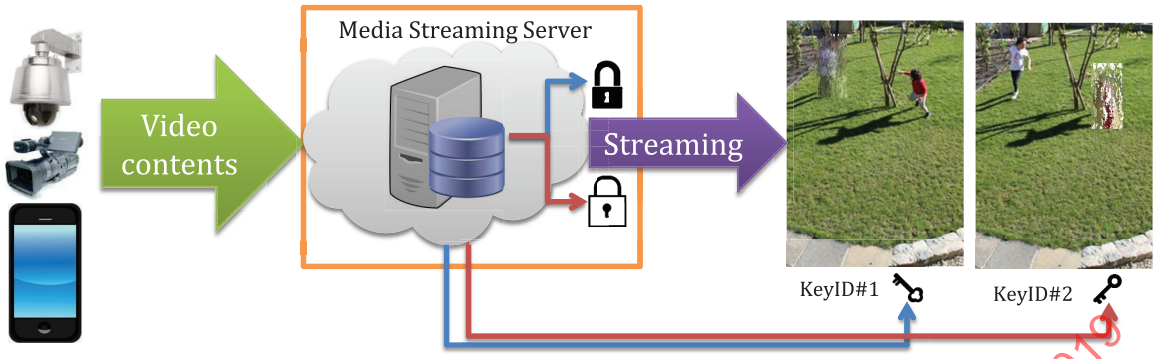

Figure 1 — Privacy management in multimedia streaming applications

Figure 2 illustrates an example of framework for managing the privacy of users when pictures or videos are taken and shared among users with different keys.

NOTE Key management is outside the scope of this document.

The steps of the privacy protection mechanism can be expressed as follows (see also Figure 2):

1. User 1:
   a. Capture a media.
   b. Select part of the media considered as 'private' or/and let an application detect and recognize automatically faces.
   c. Transmit information to the privacy description management.
2. Privacy description management:
   a. Get information and manage access control thanks to privacy policy defined by User 1 with the different relative user descriptions, potentially taking into account context descriptions and/or service descriptions through the recommendation engine.
   b. Send a unique ID of the media, and a list of encryption keys associated to different part of the media.
3. User 1:
   a. Generation (i.e. compression and encapsulation) of media file with an encryption scheme fed by the list of encryption keys and their locations.
   b. Storage or transmission to dedicated server for media sharing.
4. User 2:
   a. Get media file.
   b. Send the ID of media and associated context and user description to privacy description management.
5. Privacy description management:
   a. Use the transmitted information through the recommendation engine to evaluate whether the user can be totally or partially authorized to render the associated media.

b. Send decryption keys adapted to the users that are allowed or not to see each part of the media.

6. User 2:

a. Get decryption keys.

b. Render (i.e. de-capsulate and decompress) media with an appropriated decryption scheme depending on associated authorization.

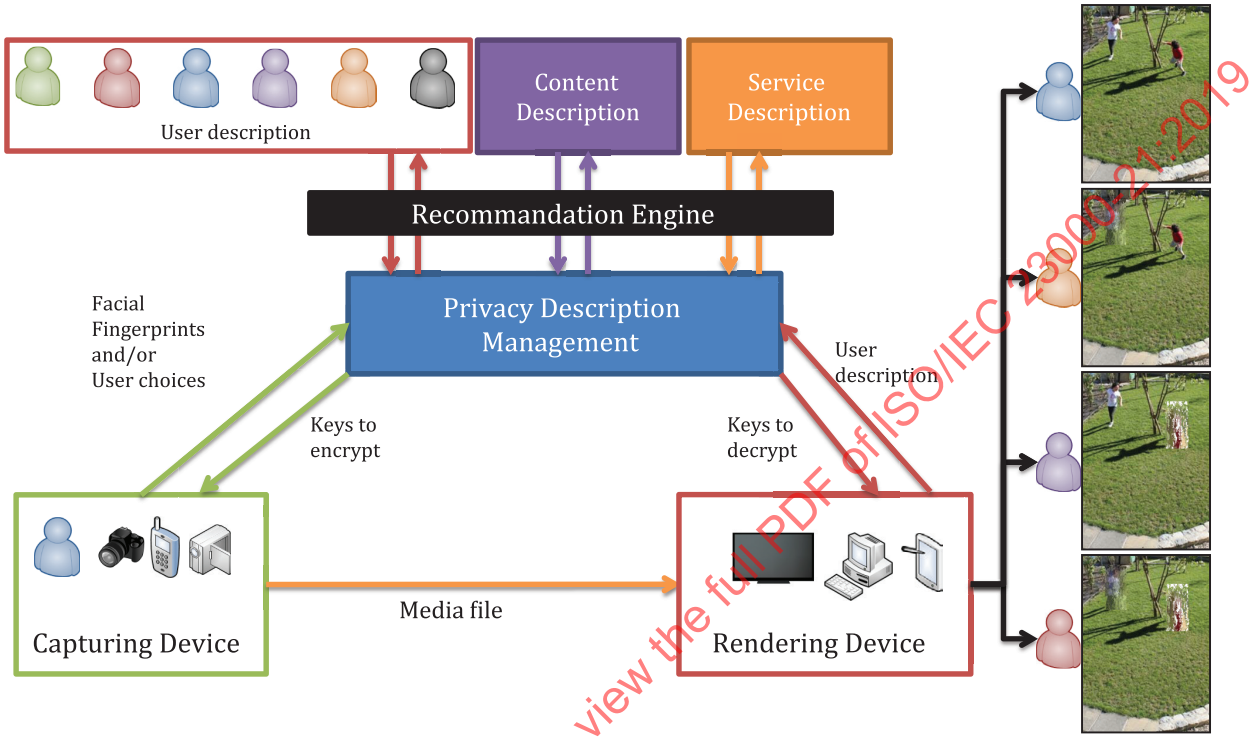

Figure 2 — Proposed framework for privacy management of media

ISO/IEC 15938-13[5] should be used to represent the fingerprint of the face of a user in a picture. For video, compact descriptors for video analysis (ISO/IEC 15938-15[6][3]) can be used. To define the area that needs to be protected, it can be selected manually or automatically, but this process is out of the scope of this document.

5.2 Applying privacy protection in ISO/IEC 21000-22

5.2.1 General description

5.2.1.1 General

This subclause refers to ISO/IEC 21000-22:—[2][4], Clause 4, that defines basic properties of each sub-element for UD (user description), CD (context description), SD (service description) and RD (recommendation description).

NOTE To consider privacy issues, several modifications and new specific types have been added to fit the requirements of ISO/IEC 21000-22.

3) Under preparation. Stage at the time of publication: ISO/IEC/FDIS 15938-15:2019.

4) Under preparation. Stage at the time of publication: ISO/IEC/FDIS 21000-22:2019.

5.2.1.2 InformationAccessUserGroup

5.2.1.2.1 Syntax

This syntax is `InformationAccessUserGroup` type.

```
<complexType name="InformationAccessUserGroup">
   <sequence>
      <element name="UserID" type="mpeg7:UserIdentifierType" maxOccurs="unbounded"/>
   </sequence>
   <attribute name="GroupID" type="anyURI"/>
</complexType>
```

`InformationAccessUserGroup` type has `InformationAccessID` type. `InformationAccessID` describes the list of other users who can access private information.

5.2.1.2.2 Semantics

Semantics of the `InformationAccessUserGroup` is given in Table 1.

Table 1 — Semantics of the `InformationAccessUserGroup`

Name	Definition
`InformationAccessUserGroup`	Describes the group of other users who can access private information.
`UserID`	Specifies the list of user's ID belonging to the group that can access private information.
`GroupID`	Specifies the identifier of the group.

5.2.2 User description

5.2.2.1 General

This subclause specifies user description (UD), which contains root elements at the basis of individual use cases.

5.2.2.2 BaseUserType

5.2.2.2.1 General

`BaseUserType` was created in `UserDescription`, and `InformationAccessID` and `InformationAccessID` were added to `BaseUserType`.

`InformationAccessID` describes the list of other users who can access private information of a given user. `InformationAccessGroupURI` references groups of users who can access private information.

5.2.2.2.2 Syntax

This syntax is `BaseUserType` type.

```
<complexType name="BaseUserType" abstract="true">
   <sequence>
      <element name="InformationAccessID" type="mpeg7:UserIdentifierType" minOccurs="0"
maxOccurs="unbounded"/>
      <element name="InformationAccessGroupURI" type="anyURI" minOccurs="0"
maxOccurs="unbounded"/>
   </sequence>
</complexType>
```

5.2.2.2.3 Semantics

Semantics of the `BaseUserType` type is given in Table 2.

Table 2 — Semantics of the `BaseUserType`

Name	Definition
`BaseUserType`	`BaseUserType` is an abstract type providing a base for description of each element.
`InformationAccessID`	Specifies the list of user's ID of other users who can access private information.
`InformationAccessGroupURI`	References groups of users who can access private information through the specification of its URI. Refer to the `GroupID` of `InformationAccessUserGroup` in `General Description`.

5.2.2.3 UserDescriptionType

5.2.2.3.1 General

This subclause describes a structure of `UserDescriptionType` data type. The `UserDescriptionType` contains several elements as the `InformationAccessGroup` that has been added and which specifies a list of users who can access the privacy information.

5.2.2.3.2 Syntax

```
<complexType name="UD">
  <complexContent>
     <extension base="ud:BaseUserType">
             <sequence>
                  <element name="InformationAccessGroup"
type="ct:InformationAccessUserGroup" minOccurs="0" maxOccurs="unbounded"/>
                  <element name="ClassificationSchemeAlias"
type="ct:ClassificationSchemeAliasType" minOccurs="0" maxOccurs="unbounded"/>
                  <element name="UserID" type="mpeg7:UniqueIDType"/>
                  <element name="UserProfile" type="ud:UserProfileType" minOccurs="0"/>
                  <element name="UsageHistory" type="ud:UsageHistoryType" minOccurs="0"/>
                  <element name="Preference" type="ud:PreferenceType" minOccurs="0"/>
                  <element name="Emotion" type="ud:EmotionType" minOccurs="0"/>
                  <element name="Schedule" type="ud:ScheduleType" minOccurs="0"/>
                  <element name="Activity" type="ud:ActivityType" minOccurs="0"/>
                  <element name="Representation" type="ct:ObjectType" minOccurs="0"/>
                  <element name="Intention" type="ud:IntentionType" minOccurs="0"/>
                  <element name="Knowledge" type="ud:KnowledgeType" minOccurs="0"/>
                  <element name="ObjectSharing" type="ud:ObjectSharingType"
minOccurs="0"/>
                  <element name="ServiceUsagePattern" type="ud:UsagePatternType"
minOccurs="0"/>
                  <element name="LoudnessPreferences" type="ud:LoudnessPreferencesType"/>
                  <element name="VisualExpressionPreference" type="ud:
VisualExpressionPreferenceType"/>
             </sequence>
             <attributeGroup ref="ct:commonAttributes"/>
     </extension>
  </complexContent>
</complexType>
```

5.2.2.3.3 Semantics

Semantics of the `UserDescriptionType` is given in Table 3.

Table 3 — Semantics of the `UserDescriptionType`

Name	Definition
`UD`	Serves as the root element of the MPEG 21 UD format. The `UD` element shall be used as the topmost element to make user description in an instance of MPEG 21 UD format.
`UserDescriptionType`	Specifies the syntax of the root element. This datatype is a set of descriptions which may contain static and dynamic information about user. Within this Type, `UserProfile`, `Preference`, `Emotion`, `Schedule` or `Activity` element shall be instantiated.
`InformationAccessGroup`	Describes the group of other users who can access private information.

Table 3 *(continued)*

Name	Definition
ClassificationSchemeAlias	Specifies an alias for a ClassificationScheme to be referenced within the UserDescriptionType by a simplified URI.
UserID	Describes the unique identifier of a user.
UserProfile	Describes user profile based on UserProfileType.
UsageHistory	Describes usage history based on UsageHistoryType. This element can represent user's history for a given service, such as searching or movie recommendations.
Preference	Describes preference based on PreferenceType.
Emotion	Describes emotion based on EmotionType. This type represents user's emotion, including its changes over time.
Schedule	Describes schedule based on ScheduleType.
Activity	Describes user activity based on ActivityType.
Representation	Describes the representation such as user character and image used by the user. This element can be used to describe user character on the social network.
Intention	Describes a list of intentions related to specific actions which a given user may perform with a multimedia object.
Knowledge	Describes a piece of knowledge that the user wants to share for recommendation purposes.
ObjectSharing	Describes the condition of authority (e.g. ID, ownership and accessibility of object) for the sharing and accessing of object to user or external service.
UsagePattern	Describes various patterns about usage of user based on UsagePatternType. Users use a lot of applications and services for a variety of reasons. Usagepattern can be used to figure out user's characteristics.
commonAttributes	Describes a group of attributes for the CommonAttributes. The syntax and semantics of commonAttributes are specified in General Description (see ISO/IEC 21000-22:—, Clause 4).
ServiceUsagePattern	
LoudnessPreferences	Describes static or dynamic personal and group audio preferences (e.g. audio levels, audio level history, and song preferences) and the usage history which helps with selecting appropriate audio service.
VisualExpressionPreference	A set of data edited by user's intention/emotion or created by recognizing user's situation/condition.

5.2.2.4 SocialInformationType

5.2.2.4.1 Syntax

```
<complexType name="SocialInformationType">
   <sequence>
      <element name="ServiceID" type="sd:ServiceDescriptionType" minOccurs="0"/>
      <element name="LoginID" type="ud:UserDescriptionType"/>
      <element name="LoginPassword" >
            <simpleType>
                  <restriction base="string">
                        <minLength value="8"/>
                        <maxLength value="512"/>
                  </restriction>
            </simpleType>
      </element>
      <element name="Nickname" type="Name" minOccurs="0"/>
      <element name="GroupID" type="mpeg7:UserIdentifierType" minOccurs="0"/>
      <element name="FriendUserID" type="mpeg7:UserIdentifierType" minOccurs="0"
maxOccurs="unbounded"/>
      <element name="PrivateInformationAccessID" type="mpeg7:UserIdentifierType"
minOccurs="0" maxOccurs="unbounded"/>
   </sequence>
</complexType>
```

5.2.2.4.2 Semantics

Semantics of the `SocialType` is given in Table 4.

Table 4 — Semantics of the `SocialType`

Name	Definition
socialinformationType	Describes a specific information of a given user, provided to a specific service, mainly referred to a social community.
ServiceID	Describes the service used by the user.
LoginID	Describes the login ID of the user.
LoginPassword	Describes the password.
Nickname	Describes the nickname of the user, if any.
GroupID	Describes the group which the user belongs to.
FriendUserID	Describes the list of friends.
PrivateInformationAccessID	Describes the list of other users who can access private information of a given user.

5.2.3 Context description

5.2.3.1 General

This subclause specifies context description (CD), which contains root elements at the basis of individual use cases.

5.2.3.2 BaseContextType

5.2.3.2.1 General

`BaseContextType` already exists in `ContextDescription`, so `InformationAccessID` and `InformationAccessGroupURI` to `BaseContextType` have been added.

`InformationAccessID` describes the list of other users who can access private information of a given user. `InformationAccessGroupURI` references to groups of users who can access private information. All types have been modified using the defined type of MPEG 21 UD in `ContextDescription`. `BaseContextType` is as follows.

5.2.3.2.2 Syntax

This syntax is `BaseContextType` type.

```
<complexType name="BaseContextType" abstract="true">
   <sequence>
      <element name="InformationAccessID" type="mpeg7:UserIdentifierType" minOccurs="0"
maxOccurs="unbounded"/>
      <element name="InformationAccessGroupURI" type="anyURI" minOccurs="0"
maxOccurs="unbounded"/>
   </sequence>
   <attribute name="InfoSource" type="anyURI"/>
</complexType>
```

5.2.3.2.3 Semantics

Semantics of the `BaseContextType` type is given in Table 5.

Table 5 — Semantics of the `BaseContextType`

Name	Definition
`BaseContextType`	`BaseContextType` is an abstract type providing a base for description of each element.
`InformationAccessID`	Specifies the list of user's ID of other users who can access private information.
`InformationAccessGroupURI`	References groups of users who can access private information through the specification of its URI. Refer to the `GroupID` of `InformationAccessUserGroup` in `General Description`.

5.2.3.3 ContextDescriptionType

5.2.3.3.1 General

This subclause describes a structure of `ContextDescriptionType` data type. The `ContextDescriptionType` contains several elements as the `InformationAccessGroup` specifies a list of users who can access the privacy information.

5.2.3.3.2 Syntax

This syntax is `BaseContextType` type.

```
<element name="CD" type="cd:ContextDescriptionType"/>
   <complexType name="BaseContextType" abstract="true">
      <attribute name="InfoSource" type="anyURI"/>
   </complexType>
   <complexType name="ContextDescriptionType">
      <sequence>
            <element name="InformationAccessGroup"
type="ct:InformationAccessUserGroup" minOccurs="0" maxOccurs="unbounded"/>
            <element name="ClassificationSchemeAlias"
type="ct:ClassificationSchemeAliasType" minOccurs="0" maxOccurs="unbounded"/>
            <element name="ContextIdentification"
type="cd:ContextIdentificationType" minOccurs="0"/>
            <element name="ValidTimeDuration" type="ct:TimeType" minOccurs="0"/>
            <element name="Season" type="mpeg7:termReferenceType" minOccurs="0"/>
            <element name="DeviceCharacteristics" type="cd:DeviceCharacteristicsType"
minOccurs="0" maxOccurs="unbounded"/>
            <element name="NetworkInfo" type="cd:NetworkInfoType" minOccurs="0"
maxOccurs="unbounded"/>
            <element name="Location" type="ct:LocationType" minOccurs="0"/>
            <element name="Weather" type="cd:WeatherType" minOccurs="0"/>
            <element name="OtherEnvironmentalInfo" type="cd:OtherEnvironmentalInfoType"
minOccurs="0"/>
            <element name="OtherContextInfo" type="anyType" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
      <attributeGroup ref="ct:commonAttributes"/>
      <attribute name="Priority" type="ct:ZeroToOnehundredOrdinalType" use="optional"/>
</complexType>
```

5.2.3.3.3 Semantics

Semantics of the `BaseContextType` type is given in Table 6.

Table 6 — Semantics of the `BaseContextType`

Name	Definition
`CD`	This data element is the root element of the context description.
`ContextDescriptionType`	This data type contains static and dynamic information about the user context.
`InformationAccessGroup`	Describes the group of other users who can access private information.
`ClassificationSchemeAlias`	Specifies an alias for a `ClassificationScheme` to be referenced within the `UserDescriptionType` by a simplified URI.
`ValidTimeDuration`	Describes valid time duration for context description. The syntax and semantics of `PeriodOfTimeType` are specified in General Description (see ISO/IEC 21000-22:—, Clause 4).
`Season`	Describes current season information according to the region.

Table 6 *(continued)*

Name	Definition
DeviceCharacteristics	Describes general characteristics of the terminal.
NetworkInfo	Describes network related information.
Location	Describes current location when a service is requested. The syntax and semantics of PlaceType are specified in ISO/IEC 15938-5.
Weather	Describes current weather when a service is requested.
OtherEnvironmentalInfo	Describes environmental information of noise or illumination characteristics around user.
commonAttributes	Describes a group of attributes for the CommonAttributes. The syntax and semantics of commonAttributes are specified in General Description Type (see ISO/IEC 21000-22:—, Clause 5).
Priority	Describes the level of priority of the context description by using the ct:ZeroToOnehundredOrdinalType.
OtherContextInfo	A placeholder for other context – related information outside the standard namespace.

5.2.4 Service description

5.2.4.1 General

This subclause specifies context description (SD) which contains root elements at the basis of individual use cases.

5.2.4.2 BaseServiceType

5.2.4.2.1 General

BaseServiceType was created in ServiceDescription, and InformationAccessID and InformationAccessGroupURI were added to BaseServiceType.

InformationAccessID describes the list of other users who can access private information of a given user. InformationAccessGroupURI references groups of users who can access private information.

5.2.4.2.2 Syntax

This syntax is BaseServiceType type.

```
<complexType name="BaseServiceType" abstract="true">
   <sequence>
      <element name="InformationAccessID" type="mpeg7:UserIdentifierType" minOccurs="0"
maxOccurs="unbounded"/>
      <element name="InformationAccessGroupURI" type="anyURI" minOccurs="0"
maxOccurs="unbounded"/>
   </sequence>
</complexType>
```

5.2.4.2.3 Semantics

Semantics of the BaseServiceType type is given in Table 7.

Table 7 — Semantics of the `BaseServiceType`

Name	Definition
`BaseServiceType`	`BaseServiceType` is an abstract type providing a base for description of each element.
`InformationAccessID`	Specifies the list of user's ID of other users who can access private information.
`InformationAccessGroupURI`	References groups of users who can access private information through the specification of its URI. Refer to the `GroupID` of `InformationAccessUserGroup` in `General Description`.

5.2.4.3 ServiceDescriptionType

5.2.4.3.1 General

This subclause describes a structure of `ServiceDescriptionType` data type. The `ServiceDescriptionType` contains several elements as the `InformationAccessGroup` that has been added and which specifies a list of users who can access the privacy information.

5.2.4.3.2 Syntax

```
<complexType name="SD">
   <complexContent>
      <extension base="sd:BaseServiceType">
            <sequence>
                  <element name="InformationAccessGroup"
type="ct:InformationAccessUserGroup" minOccurs="0" maxOccurs="unbounded"/>
   <sequence>
      <element name="ClassificationSchemeAlias"
type="ct:ClassificationSchemeAliasType" minOccurs="0" maxOccurs="unbounded"/>
      <element name="ServiceID" type="mpeg7:UniqueIDType"/>
      <element name="ServiceGeneralInformation" type="sd:ServiceGeneralInformationType"/>
      <element name="ServiceTargetInformation" type="sd:ServiceTargetInformationType"/>
      <element name="ServiceInterfaces" type="sd:ServiceInterfacesType" minOccurs="0"/>
      <element name="InternalServices" type="sd:InternalServicesType" minOccurs="0"/>
      <element name="Priority" type="ct:ZeroToOnehundredOrdinalType" minOccurs="0"/>
      <element name="IsServiceAvailable" type="boolean"/>
      <element name="ServiceObjectsInformation" type="sd:ServiceObjectType"
minOccurs="0"/>
      <element name="LoudnessControlServiceDescription" type="LoudnessInfoType"/>
            <element name="VisualExpressionServiceDescription"
type="SD:VisualExpression"/>
   </sequence>
      </extension>
   </complexContent>
</complexType>
```

5.2.4.3.3 Semantics

Semantics of `ServiceDescriptionType` is given in Table 8.

Table 8 — Semantics of the `ServiceDescriptionType`

Name	Definition
`SD`	Specifies the root element of the service description.
`ServiceDescriptionType`	Specifies the syntax of the root element containing information about a specific service.
`InformationAccessGroup`	Describes the group of other users who can access private information.
`ClassificationSchemeAlias`	The classification scheme alias information.
`ServiceID`	Describes the unique ID of the service.
`ServiceGeneralInformation`	Describes general information about a given service.
`ServiceTargetInformation`	Describes a user's preferred service.
`ServiceInterfaces`	Describes interfaces used for exchanging information among the services and/or recommendation engine and/or application.
`IntermalServices`	Describes a specific internal service.
`Priority`	Specifies the priority level associated to a given service.
`IsServiceAvailable`	Specifies availability of the service.
`ServiceObjectsInformation`	Describes the information of objects provided by a service.

6 Content sensitive encryption

6.1 Overview of content sensitive encryption

In this clause, a content sensitive encryption (CSE) method is introduced, which modifies the message content in such a way that an unauthorized receiver (i.e. any standard decoder) can normally decode any ciphered stream, while the displayed content is made non-intelligible by the encryption. Moreover, this method can encrypt only the region of interest (ROI) in the media content (human faces, confidential data, etc.) leaving the rest of the video (background) unencrypted. This scheme is applicable to AVC and HEVC video codecs in ISO/IEC 14496-12[1] and in the high efficiency image file format defined in ISO/IEC 23008-12.

The most straightforward method to achieve secure video transmission is to encrypt the entire video bitstream. This approach is called full sample encryption and is standardized in ISO/IEC 23001-7 for ISO base media file format files and ISO/IEC 23001-9 for MPEG-2 transport streams[3]. Also called complete encryption, this technique treats all the compressed video bitstream as opaque data without paying attention to the compressed video structure, hence preventing any analysis of the video bitstream by transport or low-level application layers. Consequently, it brings a non-negligible computational complexity by ciphering the whole video data, and the bitstreams encrypted by those documents are solely decodable after a correct decryption process. To reduce substantially the complexity adding due to ciphering operations, the pattern encryption and the subsample encryption were also defined in ISO/IEC 23001-7, where only parts of the video data are encrypted thanks to standardized signalization. But like the full sample encryption, those encrypted bitstreams cannot be decodable without a decryption mechanism.

Contrary to the previous encryption schemes, content sensitive encryption considers the coding structure of the video compressed bitstream and encrypts only the most sensitive information in the video bitstream. CSE generates protected bitstreams that can be decoded by any compliant decoder without requiring access to the encryption key. Moreover, CSE does not affect the compress ratio.

Since the CSE takes place inside codec, it is also possible to restrain the CSE locally (i.e. on selected macroblock or CTU) and/or temporally (i.e. frame) by the segmentation of frames in slices, and by the reference picture management to prevent propagation of visual scrambling on unwanted portion of images. So the method can split the frame into independent rectangular regions (by slice or tile concepts) and then only the parts of the frame (set of slices or tiles) belonging to the ROI are encrypted.

Thus, to ensure perfect decryption and reconstruction, the exact process to encrypt (and decrypt) protected bitstream by content sensitive encryption is described in 6.2 and 6.3 and shall be carried out as specified in Annex A. Then, to prevent the propagation of encryption errors outside the ROI with temporal coding dependencies in inter-coding configuration, some non-normative coding restrictions at the encoding side are also described in 6.4.

In the case of ROI encryption, it implies that the detection of regions to protect is done before the compression step to restrict partial encryption as shown in Figure 3, and the encoder shall use slice or tile to manage regions as defined in 6.4.

NOTE The content analysis block manages the ROI detection in the video with or without supervised algorithm, and is outside the scope of this document.

If this scheme is to be applied into regions on an already compressed video, it needs to recompress the content (i.e. by transcoding) by using slices or/and tiles. Furthermore, if the video should be totally encrypted by content sensitive encryption (i.e. the full frame), it is not necessary to recompress, this encryption scheme can be directly applied in video bitstreams.

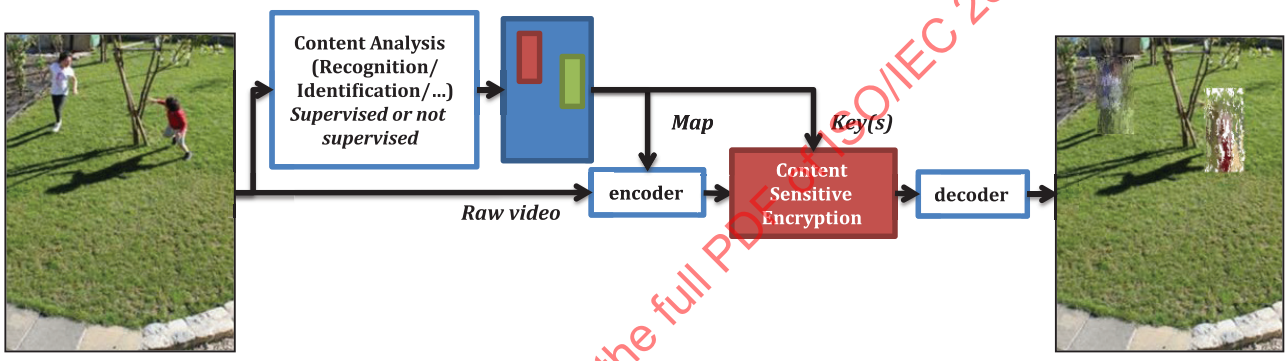

Figure 3 — ROI encryption workflow

6.2 Content sensitive encryption for Rec. ITU-T H.264 | ISO/IEC 14496-10

6.2.1 General

Rec. ITU-T H.264 | ISO/IEC 14496-10 includes two entropy coding modes, so the content sensitive encryption scheme is different and is presented in two different paragraphs.

6.2.2 Content sensitive encryption with CAVLC entropic coding

The CAVLC mode is basically based on the concept of variable length coding (VLC) where VLC tables for the various syntax elements are switched depending on the values of previously transmitted syntax elements. So the idea is to apply a ciphering that alters the video stream only visually while keeping it fully decodable even by a non-robust standard compliant decoder.

The encrypted bits are chosen with respect to the considered video standard to ensure full compatibility, achieved by selecting the bits (generally parts of code-words) for which each of the encrypted configuration modifies the decoding process contexts in the sense that their introduction does not create de-synchronization nor lead to non-compliant bitstream. As such, an encryption operation leading to a change of a symbol table used in the coding process is not negligible whereas an encryption operation that leads to interpreting a given code-word instead of another of the same size is negligible.

In each case, bits shall maintain this capacity in every coded bitstream, and cases where given configuration of bitstream allows immediate or delayed resynchronization cannot be envisaged. As such, the bits 'selected for encryption' (also called 'cipherable') correspond to cases where several code-words of the same length are available with no major context change when shifting from one to another,

and the ciphering will consist in swapping one of the bit(s) configurations for another. Carefully choosing the way encryption is performed offers two advantages: firstly, it ensures the compatibility with the requirements of the considered video standard, and secondly, it makes it difficult for cryptanalysis attacks to find an angle to break encryption key, as it is aimed at making all solutions possible, hence removing possibilities to rule out some cases based on non-respect of standard syntax.

For example, the ciphering of the code-word providing information on the **mb_qp_delta** parameter implies changing the value-change of quantification parameter (QP) in the slice data potentially for each macroblock. The value of mb_qp_delta shall be in the range of −(26 + $QpBdOffset_Y$ / 2) to +(25 + $QpBdOffset_Y$ / 2), inclusive. So this code-word is eligible to be cyphered if, and only if, its absolute value is less than:

$$2^{Floor\left(Log2(25+\mathrm{QpBdOffsetY}/2)\right)}$$

Then, the suffix bits (i.e. the bits following the first '1'), as illustrated by the bold font in Table 9, shall be selected for CSE.

Table 9 — Mb_QP_delta code-words table

Index	Mb_QP_Delta value	Code-word
0	0	1
1	1	010
2	−1	011
3	2	00100
4	−2	00101
5	3	00110
6	−3	00111
7	4	0001000
8	−4	0001001
...	...	...

Noticing that the change of the **mb_qp_delta** has no influence on the rest of the decoding, only the risk of de-synchronization rules out some configurations: it is then proposed to mark as 'selected for encryption' the suffix bits in bold font of the corresponding bits in Table 9.

Following this example, other bits can be easily 'selected for encryption' in the code words (presented in A.1) for the intra4x4 prediction mode, the **mb_qp_delta**, the intra chroma prediction mode, trailing ones, coefficient level suffix, total zeros, run before, reference id or motion vectors, etc., always keeping in mind that, before marking bit as 'selected for encryption', a check shall be made to identify if the condition on standard compliance is verified when changing those bits. In particular, blocks located on the border of the video slice are not candidates for ciphering as they are not admitting all prediction modes.

6.2.3 Content sensitive encryption with CABAC entropic coding

The proposed method performs encryption at the entropy coding level and more specifically after the binarization process. Figure 4 illustrates the encryption position within the CABAC engine. The generic design of CABAC is a combination of an adaptive binary arithmetic coding technique with a well-designed set of context models. However, due to the arithmetic encoding step, bits in CABAC compressed streams are very interdependent. The adequate spot for performing content sensitive encryption is thus after the binarization step, which allows to work on bin strings that have a meaning from a decoder point of view.

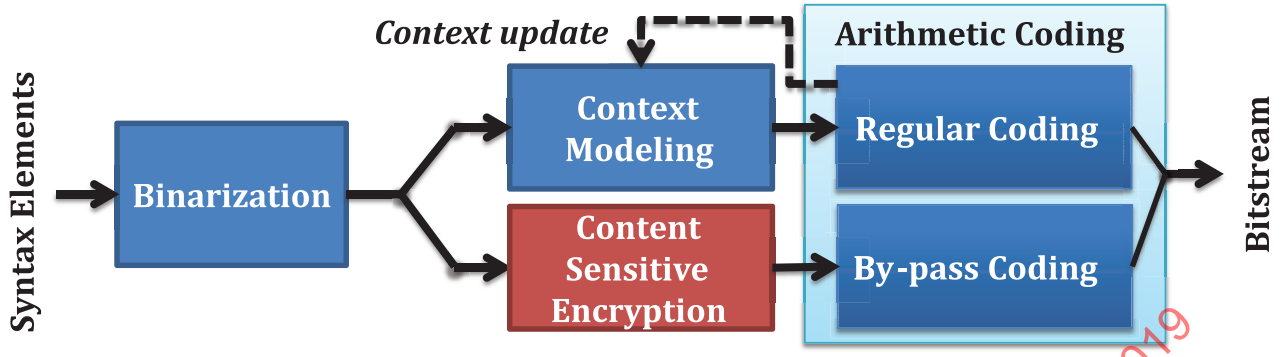

Figure 4 — Content sensitive encryption within the CABAC HEVC encoder

In the content sensitive encryption approach, a bin string is considered as cipherable if it satisfies the following constraints:

— The bin modification does not modify the overall decoding process. Namely, the interpretation of changes shall not lead to any mismatch between encoding and decoding processes (initialization and termination protocols, number/order of calls for data, selection of the related probability models, etc.).

— The encrypted symbol is still valid, which restricts modifications to a domain in which it is ensured that a standard decoder accepts the modified symbol.

Checking both constraints ensures the modified stream compatibility with the video standard. By-passed bins do not exploit and update probability models during the arithmetic coding, so that their encryption has no impact on the CABAC engine. In order to fulfil constant bit rate and format compliant encryption, only syntax elements bypass coded can be encrypted. But not all bypassed bins are automatically considered as 'cipherable', since the modification of symbols coded can lead to non-compliant bitstream. Thus, some cipherable syntax elements are in this first category of bypass encoded code words: sign of motion vector difference (MVD), sample adaptive offset (SAO) related information in HEVC, etc. In A.2, a list identifies in detail the elements of the stream that can be ciphered without disrupting a standard decoding process.

6.3 Content sensitive encryption for HEVC

As in 6.2.3 with AVC (CABAC mode), HEVC uses the same entropic coding, and the method is similar. But the list of syntax elements that fulfil constraints to be considered as 'cipherable' is different with AVC. In A.3, a list identifies in detail the elements of the stream that can be ciphered without disrupting a standard decoding process.

6.4 Content sensitive encryption for region encryption

6.4.1 General

This subclause describes how to manage region encryption inside the encoder process with some coding restrictions in order to prevent the propagation of encryption errors outside the ROI in inter-coding configuration (i.e. for P-frames and B-frames).

NOTE Even if content sensitive encryption does not directly increase the bitrate of video bitstream, all the constraints (e.g. subdividing in tiles or slices and adding restrictions of coding) presented below tend to have a negative impact on the compression efficiency.

6.4.2 AVC

To protect region in a media, AVC 'slice_groups' (as described in Annex A and ISO/IEC 14496-10:—, G.10) can be used, which define a subset of the macroblocks in a coded picture and may contain one or

more slices. This also permits mapping the sequence of coded MBs to the decoded picture in a number of flexible ways. The allocation of macroblocks is determined by a slice_group_map_type that indicates which slice_group each MB belongs to, in picture parameter set data (see ISO/IEC 14496-10:—, 7.4.2). An example of 2 slice_groups is shown in Figure 5, with map_types 2 (also called 'foreground and background').

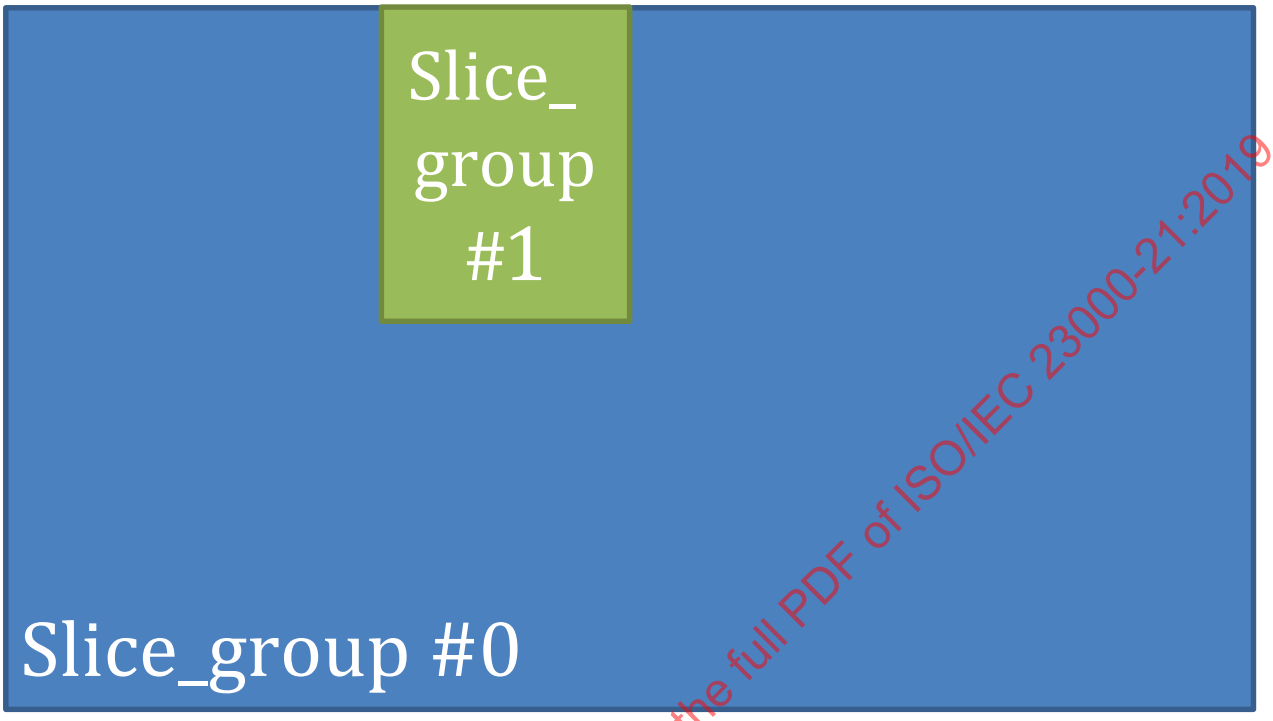

Figure 5 — Frame with 2 slice groups

Here, it can be considered that the foreground (slice_group1) is ciphered by content sensitive encryption and the background (slice_group0) is unencrypted.

Since the slices from each slice_group can be decoded independently, drift errors induced by content sensitive encryption in considered slices are prevented. However, this way (separating frame into independent slices) ensures good reconstruction inside an intra-frame, but not in the following predicted-frames. Motion prediction is to be constrained at the encoder side to prevent drift error propagation from the content sensitive encryption. Thus, to ensure perfect reconstruction over the video sequence in the 'clear' area (i.e. unencrypted area), the codec shall constraint predictions from MB in this 'clear' area, to point exclusively in the previous frames inside another 'clear' area.

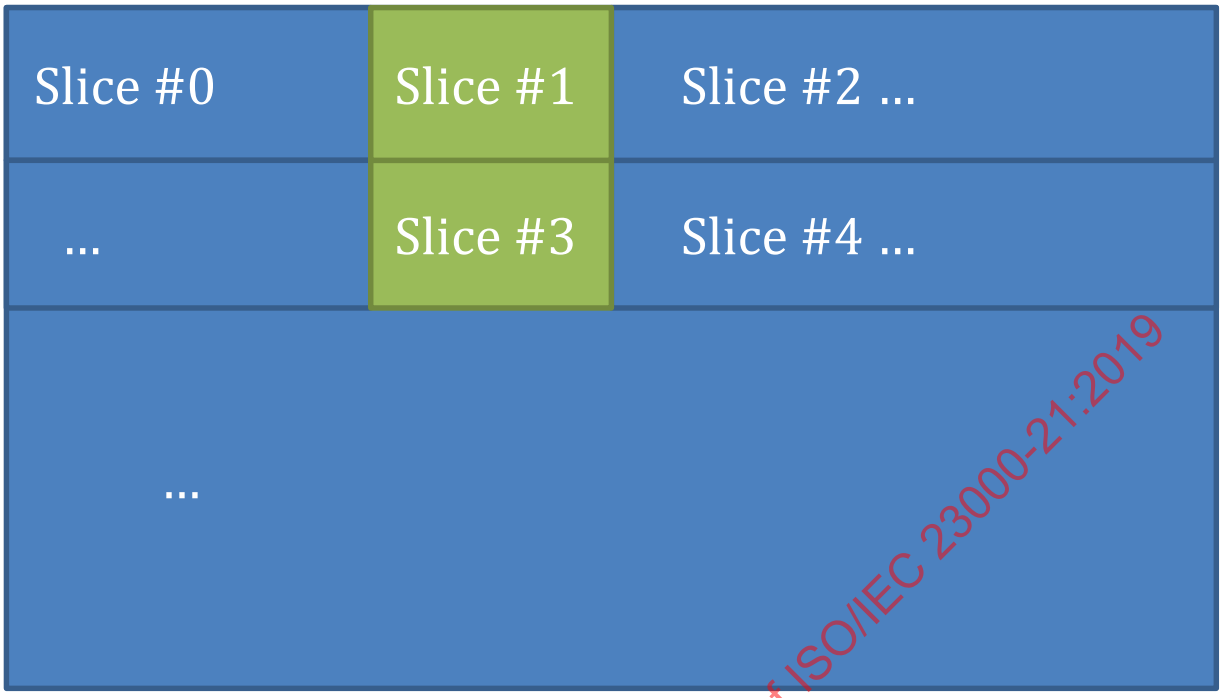

Figure 6 — Frame with several slices

The flexible macroblock ordering (FMO) coding tool (defined in ISO/IEC 14496-10:—, G.10) can also be used, which indicates, through the picture parameter set, to which slice group each MB belongs. But FMO is supported only by baseline and extended profiles, which means that profiles with the best compression efficiency (e.g. high profile) can only use "normal" slice division as shown in Figure 6.

Moreover, it exists in AVC a SEI message called motion-constrained slice group set (see ISO/IEC 14496-10:—, D.2.19) which can fill this need by indicating, for each picture, that the inter-prediction process is constrained as follows: no sample value outside the slice group set, and no sample value at a fractional sample position that is derived using one or more sample values outside the slice group set, is used for inter prediction of any sample within the slice group set.

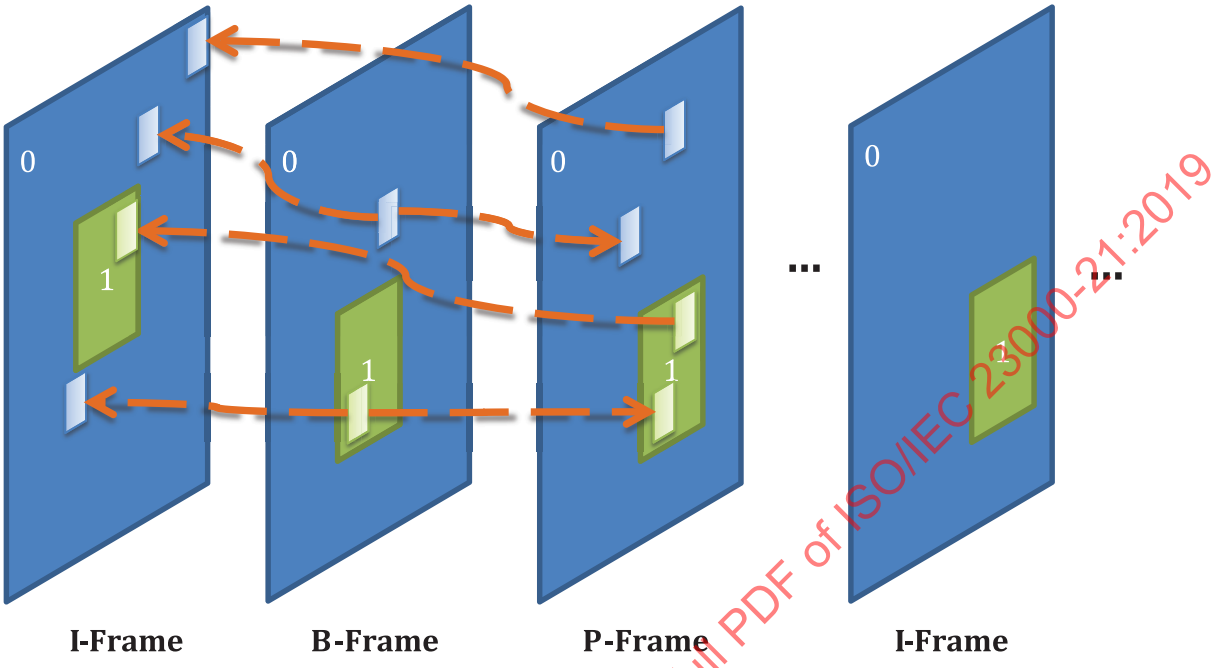

Figure 7 — Example of GOP with frames coded with 2 areas

In Figure 7, only restrict motion compensation of macroblock is imagined from background slice group to only considered sample position from background slice group of previous frames. So the background zone shall be reconstructed correctly even if a part of reference frame is ciphered by content sensitive encryption.

NOTE In this example, the ciphered' area can be predicted from any part of previous frames.

6.4.3 HEVC

The tiles split the video frame into rectangles with integer number of blocks where intra-prediction and the entropy coding dependencies are broken at the tile boundaries. The content sensitive encryption process encrypts only the tiles that contain the ROI whereas the non ROI tiles remain clear (i.e. not encrypted) as shown in Figure 8.

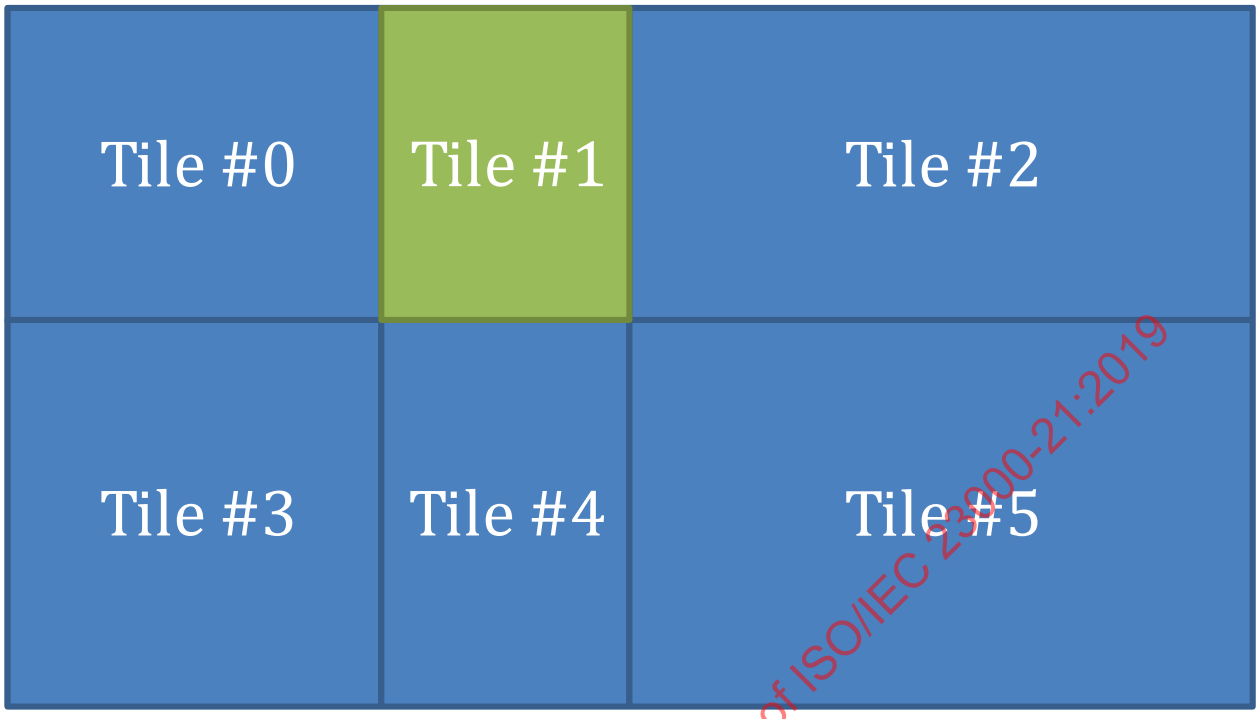

Figure 8 — Frame with 6 tiles

Moreover, as in AVC, motion prediction is to be constrained at the encoder side to prevent drift error propagation from the content sensitive encryption. Thus, to ensure perfect reconstruction over the video sequence in the 'clear' area (i.e. unencrypted area), the codec shall constraint predictions from CTU in this 'clear' area, to point exclusively in the previous frames inside another 'clear' area. This constraint can be indicated by existing SEI message called 'temporal motion-constrained tile sets' for HEVC (see Rec. ITU-T H.265 | ISO/IEC 23008-2:—, D.2.30). In order to prevent the propagation of encryption outside the ROI tile (i.e. the background tiles), some encoding constraints (not normative) are proposed:

1. Disabling the in-loop filters across the tiles boundaries.
2. Restricting the Motion Vectors (MV) in the reference frame to point only under the same tile of the predicted block (as shown in Figure 7).

7 Support for protected streams at system level

7.1 Signalization of protected stream

ISO/IEC 14496-12[1] enables signalling to encrypt a part of the picture (region), but the access unit is indicated by the legacy protection scheme information box ('sinf'), as not decodable without correct decryption (the video cannot be decoded since the encrypted bitstream is not standard format compliant).

ISO/IEC 14496-12 defines the "ScrambleSchemeInfoBox" ('scrb') to address this new property.

The `ScrambleSchemeInfoBox` contains all the information required both to understand the scramble operation applied and its parameters, and also to find other information such as the kind and location of the key management system. The `ScrambleSchemeInfoBox` is a container box.

The `ScrambleSchemeInfoBox` is mandatory in a sample entry for which media samples use a scrambling scheme resulting in media content compliant with the requirements of the media format identified by

the sample entry type of the track, if the track is not a transformed media track, or by the untransformed sample entry type, otherwise.

For an item consisting of media using a scrambling scheme resulting in media content compliant with the requirements of the media format identified by the item type, this property shall be present in the `ItemPropertyContainerBox` and the protected item shall have this property listed as a non-essential property.

This box shall not be used for any scrambling scheme resulting in a cyphered media stream not compatible with the requirements of the media format identified by the sample entry type or item type.

NOTE The `ScrambleSchemeInfoBox` is identical to the `ProtectionSchemeInfoBox` except that (a) it uses the four-character code `'scrb'` and (b) does not contain an `OriginalFormatBox`.

Definition

Box types: 'scrb'

Container: Sample entry or `ItemPropertyContainerBox`

Mandatory: Yes, if the media sample(s) associated with the sample entry or the associated item(s) use a scrambling scheme

Quantity: One or more

The ScrambleSchemeInfoBox contains all the information required to understand the scramble operation applied and its parameters, and also to find other information such as the kind and location of the key management system. The ScrambleSchemeInfoBox is a container box. It is mandatory in a sample entry that uses a code indicating a protected stream for visually scrambling.

Syntax:

```
aligned(8) class ScrambleSchemeInfoBox extends Box('scrb') {
SchemeTypeBox scheme_type_box;
SchemeInformationBox info; // optional
}
```

7.2 Signal of multiple access in protected stream

In ISO/IEC 23001-7, a common encryption ('cenc') protection scheme enables multiple key systems to decrypt the same media content. Each key is identified by a keyID and each encrypted sample is associated with the keyID of the key needed to decrypt it. This association is signalled either through the specification of a default key ID in the track encryption box ('tenc') or by assigning the sample to a sample group, the definition of which specifies a key ID. Files may contain a mixture of encrypted and unencrypted samples. But it is not possible in common encryption design to have different keyIDs for a given sample.

```
aligned(8) class CencSampleEncryptionInformationGroupEntry
extends SampleGroupDescriptionEntry( '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;
      }
   }
}
```

Where:

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

— `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.

A protected sample is usually defined by byte ranges, called subsample encryption information, of clear and encrypted data; this ensures that important media bitstream syntax, such as slice headers, may be left in the clear. A protected sample shall also have an initialization vector associated with it. Both *initialization vectors* and *subsample encryption information* are provided through a single *sample auxiliary information* with aux_info_type equal to the scheme 'cenc' and `aux_info_type_parameter` equal to 0.

However, if different keys are used to protect different parts of the samples, the syntax does not allow for associating different keyIDs to different byte ranges; using the regular design would therefore require a single key to be used by all users, defeating the purpose of this document. It is therefore proposed to use the aux_info_type_parameter to modify the syntax the sample auxiliary information to

provide the needed information (IV and keyIDs per byte ranges rather than per sample). The proposed format of the sample auxiliary information for samples with this type can be as follows:

```
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]
   }
}
```

Where:

— `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.

Figure 9 is a subsample encryption example showing two samples, each containing three subsamples, with two different key IDs applied on each different subsample, with an associated per-sample initialization vector and a logically continuous sequence of 16-byte cipher blocks interspersed with unencrypted byte ranges.

NOTE 1 In each subsample, there can only be one key ID applied.

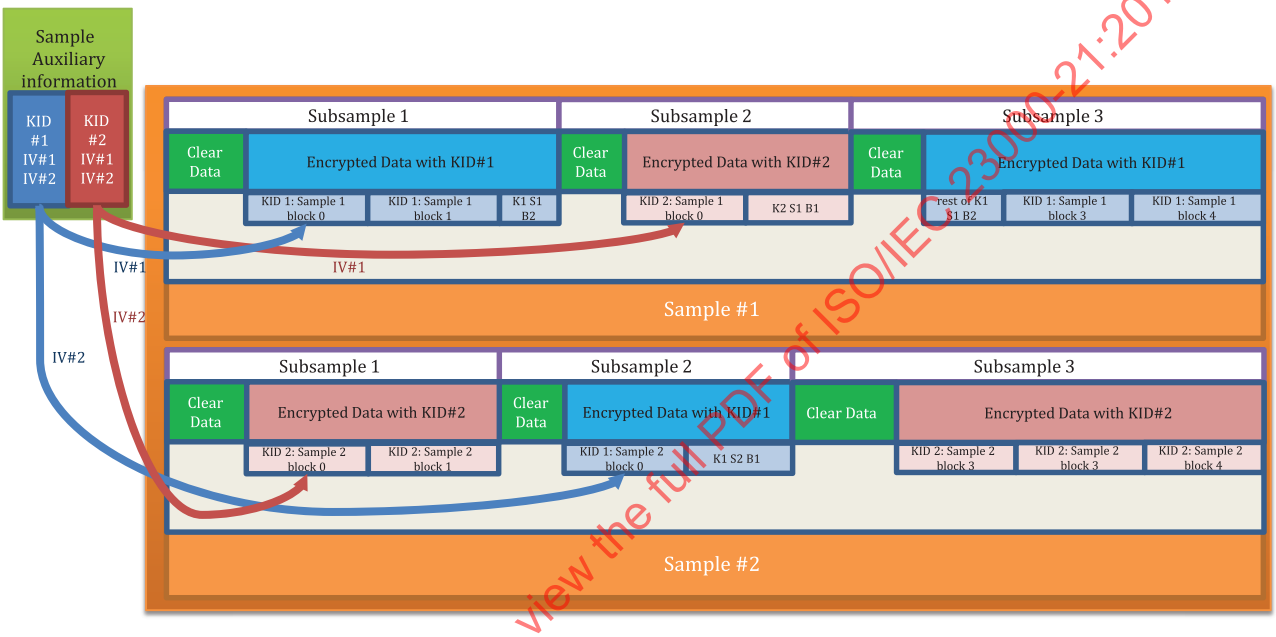

Figure 9 — Subsample encryption with different keys

Figure 10 is an example with several slices dividing a frame compressed with AVC and associated with different Key IDs (and different access control).

NOTE 2 It is possible to reduce the number of slices by using the 'clear data' size and only cipher the last macroblocks of the slices as shown in Figure 11.

Figure 10 — Frame divides in several slices (NALs) and associated to different keys and access control

Figure 11 — Frame divides in several slices (NALs) where only the last MBs are encrypted with different keys and access control

7.3 Signal of content sensitive encryption

7.3.1 Definition of content sensitive encryption

Content sensitive encryption is a bitstream syntax aware scrambling scheme that modifies media bitstream in such a way that an unauthorized receiver (i.e. a standard decoder) can normally decode the ciphered stream, while the displayed content is made non-intelligible by the encryption. This scheme is currently applicable to AVC and HEVC and defined in Annex A. The scheme can be applied to both media tracks and image items.

The scheme operates in compressed domain-based on entropic coder, by identifying the elements of the stream that can be ciphered without disrupting a standard decoding process. The bits to be encrypted are chosen with respect to the considered video standard to ensure full compatibility, achieved by selecting the bits (generally parts of code-words) for which each of the encrypted configuration modifies the decoding process context but does not create desynchronization nor lead to non-compliant bitstream.

The selected elements depend on the video coding specification used:

— A.1 gives the list of elements and possible bit encryption for AVC CAVLC;

— A.2 gives the list of elements and possible bit encryption for AVC CABAC;

— A.3 gives the list of elements and possible bit encryption for HEVC.

NOTE For CABAC entropic coding (i.e. in A.2 and Annex A.3), the bits considering as cipherable regarding to content sensitive encryption process are listed as bins since the considered code words are first binarized before the bypassed coded engine.

7.3.2 Content sensitive encryption applied to a video NAL unit

In content sensitive encryption, 16 bytes block cypher cannot be used directly on the payload. Consequently, a video parser shall be used to locate bits to cipher/decipher. These bits, listed in Annex A, depend on the coding standard and potentially the entropic coding mode used.

A content sensitive encryption scheme shall use the AES-CTR mode for its cipher as defined in ISO 23001-7:2016, 4.2:

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

The encryption and decryption processes are performed with a simple XOR operation between the identified bits in the syntax and the cypher blocks, as shown in Figure 12.

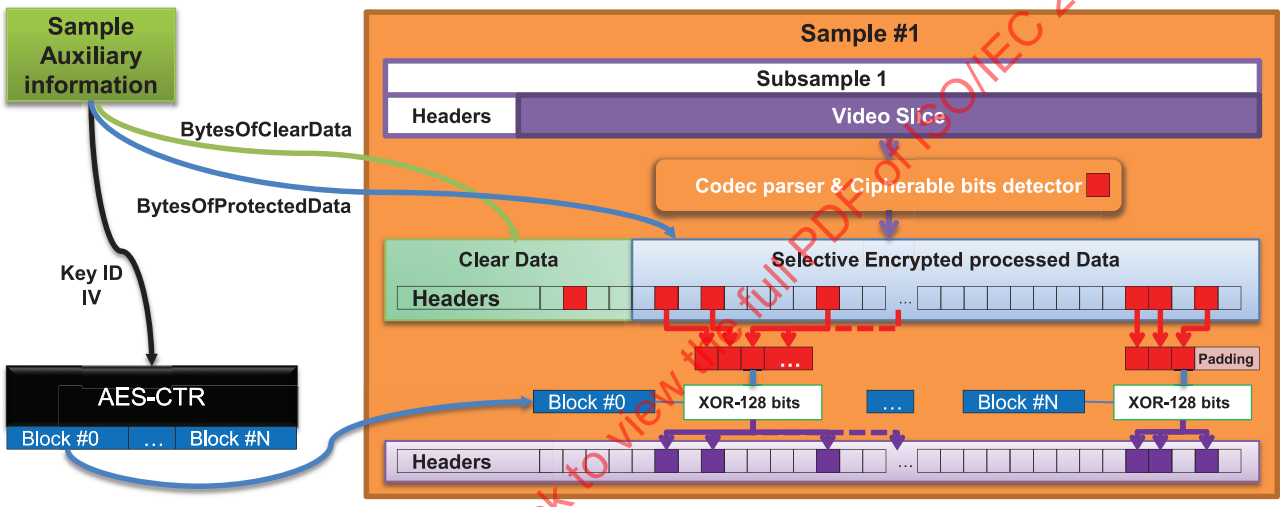

Figure 12 — Content sensitive encryption scheme

Samples protected with content sensitive scheme shall follow subsample encryption of NAL structured video tracks as defined in ISO 23001-7:2016, 9.5.2. The samples shall be divided into one or more contiguous subsamples. Each subsample consists of an unprotected part of `BytesOfClearData`, potentially 0, followed by a protected part. Bits contained in the clear part of the subsample shall not be selected for the decryption process. The first bit selectable for the encryption process in the range of protected data shall be XORed with the first bit of the first encrypted cipher block.

The AES-CTR counter shall be incremented after each completely used encrypted cipher block (128 bits) or at the subsample boundaries; this implies that if not all bits were consumed on the last cipher block of a subsample, the block shall be considered as consumed and the counter shall be incremented before processing any subsequent subsample associated with that cipher.

7.3.3 `'sve1'` AES-CTR sensitive encryption scheme

The `scheme_type` field of the scheme Type Box (`'schm'`) shall be set to the four character code `'sve1'`. Tracks or items carrying media coding types for which no content sensitive encryption mode is defined in Annex A shall not use this protection scheme.

Encrypted video tracks or items using NAL unit structured video conforming to ISO/IEC 14496-15 shall be protected using subsample encryption specified in ISO 23001-7:2016, 9.5, and shall use content

sensitive encryption as specified in ISO 23001-7:2016, 9.7. The NAL unit header should be left in the clear part of the subsample.

NOTE In AVC CAVLC, the slice header can be encrypted but can still be parsed.

Pattern encryption shall not be used. As a result, the fields `crypt_byte_block` and `skip_byte_block` shall be 0 and the version of the Track Encryption Box (`'tenc'`) shall be 0.

Non-video encrypted tracks shall be protected using full-sample encryption as specified in ISO 23001-7:2016, 9.4 and shall use content sensitive encryption as specified in ISO 23001-7:2016, 9.7.

For tracks, the version of the `TrackEncryptionBox ('tenc')` shall be 0. For items, the version of the `ItemEncryptionBox ('ienc')` shall be 0.

Constant IVs shall not be used; `Per_Sample_IV_Size` shall not be 0, except for unencrypted sample groups.

Annex A
(normative)

Content sensitive encryption scheme

A.1 AVC CAVLC: Code words containing bits 'selected for encryption'

A.1.1 General

This clause gives the list of all variable-length coding tables and code words that shall be encrypted by the CSE process for AVC CAVLC mode (i.e. entropy_coding_mode is equal to 0 in Rec. ITU-T H.264 | ISO/IEC 14496-10 in picture parameter set RBSP syntax).

A.1.2 Slice QP Delta

In ISO/IEC 14496-10:—, 7.3.3, **slice_qp_delta** is coded in se(v) (i.e. signed Exponential-Golomb code-word), and the value is given in Table A.1. But the value of slice_qp_delta shall be limited such that $SliceQP_Y$ is in the range of $-QpBdOffset_Y$ to +51, inclusive. So this code-word is eligible to be cyphered if, and only if, its absolute value is less than:

$$2^{Floor\left(Log2\left(Min\left(QpBdOffset_Y+26+pic\_init\_qp\_minus26,\ 25-pic\_init\_qp\_minus26\right)\right)\right)}$$

Then, the suffix bits (i.e. the bits following the first '1'), as illustrated by the bold font in Table A.1, shall be selected for CSE.

Table A.1 — Slice_QP_delta code-words

Index	Slice_QP_Delta value	Code-word
0	0	1
1	1	01**0**
2	−1	01**1**
3	2	001**00**
4	−2	001**01**
5	3	001**10**
6	−3	001**11**
7	4	0001**000**
8	−4	0001**001**
…	….	**…**

A.1.3 Macroblock type

In Rec. ITU-T H.264 | ISO/IEC 14496-10:—, 7.3.5, **mb_type** specifies that the macroblock type is coded in ue(v) (i.e. unsigned Exponential-Golomb code-word) but the semantics of mb_type depend on the slice type.

If slice_type is I, the bits in bold font in Table A.2 shall be selected for CSE. Moreover, blocks located on the border of the video slice shall not be selected for CSE.