

国 际 电 信 联 盟

ITU-R

国际电联无线电通信部门

ITU-R BT.1563-1 建议书

(03/2011)

使用密钥长度数值的数据编码协议

**BT 系列
广播业务
(电视)**



国际电信联盟

前言

无线电通信部门的职责是确保卫星业务等所有无线电通信业务合理、平等、有效、经济地使用无线电频谱，不受频率范围限制地开展研究并在此基础上通过建议书。

无线电通信部门的规则和政策职能由世界或区域无线电通信大会以及无线电通信全会在研究组的支持下履行。

知识产权政策 (IPR)

ITU-R的IPR政策述于ITU-R第1号决议的附件1中所参引的《ITU-T/ITU-R/ISO/IEC的通用专利政策》。专利持有人用于提交专利声明和许可声明的表格可从<http://www.itu.int/ITU-R/go/patents/en>获得，在此处也可获取《ITU-T/ITU-R/ISO/IEC的通用专利政策实施指南》和ITU-R专利信息数据库。

ITU-R 系列建议书

(也可在线查询 <http://www.itu.int/publ/R-REC/en>)

系列	标题
BO	卫星传送
BR	用于制作、存档和播出的录制；电视电影
BS	广播业务（声音）
BT	广播业务（电视）
F	固定业务
M	移动、无线电定位、业余和相关卫星业务
P	无线电波传播
RA	射电天文
RS	遥感系统
S	卫星固定业务
SA	空间应用和气象
SF	卫星固定业务和固定业务系统间的频率共用和协调
SM	频谱管理
SNG	卫星新闻采集
TF	时间信号和频率标准发射
V	词汇和相关问题

说明： 该ITU-R建议书的英文版本根据ITU-R第1号决议详述的程序予以批准。

电子出版
2011年，日内瓦

© ITU 2011

版权所有。未经国际电联书面许可，不得以任何手段复制本出版物的任何部分。

ITU-R BT.1563-1建议书
使用密钥长度数值的数据编码协议
(ITU-R第130/6号课题)

(2002-2011年)

范围

本建议书定义了代表数据项和数据组的字节层面数据编码协议。该协议规定了与所使用的应用或传输方法无关的数据结构。

本建议书将密钥长度数值（KLV）三位字节定义为数据项目或数据组的数据互换协议。在这些数据项目和数据组中，密钥确定数据，密钥长度规定了数据的长度而数值就是数据本身。KLV协议为所有合规应用提供了通用互换点，不论实施或传输方法如何。

国际电信联盟无线电通信全会，

考虑到

- a) 很多国家基于对符合ITU-R BT.601、ITU-R BT.656和ITU-R BT.799建议书的数字视频组件的使用安装了数字电视制作设施；
- b) 高清电视（HDTV）制作系统已根据符合ITU-R BT.1120建议书的数字HDTV接口获得安装；
- c) 符合ITU-R BT.656或ITU-R BT.799建议书的信号内存在用串行数据流中复用的附加数据信号容量；
- d) 对串行数据流辅助数据信号复用将产生运行和经济收益；
- e) 如辅助数据信号使用的格式种类差异保持最低则加强运行收益；
- f) 辅助数据包格式化的规定见ITU-R BT.1364建议书；
- g) 将辅助数据包作为一种传输形式的多种数据的通用格式化将有助于广播传输操作，

建议

- 1 附件1规定的密钥长度数值（KLV）数据格式化 –（使用密钥长度数值的数据编码协议）作为串行数字接口中多种数据使用的方法；
- 2 是否遵守该建议书属自愿行为。然而，该建议书可能包含一些强制性规定（以确保，如互操作性或可执行性）。当满足了所有强制性规定时，就已符合该建议书。“应”或一些其它强制性用语，如“必须”以及以否定形式表达相同含义的词语用来表述要求。这些措辞的使用不应表示部分或全部地符合本建议书。

附件1

1 KLV协议¹

表1和图1是对KLV编码数据协议的介绍。所编代码可能是单一数据项，或是一个数据组。

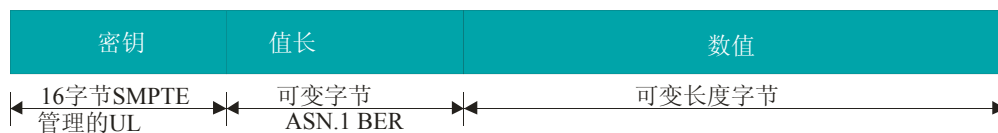
KLV编码协议由通用标签（UL）标识“密钥”、之后的以数字表述的“长度”（值长）和之后的数据“值”组成。

全密钥的长度应为16字节。该值是相关建议书规定的数据类型的数据类型的字节序列，无需经KLV协议再次编码。值字段长度不同，所有相关限制规定在相关定义建议书中。

表 1
数据编码的KLV字段

字段	描述	长度	内容/格式
密钥	用于识别值的UL	16字节	第1.1节
长度	值字段的长度	定义在相关注册表、基本要素、应用标准中，但长度不一。	第1.2节
值	与密钥相关的值	可变	第1.3节

图 1
KLV编码



BT.1563-01

1.1 通用标签密钥

按照SMPTE 298M，KLV编码协议应使用并仅可以使用固定的16字节SMPTE管理的通用标签，作为识别值字段中数据的密钥。术语UL在本建议书通篇指SMPTE管理的通用标签（见附录2）。

整个密钥应包含一个16字节字段，其中包括一个对象身份（0x06）和UL大小（0x0E，表明密钥总长为16字节），之后有以UL代码（0x2B）和SMPTE标号（0x34）开始的标识系列。标识按表2规定确定了UL标号（字节3~8）和项目标号（字节9~16），。

¹ SMPTE是有关通用标签、类型值和元数据字典中条目的注册表。本建议书的读者需在以下网站 <http://www.smpte-ra.org/> 核对最新条目的注册值。

表2

KLV数据编码的密钥字段描述

编号	字段	描述	长度	内容/格式
	UL字头			
1	OID	对象标识	1字节	永远是0x06
2	UL大小	UL的大小为16字节	1字节	永远是0x0E
	UL标号			
3	UL代码	级联分标识ISO、ORG	1字节	永远是0x2B
4	SMPTE标号	SMPTE分标识	1字节	永远是0x34
5	类别标号	确定所述注册类别（如字典）的类别标号	1字节	见表3
6	注册标号	确定类别中具体注册表（如元数据字典）的注册标号	1字节	见表3
7	结构指定方	给定注册标号中不同结构的标号	1字节	第4.1.3节
8	版本编号	最早按照项目标号确定项目的给定注册表版本	1字节	递增编号
9至16	项目标号	在UL标号范围内某些项目的独特识别	8字节	见相关建议书和版本

在SMPTE标号后的前两个分标识应按照本建议书为KLV编码协议预留数值。

SMPTE 298M UL中的各项词汇使用ISO/IEC 8825-1规定的有关对象标识编码的ASN.1基本编码规则（BER）编码。

UL标号各字节数值应限制在0x01至0x7F范围内，在BER对象标识符编码中由一个字节表示。

项目标号数值使用有关对象标识的ASN.1基本编码规则(BER)编码，长度应为8字节。

UL标号和项目标号中的分标识的重要性从左至右递减，最左边的分标识最重要。密钥中最左边分标识的数值为0x00，它应确定标签的终止，所有意义较轻的分标识也应将值设为0x00。数值为0x00的分标识对密钥的含义没有影响。

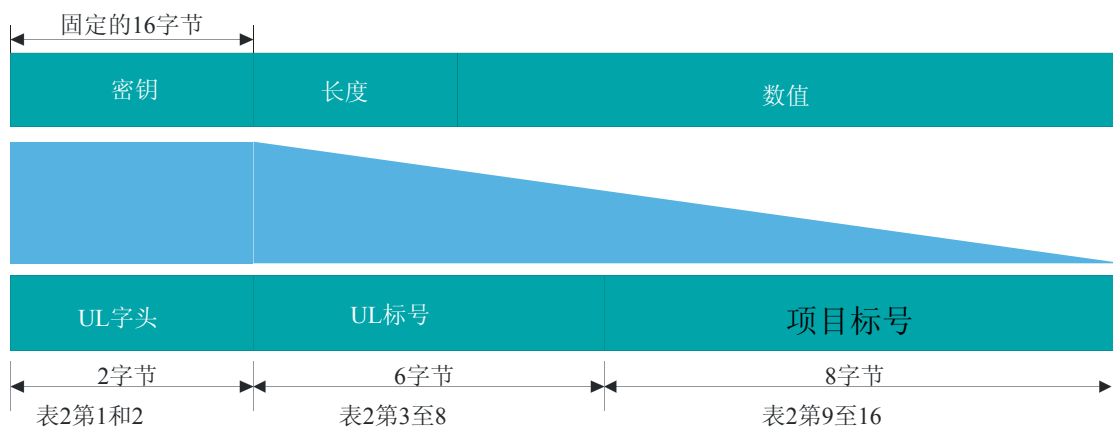
SMPTE 298M仅定义UL前四个字节：对象ID、UL大小、UL代码和SMPTE标号。本建议书规定了用于密钥长度数值编码的SMPTE 298M UL的应用，同时规定了UL标号其余分标识字节5至8）的语义。项目标号（字节9至16）的语义是由一些不同文件定义的，所有这些文件共同探讨了UL标号的确定值。

辨别密钥的解码器不想或不能对相关数值予以解码，这样可忽视该项目并应继续使用密钥长度数值对其余项目进行解码，以便“跳过”未解码项目值。如解码器存储转发该项目，应在不对项目做出任何修改的情况下予以转发。

密钥的字节5和6应用来识别所有给定类别内项目标号各值的内容以及注册标号的内容。表3定义了字节5和6的使用。当字节5和6与表3中数值不符时，语法分析程序则不对“V”字节的内容予以解释。并应将K L和V提供给应用处理，同时继续对在“V”之后的字节进行语法分析。

注 1 – 应用编写器应认识到公共和专用SMPTE UL编号空间的注册表的存在。这些注册表将包含有效的KLV密钥，这些KLV密钥是语法分析程序不知道的。对未识别的KLV密钥提供应用层面解释对于互操作性至关重要。

图 2
密钥结构



BT.1563-02

1.1.1 UL标号

表3定义了UL标号字节5至7使用的标号字节值。SMPTE建议书和建议做法（RP）确定了字节5（注册类别标号）值在0x01至0x04范围内的密钥并将整个密钥注册在注册表中，由字节6和7（注册标号和结构标号）确定SMPTE注册机构。

1.1.1.1 字典

SMPTE标准和RP将密钥字5的值确定为0x01，这些标准为字典建议，应用于确定具有KLV数据结构的单一数据项目。

1.1.1.2 组（集和包）

将密钥字5值定义为0x02的SMPTE标准和RP为集和包建议，应用于确定KLV编码的数据项目组。

表3
字节5至7的UL标号

类别 标号	注册标号	定义出处	结构标号	外部参考 (供参考)
字节5	字节6		字节7	
0x01 – 字典		第5节		
	01 – 元数据字典	第5.1.1节	0x01~0x7F	0x01: SMPTE 335M
	02 – 基本要素字典	第5.1.2节	0x01~0x7F	
	03 – 控制字典	第5.1.3节	0x01~0x7F	
	04 – 类型字典	第5.1.4节	0x01~0x7F	0x01: 类型草案, CD2003
0x02 – 组 (集和包)		第6节		SMPTE 395M
	01 – 通用集	第6.1节, 表4	0x01~0x7F	
	02 (默认) – 全集	第6.2节, 表6	第6.2节, 表5	
	03 (默认) – 本地集	第6.3节, 表8	0x01~0x7F	
	04 (默认) – 可变长度包	第6.4节, 表10	0x01~0x7F	
	05 – 长度确定包	第6.5节, 表11	0x01~0x7F	
	06 – 预定	第6.6节	0x01~0x7F	
03 – 包装和容器		第7节		
	01 – 简单包装和容器	第7.1节	0x01~0x7F	
	02 – 复杂包装和容器	第7.2节	0x01~0x7F	
04 – 标签		第8节		
	标签注册	第8节	0x01~0x7F	0x01: SMPTE 400M
05 – 注册的专用信息		第9节		RP225
06 – 7E – 预留				

1.1.1.3 包装和容器

将密钥的字5值定义为0x03的SMPTE标准和RP为包装和容器建议, 使用该密钥确定包装和容器及其内容。术语“包装”和“容器”的定义见附录A。

1.1.1.4 标签

标签编码见第5节。

1.1.1.5 注册的专用信息

注册的专用信息编码定义见第6节。

1.1.1.6 预留

预留注册类别是为本建议书未来扩展或其它SMPTE标准确定的规定保留的。没有其它规范使用这些预留给值。

1.1.2 注册标号

如表2和表3所示，字节6应定义确定一类别（如元数据字典）内具体注册的注册标号。全集、本地集和可变长度包使用多个值确定长度字段的长度。本地集以及本地标签字段的情况相同。

注册标号值的使用定义见表3。

1.1.3 结构标号

如表2和表3所示，字节7应定义给定注册的结构标号。

结构标号值的分配旨在区分同一注册的不可兼容版本。它们可被看作为主要版本编号。

有关结构标号值的使用定义见表3。

1.1.4 版本编号

由SMPTE注册的项目，字节8应确定最早按照项目标号确定项目的给定注册的版本号。

对于SMPTE未注册的项目，负责项目注册的实体应定义其自身版本号政策。

注1 – SMPTE将分配确定非SMPTE组织的第一个节点。该节点将获得一个SMPTE版本号。在SMPTE指定节点之下的版本号将由适当的非SMPTE组织而不是SMPTE分配。

注册表在控制建议书、标准或RP最初获得批准后可增加新的项目。每次增加一套项目定义时，目前的注册表版本号将得到递增。注册中各条目包括有关项目最初定义的版本号。该编号放在字节8中。

语法分析可能会忽视版本号或将版本号作为密钥语法分析过程中的附加指导或一致性检查。

1.1.5 项目标号

密钥字节9至16包含项目标号。

项目标号字段应确定为8字节长度。项目标号值从1至8字节不等。放在右侧，使用零字节填充8字节字段，同时使用第4.1字所述ASN.1 BER对象标识进行编码。

项目标号的准确含义和结构取决于具体的注册表和结构标号值，详情见以下各节。

1.2 KLV长度字段编码

在KLV编码协议中，长度字段值将使用基本编码规则（BER）进行编码或是ISO/IEC 8825-1第8.1.3、8.1.3.3至8.1.3.5段（见附件K）规定的短模式或长模式长度字节编码。这种长度字节的编码方法是独立的，可以对KLV编码数据进行高效语法分析。当KLV编码协议用于KLV编码单位组时，各单位的长度字节可能采用有关该组编码建议书规定的不同方法（见第3节）。

每个应用建议和RP可酌情定义长度字节的最大字节长度或对长度字节的数值范围做出限制，从而简化解码要求。

注1 – 尽管本建议书中没有对长度字节最大字节数量做出任何限制，大长度字节的出现可用ASN.1 BER长模式长度编码的第一个字节确定。

注2 – 建议对所有小于或等于127 (0x7F)的值字段使用ASN.1 BER的短模式。

在实施中，应尽可能对长度字段实施应用有效数值。然而，在一些运作中，规定值字段的长度可能不现实。这种情况是只输入数据流，其密钥和长度字段是在开始时分配的。在此情况下，长度字段值只能在数据流中止时才能确定。在此，有可能无法回到长度字段输入数值。出现这些情况下，长度字段应设为(0x80)，表明值字段的非确定性长度。任何允许不定义值字段长度的应用文件必须确定定位值字段结尾的其它方法。

注3 – 使用长度值(0x80)是因为作为ASN.1 BER长模式数值，它通常没有含义因为它所指的是零个后续字节。

1.3 数据值的编码

数据值可为单个数据项目或数据组。不论哪种情况，数据均为长度由长度字段值确定的字节串。值字段的最后字节应为数据序列的终止字节。

1.4 空数据项

包括KLV包之间的差距在内的KLV包连续性规范不在本建议书范围内，这些问题由适当的传输层文件予以处理。

然而，如应用需要，可通过插入具体的“空”数据项形成数据序列空当。“空”数据项的使用不具强制性。

“空”数据项是一个KLV编码包，它应定义一个长度值，之后是一个空白值字段。无需解释值字段的数据。

“空”数据项可作为单独项目或在具体集规定的情况下放入集内编码。

应用在收到后可删除或跳过任何或所有“空”数据项，应用可插入“空”数据项，但不得要求其它应用保留这些项。

“空”数据项应在允许元数据字典中定义，也可在其它字典中定义。

注1 – 第4.1.1节为版本编号字节的使用提供了指导。在具体的空数据项中，实施使用了不同的版本值，因此建议解码器忽略版本编号值。

注2 – 空数据项通常称为填充项。

2 单独数据项的KLV编码

单独数据项的KLV编码是简单的密钥、长度和数值应用，如第4节所示。

单独数据项的密钥用注册表和一系列长度范围以及数值本身的规范确定。对于单个数据项，密钥字节5的值应为0x01。

2.1 单独数据项注册表

为管理数据，KLV编码的单独数据项收集在注册表中。

本建议书定义了表3确定的四种不同注册表。这些注册表为“字典”，它们是以以下密钥字节6的值确定的：

2.1.1 元数据字典

元数据字典是使用字节6确定的，数值为0x01。元数据字典是元数据项的注册表。

元数据提供的信息不同于基本要素，它本身不具备独立数值，但与基本要素相关（它具有上下相关的特点在其与相关基本要素之间的关系外没有其它含义）。元数据包括：URL、URI、时间编码、MPEG-2 PCR、文件名、程序标签、版权信息、版本控制、水印、有条件接入密钥等。

2.1.2 基本要素字典

基本要素字典由字节6确定，其数值为0x02。基本要素字典是基本要素项目的注册表。

基本要素是代表图像、声音和文本的数据。基本要素类型包括视频、音频和各类数据，其中包括字幕、图表、静止图像、文本、增强和各应用需要的其它数据。

2.1.3 控制字典

控制字典应由字节6确定，其数值为0x03。控制字典应为控制数据项的注册表。

2.1.4 类型字典

类型字典应由字节6确定，其数值为0x04。类型字典应为各类数据类型的注册表。

很多注册表值具有代表多种数据的通用定义集。为简化注册表定义，“类型”字典应用来确定这些数据表示法。类型字典应用作所有其它字典的共用资源。

2.2 值数据表示法的标识

很多数据项的值可用一种以上方式表示。举例而言，元数据字典的开始时间可用由时间编码构成的字符串或比特包形式表示。第一个数据项是对显示的直接映射，而第二个数据项提供高传输效率以便用于窄带数据信道。具有多重数据表示法的元数据字典项目很多。

当一数据项具有多种数值表示法时，一种表示法应被指定为默认表示法，同时被分配一个至少具有一个追加零字节的密钥。其它表示法应通过使用非零值取代最左边追加零分配密钥。分配是顺序进行的。每个表示法应在注册表中作为单独条目录录。

示范示例

- 01.02.03.04.00.00.00.00是“名称”（使用16位 Unicode字符的默认数据表示法）。
- 01.02.03.04.01.00.00.00是“名称”（使用ISO 7位字符的不同数据表示法）。
- 01.02.03.04.02.00.00.00是“名称”（使用UTF-8 Unicode字符的另一种数据表示法）。

语法分析器将所有表示法看作相同的数据项目，即先识别01.02.03.04.00，然后在“00”的位置上寻找xx以便确定不同的编码。由于默认表示法是确定的，第5个位置上的附加非零项目是默认注册项目的新数据表示法。应注意，不同表示法可能对数据项目可使用的值造成限制。

3 KLV组编码

数据项目的组编码可用来减少通用集中各单位密钥中出现的重复信息开销。组编码还能将单独的数据项或项目组进行逻辑分组，从而一起编码并为提高比特效率提供方案。为提高编码效率，K-L-V编码协议可用来支持以下通用集、全集、本地集、可变长度包和确定长度包：

- 通用集应用来搭建数据项目和其它KLV编码项目的逻辑组。通用集自始至终使用完整的KLV编码构造。
- 全集是按照通用集确定的但通过共享共用密钥字头提高编码效率。该编码收益无损耗，每个密钥都可通过全集数据得到全面恢复。
- 本地集是按照通用集确定的，但通过使用短的本地标签提高编码效率。这些标签的含义是在本地集的上下文内确定的。本地集保留KLV的数据构造，但需要单独的建议书或RP，以定义本地标签的含义，同时提供本地标签值向密钥值的映射。
- 可变长度包被定义为数据项的再组合，取消了组内各项目密钥和本地标签的使用。因此，可变长度包依赖于定义了包内数据项目顺序以及包内各项目的UL的建议书或RP。
- 确定长度包是数据项目最有效（但最不灵活）的分组，取消了对密钥和本地标签的使用，也取消了组内各项的长度。因此，确定长度包依赖于定义数据项目、包中各数据项目长度以及包内各项目UL的建议书或RP。

组编码应仅用于本建议书所述的集和包编码。

集和包应包括由KLV集或包数据构造编码为一组的单独数据项目数量。集或包应通过完整密钥定义，其数值应注册在SMPTE注册机构中。

集和包应对本身就是集和包的数据以及单独注册项目编码。这称为KLV递归编码。本建议书没有规定任何应用可使用的递归层面数量。

集或包的出现应通过集或包密钥中注册类别标号字段（字节5）的0x02表示。注册标号字段（字节6）应用来确定集或包的类型。集或包注册应通过结构标号字段（字节7）确定，注册版本应通过版本编号字段（字节8）确定。

集或包值应包括由集或包类型确定编码的单独数据项目数。在包中，项目的顺序和出现是确定的。默认情况下，集内项目的顺序和出现是不确定的。具体的建议作法或建议书可确定任何具体集或集组内数据项目的顺序和出现。

以下各节确定了通用集、全集、本地集、可变长度包和确定长度包数据项目的编码方式。

3.1 通用集

通用集被定义为根据应用或管理原因分组的数据项目数。在通用集中数据项目可采用任何顺序，既可出现，也可不出现。

通用集编码密钥的使用应由包括结构标号在内的相关建议书或RP以及包括版本编号在内的相应通用集注册表确定。

通用集密钥的长度应为16字节。

通用集的长度应按照ASN.1表示法编码，长短模式按规定进行。

通用集值应为一个KLV编码项目序列，其总长是由长度字段确定的。通用集中的每个数据项目应实施KLV数据编码协议，包括整个密钥值。

相关应用建议书或RP可对通用集值（如项目数量和大小）、所允许的项目序列以及项目是否为强制或任选做出限制规定。

有关通用集密钥的阐述见表4。通用集标号是由通用密钥最后8字节确定的。通用集密钥应规定在相关建议书或RP中，密钥值应按照相关建议书或RP的规定注册在SMPTE注册机构，从而确保密钥值的唯一性。

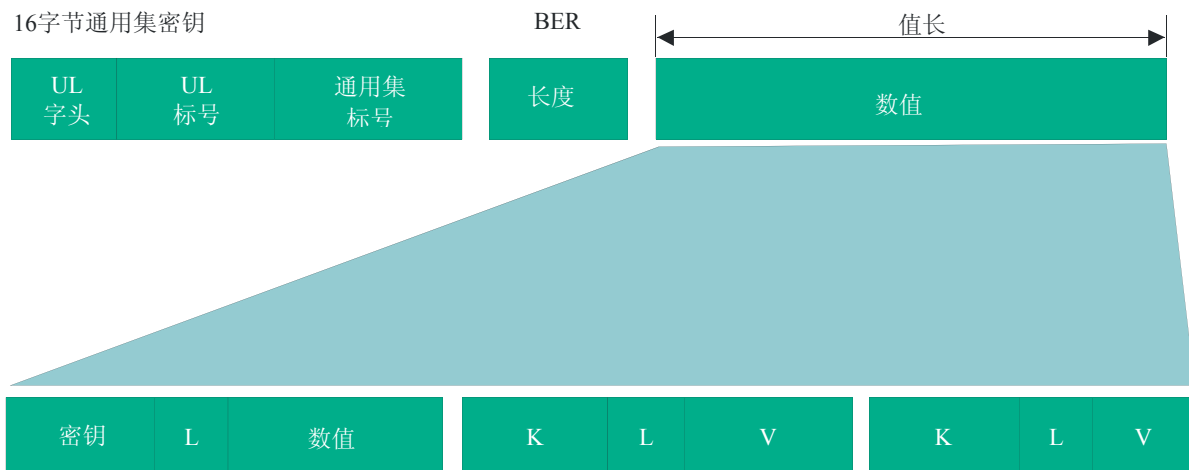
图3显示出通用集编码的数据结构。

表 4
通用集KLV编码密钥的字段描述

编号	字段	描述	长度	内容/格式
	UL字头			
1	OID	对象标识	1字节	永远是 0x06
2	UL大小	UL大小为16字节	1字节	永远是 0x0E
	UL 标号			
3	UL 代码	分标识的级联ISO、ORG	字节	永远是 0x2B
4	SMPTE 标号	SMPTE分标识	1字节	永远是 0x34
5	注册类别 标号	集和包	1字节	永远是 0x02
6	注册标号	通用集	1字节	永远是 0x01
7	结构标号	通用集注册表内不同结构的标号	1字节	由通用集注册表和建议书或 RP确定
8	版本编号	最早按项目标号确定项目的给定注册表版本	1字节	递增编号
9至 16	通用集标号	某个通用集的独特标识	8字节	见通用集注册表

图 3

KLV编码通用集数据结构



数值由使用16字节密钥和BER长度单独编码的KLV项目

3.2 全集

全集被定义为为无损失地缩短集内各项密钥长度而组合的数据项数量。数据项在全集内可采用任何顺序，即可出现，也可不出现。

全集编码密钥的使用应由包括结构标号在内的相关建议书或RP以及包括版本编号在内的相关全集注册表确定。

全集密钥的长度为16字节。

全集的长度应按照ASN.1表示法进行默认编码；长短模式按要求选择。

全集值应为一个KLV编码项序列，其总长由长度字段确定。全集的每个数据项应使用KLV数据编码协议，但取代密钥的全球标准标签值较短，如下文所述。

全集UL应分两个部分定义：

第一组的8字节（UL字头和UL标号）应注册在SMPTE注册机构，用来确定包括结构标号在内的全集建议书或RP。全集注册表中每条目应记录最初定义的版本编号。

第二组的8字节被称为全集标号，用来确定通用UL字头和全集内各密钥的UL标号。第二组的8字节应包括UL字头字段以及尽可能多的UL标号，因为全集中各项目的标号相同并通过结构标号（字节7）表示。全集标号可使用零值字节终止，以表示通用UL标号根的结合。零值终止符的第二组长度为2至8字节。如第二组的长度为8字节，则不需要零值终止符字节。

各全球标准标签的长度为2至12词。少于12词的长度全球标准标签使用单一零值终止，由此取消备用UL数据。

全集中各数据项的16字节密钥可通过将全集标号和各项目全球标准标签的排列进行毫无损失的再创建。如所产生的级联在长度上小于16字节，16字节空间的其余字节将用零填充。

相关应用建议书或RP可对全集值，如项目数量和大小，允许的项目序列以及项目是否为强制性或可选性施加限制。

表5描述了全集的密钥。全集标号在全集密钥中定义在最后8个字节内。全集密钥将定义在相关建议书或RP中，密钥值将按照相关建议书或RP注册在SMPTE注册机构，以确保密钥值的唯一性。

表 5
全集编码密钥的字段描述

编号	字段	描述	长度	内容/格式
	UL字头			
1	OID	对象标识	1字节	永远是0x06
2	UL大小	UL大小为16字节	1字节	永远是0x0E
	UL标号		1字节	
3	UL代码	分标识的级联ISO、ORG	1字节	永远是0x2B
4	SMPTE 标号	SMPTE分标识	1字节	永远是0x34
5	注册类别 标号	集和包	1字节	永远是0x02
6	注册标号	全集	1字节	见表6
7	结构标号	全集注册内不同结构的指定	1字节	见本表最后的注解
8	版本编号	最早按全集标号确定项目的全集注册表版本	1字节	递增编号
9至16	全集标号	所有全球标准标签共用密钥的通用部分	8字节	编号确定了建立所有全球标签标签通用根需要的字节（2至8字节）

注 1 – 该值等于1加通用部分开始前从本表考取的有关键的最初字节数量。数值1（复制0字节）至9（复制8字节）为允许数值。数值5（复制4字节）最合理。

图4显示出全数据集编码结构。

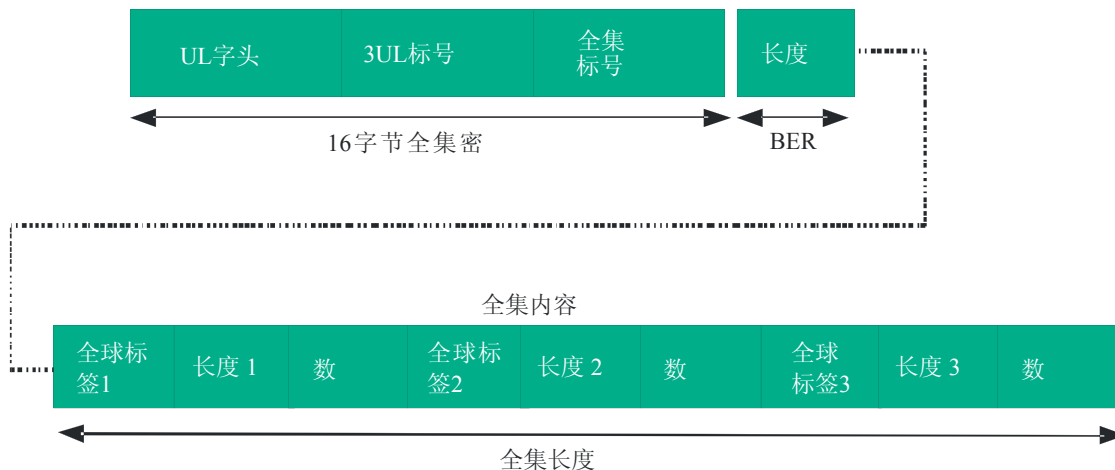
16字节全集密钥之后使全集长度（使用ASN.1 BER长度编码方式编码），在之后为数据项数量，由全球标准标签、长度和数值组成。

各数据项长度字段的默认规范为ASN.1 BER长短模式编码。当使用非ASN.1 BER编码时，字节顺序从达到小。允许长度字段长度的全部范围由注册标号按照表6确定。全集中的所有长度字段应采用同样的句法。

全集可容纳递归，因此与全球标准标签关联的密钥可辨别来自注册表中的单一数据项、还是来自集或包建议书或RP以及相应注册中的数据集或包。

图 4

KLV编码全集数据结构



BT.1563-04

表 6

全集句法注册标号（字节6）的编码

字节6值	长度字段	描述
0x02	ASN.1 BER长或短	任何长度（默认）
0x22	1 字节	最长长度为255
0x42	2 字节	最长长度为65535
0x62	4 字节	最长长度为 $2^{32}-1$

3.3 本地集

本地集被定义为为缩短集内各数据项密钥长度而组合的数据项数量。数据项在本地集内可采用任何顺序，既可出现，也可不出现。

本地集编码密钥的使用应按照包括结构标号在内的相关建议书或RP以及包括版本编号在内的相关本地集注册表确定。

本地集密钥长为16字节。

本地集长度应按照ASN.1 BER长度表示法默认编码，长短模式按要求进行。

本地集值应为一个KLV编码项目序列，总长由长度字段确定。

本地集密钥的描述见表7。本地集标号规定在本地集密钥的最后8个字节中。本地集密钥应被定义在相关建议书、标准或RP中。密钥值按照相关建议书或RP的规定注册在SMPTE注册机构，以确保密钥值的唯一性。

表 7

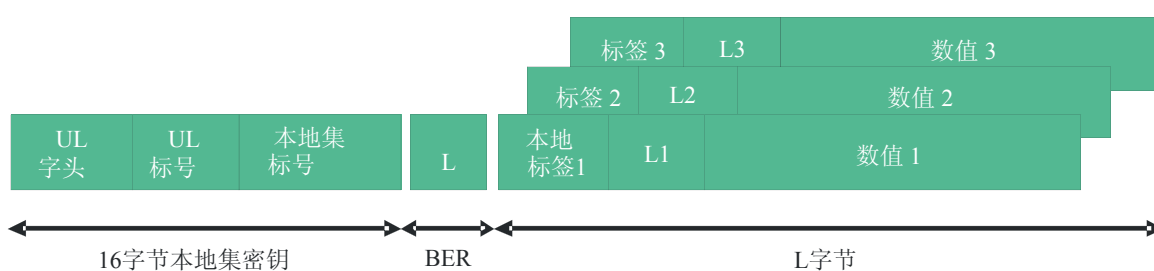
本地集编码密钥的字段描述

编号	字段	描述	长度	内容/格式
	UL字头			
1	OID	对象标识	1字节	永远是0x06
2	UL大小	UL大小为16字节	1字节	永远是0x0E
	UL 标号			
3	UL 代码	分标识的级联ISO、ORG	1字节	永远是0x2B
4	SMPTE 标号	SMPTE分标识	1字节	永远是0x34
5	注册类别 标号	集和包	1字节	永远是0x02
6	注册标号	本地集	1字节	见表8
7	结构标号	本地集注册表内不同结构的标号	1字节	由本地集注册表和建议书或RP确定
8	版本编号	最初按本地集标号确定项目的本地集注册表版本	1字节	递增编号
	本地集标号			
9至16	本地集标号	确定分层机构中本地集的位置	8字节	由本地集注册表和建议书或RP确定

本地集编码的数据结构见图5。

图 5

KLV编码本地集结构



BT.1563-05

16字节本地集密钥之后为集长度，再之后是数据项数量，每个项目由本地标签、长度和值组成。

本地标签字段的合理长度。长度字段的默认规范见ASN.1 BER长短模式编码。当使用非ASN.1 BER编码时，字节顺序从大到小。允许的本地标签和长度字段组合范围见表8确定的注册标号。

表 8

本地集句法注册标号（字节6）编码

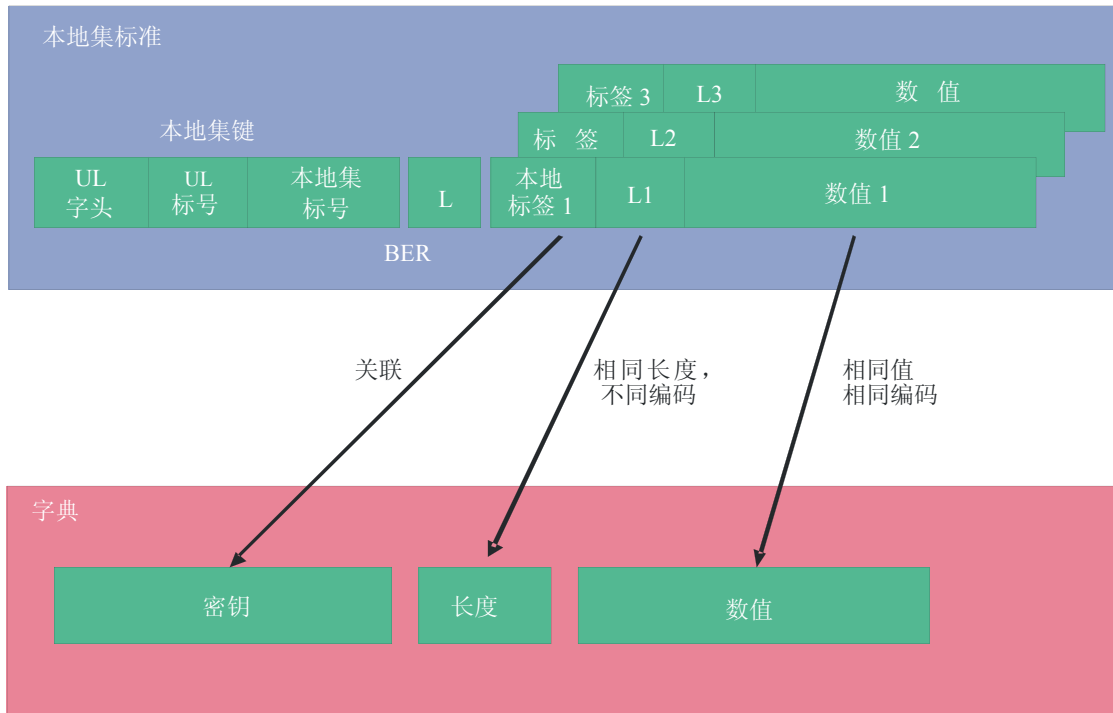
字节6值	长度字段	本地标签字段	描述
0x03	ASN.1 BER 长或短	1字节	任何长度（默认）
0x0B	ASN.1 BER 长或短	ASN.1 OID BER	
0x13	ASN.1 BER 长或短	2字节	
0x1B	ASN.1 BER 长或短	4字节	
0x23	1字节	1字节	最大长度为255
0x2B	1字节	ASN.1 OID BER	
0x33	1字节	2字节	
0x3B	1字节	4字节	
0x43	2字节	1字节	最大长度为65535
0x4B	2字节	ASN.1 OID BER	
0x53	2字节	2字节	
0x5B	2字节	4字节	
0x63	4字节	1字节	最大长度为232-1
0x6B	4字节	ASN.1 OID BER	
0x73	4字节	2字节	
0x7B	4字节	4字节	

注 1 – 相关本地集建议书或RP应定义各数据项本地标签和相应密钥值之间的关系。这种关系应在相关本地集建议书或RP中规定。这些建议书或RP为每个本地标签提供定义项目密钥。这种关联的定义是向本建议书读者获得为提供编码效率而灵活确定化名的机制。相关本地集建议书和RP还应定义本地标签或化名在规范的应用范围。本地集开发人员应提供本地集各标签和定义密钥之间的映射。与通用集和全集不同，通用集和全集的各数据项密钥在集内可毫无损失地重新创建，没有定义建议书或RP以及相应注册表，本地集各标签密钥无法重建。

本地集可容纳递归，使与本地标签关联的密钥辨别来自注册的单一数据项目或来自集或包建议书或RP或相应注册表的数据集或包。

图6显示出本地标签和整个密钥之间的关联。

图6
本地集标签与整个密钥关联显示



BT.1563-06

3.4 可变长度包

可变长度包与本地集相似，但没有本地标签。因此可变长度包中的每个项目仅包含长度字段和值字段。可变长度包中的项目必须按照定义的顺序出现。

可变长度包编码密钥的使用应由包括结构标号在内的相关建议书或RP以及包括版本号在内的相关可变长度包注册表确定。

可变长度包密钥的长度为16字节。

可变长度包长度应在默认情况下按照ASN.1 BER长度表示法进行编码，长短模式按要求进行。

可变长度包值应为一个KLV编码项目序列，其总长由长度字段确定。

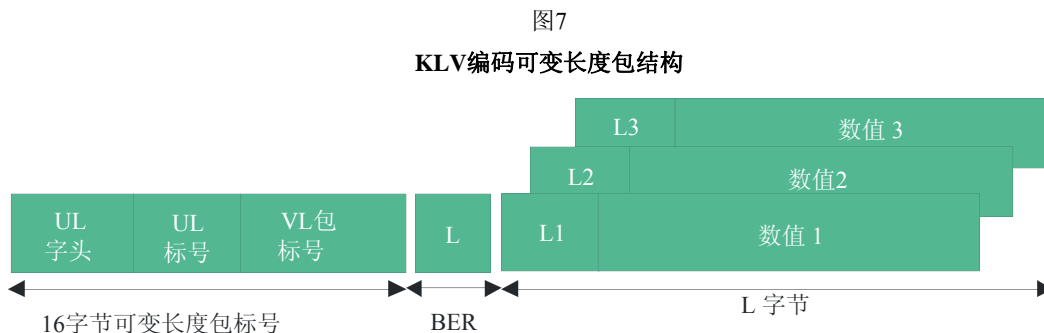
可变长度包密钥的描述见表9。可变长度包标号是在本地集密钥内最后8个字节中确定的。可变长度包密钥应由相关建议书或RP规定。密钥值应按照相关建议书或RP的规定注册在SMPTE注册机构中，以确保密钥值的唯一性。

表9

可变长度包编码密钥的字段描述

编号	字段	描述	长度	内容/格式
	UL字头			
1	OID	对象标识	1字节	永远是0x06
2	UL大小	UL长度为16字节	1字节	永远是0x0E
	UL标号			
3	UL代码	分标识级联ISO、ORG	1字节	永远是0x2B
4	SMPTE标号	SMPTE标号	1字节	永远是0x34
5	注册类别标号	集和包	1字节	永远是0x02
6	注册标号	可变长度包	1字节	见表10
7	结构标号	可变长度包注册中不同结构标号	1字节	由可变长度包注册表和建议书或RP确定
8	版本编号	最早按照可变长度包标号确定的项目的可变长度包注册表版本	1字节	递增编号
	可变长度包标号			
9至16	可变长度包标号	定义了分层结构中可变长度包的位置	8字节	由可变长度包注册表和建议书或RP确定

可变长度包编码的数据结构见图7。



BT.1563-07

16字节可变长度包密钥之后是可变长度包长度（使用ASN.1 BER长度编码进行编码），再之后是项目数量，各项目由长度和数值组成。

各项目长度字段的默认规范为ASN.1 BER长短编码规范。当使用非BER编码时，字节顺序从大到小。所允许的长度字段全部范围由注册标号按照表10定义。

表10
可变长度包句法注册标号（字节6）的编码

字节6值	长度字段	描述
0x04	ASN.1 BER 长或短	任何长度（默认）
0x24	1字节	最长长度为255
0x44	2字节	最长长度为65535
0x64	4字节	最长长度为 $2^{32}-1$

由于包内各项没有本地标签，各项顺序由定义建议书、标准或RP确定。

相关可变长度包建议书或RP将确定各数据项和相关密钥值之间的关系，提供可变长度包各项的密钥。关联定义是使本建议书使用者得以灵活确定自己实现高效编码的化名的机制。可变长度包开发人员应注册可变长度包内各项和定义密钥之间的映射。通用集和全集各项密钥可以毫无损失的重建，与此不同的是，没有定义建议书或RP及相应注册表，可变长度包中的各项密钥无法重建。

可变长度包可容纳递归，使与项目关联的密钥可以辨别来自注册的单一数据项或来自集或包建议书或RP以及相应注册表的数据组。

3.5 确定长度

注1 – 术语确定长度包在本版建议书内已被确定长度包取代。

确定长度包与可变长度包相似，但没有长度字段。因此定义长度包中各项仅包括值字段。确定长度包中各项应按规定顺序出现。

确定长度包编码密钥的使用应由包括结构标号在内的相关建议书或RP以及包括版本号在内的相关确定长度包注册表确定。

确定长度包密钥的长度为16字节。

确定长度包的长度按照ASN.1 BER长度表示法进行编码，长短模式按要求进行。

确定长度包值是一个项目序列，其总长是由长度字段确定的。

在确定长度包中各个项目按照确定顺序出现，没项目具有确定的长度。包内各项的长度值需要分析项目予以确定。因此产生了总体长度确定但可变化的包。本建议书中对确定长度包没有固定、恒定长度值的要求。

确定长度包密钥的描述见表11。确定长度包标号是由本地集密钥最后8个字节确定的。确定长度包密钥通过相关建议书或RP定义，密钥数值应按照相关建议书或RP的规定注册在SMPTE注册机构以保证密钥值的唯一性。

确定长度包编码的数据结构如图8所示。

表 11
确定长度8编码密钥的字段描述

编号	字段	描述	长度	内容/格式
	UL字头			
1	OID	对象标识	1字节	永远是0x06
2	UL大小	UL长度为16字节	1字节	永远是0x0E
	UL标号			
3	UL代码	分标识级联ISO、ORG	1字节	永远是0x2B
4	SMPTE标号	SMPTE标号	1字节	永远是0x34
5	注册类别标号	集和包	1字节	永远是0x02
6	注册标号	确定长度包	1字节	永远是0x05
7	结构标号	确定长度包注册中不同结构标号	1字节	递增编号
8	版本编号	最早按确定长度包标号定义的项目确定的确定长度包注册表版本	1字节	递增编号
	确定长度包标号			
9至16	确定长度包标号	确定长度包在分层结构中的位置	8字节	由确定长度包注册表和建议书或RP确定

图 8
KLV编码确定长度包结构



由于确定长度包内项目没有本地标签，这些项目的顺序由定义建议书或RP规定。

相关确定长度包建议书或RP应定义各数据项目和相关密钥值之间的关联，提供确定项目的密钥。这种关联定义是使本建议书使用者获得定义其自己化名以提高编码效率的机制。确定长度包开发人员应注册确定长度包内各项目之间的映射并定义密钥。与集内各项目密钥可毫无损失地重建的通用集和全集。没有定义建议书或RP和相关注册表，确定长度包内各项目密钥无法重建。

确定长度密钥可容纳递归，使与项目关联的密钥辨别来自注册表的单一数据项或来自集或包建议书或RP以及相关注册表的数据组。

注 2 – 在很多情况下，组的编码可以采用通用集、本地集、可变长度包和确定长度包的做法，无需改变组内各元数据项值。在各种情况下，对于给定组，只有该组密钥字节6将改变，字节9至16保持不变。

3.6 禁止使用

值为0x06的字节6不得用于KLV编码。

4 包装和容器

包装和容器应使用值为0x03的字节5确定。

包装和容器不同于集和包，因为他们没有必要为整个包装或容器的内容使用整体KLV数据构造。因此，建议包装或容器的各个部分对使用KLV编码协议的数据进行编码，但这些部分可能要与其他技术结合起来。在某些情况下，包装或容器可能在一些应用中（如流接口）使用整体KLV构造，但是在其他应用（如存储容器）中使用另一种技术。在这些情况下，包装或容器不被重新定义为集或包，但保留作为包装或容器的定义，以保持标识的一致性。

简单的包装和容器定义为将所有数据嵌入单一框架，没有外部参考。简单的包装和容器采用值为0x01的字节6确定。

复杂的包装和容器用框架定义，其中，每个数据项可能通过参考而不是嵌入包含在文件中。复杂的包装和容器效率更高，适合于可以方便解决参考的本地环境。复杂的包装和容器采用值为0x02的字节6确定。

每项包装和容器规范定义不在本建议书范围内，见其他文件。

5 SMPTE标签

SMPTE标签采用值为0x04的字节5确定。SMPTE标签不得作为KLV编码中的密钥。SMPTE可作为KLV编码三个组成部分中的值，或放在其他编码结构中。

SMPTE标签应用来确定由SMPTE管理的UL本身传递全部含义的对象。SMPTE标签可用于确定实质编码方案，提供独一无二的参数值标识并确定元数据构造及其他。

在包装和容器中，有时甚至在集内，有必要确定集、包装或容器密钥未确定的内容。这些内容可以通过在集、包装或容器内加入一个作为数据项的SMPTE标签加以确认。有必要定义SMPTE UL结构中高层SMPTE标签的出现，因此使解码器了解该项目是SMPTE标签，而不是KLV编码的三位字节的密钥。SMPTE标签的SMPTE UL字段描述见表12。SMPTE标签示图见图9。

表 12

SMPTE标签SMPTE管理的UL字段描述

编号	字段	描述	长度	内容/格式
	UL字头			
1	OID	对象标识	1字节	永远是0x06
2	UL大小	UL长度为16字节	1字节	永远是0x0E
	UL标号			
3	UL编码	分标识级联ISO、ORG	1字节	永远是0x2B
4	SMPTE标号	SMPTE标号	1字节	永远是0x34
5	注册类别标号	标签	1字节	永远是0x04
6	注册标号	具体标签注册	1字节	递增编号
7	结构标号	标签注册内不同结构标号	1字节	递增编号
8	版本编号	最早按标签标号确定的项目的标签注册表版本	1字节	递增编号
	标签标号			
9至16	标签标号	在分层结构中定义标签位置	8字节	使用标签注册表和建议书或做法

图 9

SMPTE管理的UL用于SMPTE标签



6 注册的专用信息

注册的专用信息使用SMPTE管理的UL集的注册类别字段设为0x05。该类别的目的是提供建议，清楚表明将传递注册在外部机构的信息，这些信息不打算公开注册在SMPTE注册表中，或不打算作为元数据或基本要素。

注册的专用信息不将SMPTE管理的UL定义为密钥或SMPTE标签。

当等级类别标号设为0x05时，参考SMPTE RP 225寻找有关UL字段剩余部分的权威定义。

附录 A

缩略语和首字母缩略语表

在本建议书中，使用了以下术语和定义：

ASN – 虚拟句法表示法（见ISO/IEC 8825-1 (ITU-T X.690)）

基本编码规则（BER） – 在ASN.1.中有关不同构造编码的ISO建议。它包括对象标识编码以及长度字段编码。KLV包的长度字节应符合ISO/IEC 8825-1第8.1.3.4和8.1.3.5段规定的长短编码基本编码规则（BER）。

大端顺序–任何多八位字节（多字节）数据实体中框图内第一或最左边的八位字节（字节）最重要。

字节 – 普遍使用的术语“八位字节”的另一种说法。

CER – 典型编码规则 (见ISO/IEC 8825-1 (ITU-T X.690)).

容器 – 为“包含”不同类型信息提供框架的数据对象的一般性名称。该术语一般用于音频、视频、基本要素数据和元数据构成单一数据对象的多媒体。

控制数据 – 为基本要素数据或元数据提供控制功能的数据项。

数据组 – 数据项的集合

数据项 – 本建议书中的数据实体。请注意，其他文件中也广泛使用“项目”，但含义可能不同。另请注意，数据项在本建议书不是一个组。

数据类型 – （见下文类型定义。）

DER – 独特的编码规则（见ISO/IEC 8825-1 (ITU-T X.690)）。

字典 – 为注册表中的数据项提供语义解释的注册表。

基本要素 – 描述有必要代表单一类型视频、音频或其它独立于编码方法的感知体验的任何数据或符号的抽象术语。SMPTE/EBU “有关将程序项目作为比特流交换的统一建议书任务组” (TFHS) 将此作为视频、音频和/或数据信息。基本要素可为图表、遥测、照片或其它信息。

ISAN – 国际建议书音像编号。

密钥 – 16字节SMPTE管理的用于KLV数据编码的通用标签。

KLV – 密钥长度数值。

元数据 – 一般情况下指“有关数据的数据”或“描述其他数据的数据”。元数据被认为是辅助于基本要素或直接补充基本要素的数据。元数据也是被认为有益于或与基本要素相关的信息。

元数据字典 – 批准的元数据项（包括定义和允许的格式）建议书数据库。

元数据项 – 元数据单位的广义术语。

对象标识(OID) – UL中被确定为UL的第一字节 — 缩写为OID。在十六进制(hex)表示法(0x06)中永远是“06”。

八位字节 – 由8个二进制位组成的数据词。

元编码 – 在ASN.1表示法中，确定长度编码方法适用于简单的编码类型和通过隐含标签从简单类型推导得出的类型。它要求提前了解分标识的长度。

注册表 – 通过注册系统维护的信息存储或数据库。

注册系统 – 记录数据的信息系统。

SMPTE管理的UL – SMPTE按照ANSI/SMPTE 298M管理的UL。所有SMPTE管理的UL长度为16字节。

SMPTE标签 – 自我识别的SMPTE UL（见第7节）。

SMPTE注册机构 – 注册机构保留所有ANSI/SMPTE 298M UL密钥和其它参考数据的使用记录。

类型或数据类型 – 定义数据表示法的信息。

SMPTE UL – SMPTE管理的UL的缩略语。

标签 – 编码格式中一种特殊的标识形式。在全扩展的形式下，标签可能与项目标号相同。

UL – 通用标签，符合ISO/IEC 8824-1的对象标识符（亦见ANSI/SMPTE 298M）。在本建议书中该术语用来表示SMPTE管理的UL。

包装 – 由SMPTE/EBU “将程序材料作为比特流交换的统一建议书任务组” (TFHS) 作为包装视频、音频、数据内容和元数据信息进入共同框架的手段。在此定义中，它与容器定义相同，但包装可能还用于“包装”有关已定义的容器的其它元数据。在此含义中，容器是一个具有音视频信息的多功能盒子，而包装是围绕盒子的包装，包括标签和其他支撑元数据。

附录 B (供参考)

ASN.1 BER长度编码

注 1 – 本建议书使用的“字节”术语与ISO/IEC 8825-1术语“八位字节”为同义词。

以下一节引自ISO/IEC 8825-1:

“8.1.3.3 作为确定形式，长度八位字节由一个或多个八位字节组成应代表内容八位字节中的八位字节数量，按照发送者的选择，既可采用短模式（见8.1.3.4段），也可采用长模式（见8.1.3.5段）。

注 2 – 短模式只在内容八位字节数量少于或等于127时使用。”

“8.1.3.4 在短模式中，长度八位字节由一个八位字节组成，其中比特8为零，比特7至1对内容（数值）八位字节（可能是零）的八位字节数量进行编码，作为不带符号的二进制整数，比特7是最重要的比特。

示例:

$L = 38$ 时可编码为 00100110_2 ”。

“8.1.3.5 在长模式中，长度八位字节由一个开始的八位字节和一个或更多的后续八位字节组成。开始的八位字节应按以下形式编码:

- a) 比特8为一;
- b) 比特7至1应对长度八位字节中的后续八位字节数量进行编码，作为不带符号的二进制整数，比特7是最重要的比特;
- c) 值 11111111_2 不得使用。

注 3 – 这一限制是为未来可能引入的扩展。

第一个后续字节的比特8至1之后为第二个后续字节的比特8至1，然后是再后边字节的比特8至1，包括最后的字节，它对等于值字段中字节数相等的而不带符号的二进制整数进行编码，其中首个后续字节的比特8位为最重要比特。

注 4 – 有时被称为“从大到小的”字节顺序。

示例:

$L = 201$ 的编码为:

字节 1 = 10000001_2 , 字节 2 = 11001001_2 [b7b0].

附录 C (供参考)

ASN.1 BER对象标识值的编码

注 1 – 本建议书使用的“字节”术语与ISO/IEC 8825-1使用的“八位字节”术语为同义词。

下一节引自ISO/IEC 8825-1:

8.19 对象标识值的编码

8.19.1 对象标识值的编码为元编码。

8.19.2 内容八位字节应为（按顺序）分标识编码清单（见8.19.3和8.19.4），级联排列。

每个分标识用八位字节系列（一个或多个）代表。各八位字节比特8表明它是否是系列中的最后一个。最后一个八位字节的比特8为零。每个前一个八位字节的比特8为一。系列中八位字节比特1至7为分标识联合编码。从概念上说，这些比特组通过级联构成不带符号的二进制号码，最重要的比特为第一个八位字节的比特7，最后一个八位字节的比特1为最不重要比特。分标识符应以最少的八位字节编码，即分标识符的主导八位字节不得拥有 80_{16} 的值。

8.19.3 分标识数量（N）应小于编码对象标识值对象标识组件数量。

8.19.4 第一个分标识的数字值来源于编码对象标识值中前两个对象标识组件，使用以下算式：

$$(X*40) + Y$$

其中X是第一个对象标识组件的值，而Y是第二个对象标识组件的值。

注 – 前两个对象标识组件的值包表示，只有三个值是从根节点分配的，实现39个节点后续值的方式是通过 $X=0$ 和 $X=1$ 。

8.19.5 第i个分标识的数字值（ $2 \leq i \leq N$ ）属于第（i+1）个对象标识组件。

示例

对象标识值为：

{join-iso-itu-t 100 3}

与以下相同：

{2 100 3}

第一个分标识符为180时，第二个分标识为3。所产生的编码为：

对象

标识	长度	内容
06_{16}	03_{16}	813403_{16}