ICS 35.040
L71

中华人民共和国国家标准

GB/T 20090.6—YYYY

信息技术 先进音视频编码
第 6 部分：数字媒体版权管理
Information technology - Advanced coding of audio and video
Part 6: Digital Rights Management of Media

（送审稿）
(FCDr1)

2006 年 7 月

××××-××-××发布　　××××-××-××实施

国家质量监督检验检疫总局 发布

目　次

目　次 ..... **I**

前　言 ..... **VI**

引　言 ..... **VII**

0.1 *基本理念* ..... *VII*
0.2 *档次划分* ..... *VII*
0.3 *如何阅读本部分* ..... *VII*
0.4 *相关专利情况说明* ..... *VII*

信息技术　先进音视频编码　第 6 部分：数字媒体版权管理 **9**

1.　范围 **9**

2.　规范性引用文件 **9**

3.　术语和定义 **9**

3.1.　定义 9
3.2.　缩写 11
3.3.　数据类型 11

4.　AVS 数字版权管理概要参考模型与档次划分 **12**

4.1.　AVS 数字版权管理概要参考模型 12
4.2.　档 12

5.　核心档 **13**

5.1.　密码算法 13
5.2.　内容解密单元 14
5.3.　认证单元 14
5.4.　解码输出流加密单元 14
5.5.　带保护的压缩流输出功能 15
5.6.　明文重构单元 15
5.6.1.　内容保护的独立区间 ..... ***错误！未定义书签。***
5.6.2.　内容保护模式规定 ..... ***错误！未定义书签。***
5.6.2.1.　全保护模式（模式 1） 15
5.6.2.2.　视频帧基本流模式（模式 2） 15
5.6.2.3.　帧内 M 序列选择保护模式（模式 3） 15
5.6.2.4.　音频帧增强层保护模式（模式 4） 15
5.6.3.　明文重构 *15*
5.7.　密钥管理 15
5.7.1.　内容密钥 *15*

5.7.1.1. 内容密钥推导算法 16
5.7.2. 内容主控密钥 MK *16*
5.7.3. 认证单元公私钥 *16*
5.8. 可信解码器的安全保护要求 16
6. 网络电视档 17
6.1. 网络电视档数字版权管理参考模型 17
6.2. 系统协议消息 17
6.2.1. 概述 *17*
6.2.1.1. 注册协议 17
6.2.1.2. 许可证获取协议 18
6.2.1.3. 许可证撤销协议 18
6.2.2. 消息定义 *18*
6.2.2.1. 注册管理协议 18
6.2.2.1.1. ClientRegHello 18
6.2.2.1.2. ServerRegHello 18
6.2.2.1.3. RegisterRequest 19
6.2.2.1.4. RegisterResponse 19
6.2.2.2. 许可证获取协议 20
6.2.2.2.1. ClientLicHello 20
6.2.2.2.2. ServerLicHello 20
6.2.2.2.3. LicenseRequest 21
6.2.2.2.4. LicenseResponse 22
6.2.2.3. 许可证撤销协议 22
6.2.2.3.1. LicenseRevokeRequest 22
6.2.2.3.2. LicenseRevokeResponse 23
6.2.2.3.3. LicenseRevokeInstruct 24
6.2.2.3.4. LicenseRevokeConfirm 24
6.2.3. 消息命令描述形式 *25*
6.2.3.1.1. 注册协议 25
6.2.3.1.2. 许可证获取协议 27
6.2.3.1.3. 许可证撤销协议 28
6.2.4. 流程 *31*
6.2.4.1. 注册协议 31
6.2.4.1.1. 许可证获取协议 31
6.2.4.1.2. 许可证撤销协议 32
6.3. 对 AVS DREL 数字权利描述语言的采用及限制 33
6.3.1. 权利接受者 *33*
6.3.2. 权利 *33*
6.3.3. 约束 *34*
6.3.4. 义务 *34*
6.4. 内容打包格式 34
6.4.1. RTP 净载动态保护格式 *34*
6.4.1.1. 支持内容保护的 AVS RTP 净载格式 34

6.4.1.1.1. AVSRTPDRMHeader 35
6.4.1.2. SDP 格式 35
6.4.1.2.1. 举例 （资料性） 36
6.4.1.3. 应用过程 (资料性) 36
6.4.2. 支持版权管理的 AVS 文件格式 36
6.4.2.1. 文件的完整性验证 36
文件格式中的 AVS DRM 扩展信息 37
6.4.2.2. 37
6.4.2.2.1. 文件类型盒 37
6.4.2.2.2. IPMP Control Box 38
6.4.2.2.3. Protection Scheme Info Box 39
6.4.2.2.4. 自由空间盒 41
6.4.2.3. 媒体数据的保护 42
6.4.2.4. 文件的流化 42
7. 广播档 42
7.1. 参考模型 42
7.1.1. 所需技术规范 43
7.2. 对 AVS DREL 数字权利描述语言的采用及限制 43
7.2.1. 权利接受者 43
7.2.2. 权利 43
7.2.3. 约束 43
7.2.4. 义务 43
7.3. 内容打包格式 43
7.3.1. 在 AVS 系统传输流上的控制信息表及保护标记 43
7.3.1.1. AVS 数字媒体版权保护控制信息表 43
7.3.1.1.1. 在传输流上的 AVS DRM 控制信息 44
7.3.1.1.2. 在传输流上的 AVS DRM 控制信息段语法 44
7.3.1.1.3. 语义定义 44
7.3.1.1.4. AVS DRM 控制信息类 45
7.3.1.2. AVS DRM 保护标记 47
7.3.1.2.1. 在传输流中 AVS DRM 保护标记 47
7.3.1.2.2. AVS DRM 描述子 48
7.3.2. 媒体内容保护 49
7.3.2.1. TS 层的保护 49
7.3.2.2. PES 层的保护 49
7.3.3. 广播档与原有 CA 系统的关系 （参考性资料） 50
7.3.3.1. 反向兼容 50
7.3.3.2. 前向兼容 50
7.3.3.3. 共存 50
8. AVS 数字权利描述语言 50
8.1. 概述 50
8.2. AVS 数字权利描述语言信息模型 51
8.3. 权利描述语言模型 51

8.3.1. 定义和结构 51
8.3.2. 标识 52
8.3.3. 属性 52
8.3.4. 元素 52
8.3.4.1. 主体 52
8.3.4.1.1. 权利发布者 52
8.3.4.1.2. 权利接受者 53
8.3.4.1.3. 主体属性 53
8.3.4.2. 权利 53
8.3.4.2.1. 普通权利 53
8.3.4.2.2. 高级权利 54
8.3.4.3. 资源 54
8.3.4.3.1. 资源属性 55
8.3.4.4. 约束 55
8.3.4.4.1. 空间的约束 55
8.3.4.4.2. 范围的约束 55
8.3.4.4.3. 时间的约束 56
8.3.4.4.4. 使用环境的约束 56
8.3.4.5. 义务 56
8.3.4.5.1. 付费方式 56
8.3.4.5.2. 交互性的要求 57
8.3.4.5.3. 对用户的要求 57
8.3.5. AVS-DREL 安全模型 57
8.3.5.1. 资源加密信息描述 57
8.3.5.1.1. 摘要 57
8.3.5.1.2. 密钥信息 57
8.3.5.2. 数字签名 58
8.4. AVS 数字权利描述语言数据字典 58
8.4.1. 数据结构概述 58
8.4.1.1. 元素的基本结构 58
8.4.1.2. 元素属性 58
8.4.1.3. 列表值 59
8.4.1.4. 约束性 59
8.4.2. 数据元素列表 59
8.4.2.1. 许可证 59
8.4.2.2. 主体 59
8.4.2.3. 权利 60
8.4.2.4. 资源 63
8.4.2.5. 约束 64
8.4.2.6. 义务 67
8.4.2.7. 元素基本属性 69
8.4.2.8. 许可证标识 70
附录 A 数字权利描述语言 XML 绑定 71

A.1 名称空间前缀 71
A.2 AVS-DREL SCHEMA 71
A.3 AVS-RDD SCHEMA 76
A.4 AVS-DREL 中定义的元素 80
A.4.1 < characterGroup >属性组 *80*
A.4.2 < license >元素 *80*
A.4.3 < licenseUnit >元素 *82*
A.4.4 < subjectType >元素 *83*
A.4.5 < right >元素 *84*
A.4.6 < resource >元素 *84*
A.4.7 < extends >元素 *86*
A.4.8 < constraint >元素 *87*
A.4.9 < duty >元素 *88*
A.4.10 < container >元素 *89*
A.4.11 < sequence >元素 *90*
A.4.12 < context >元素 *92*
A.4.13 < keyInfo >元素 *92*
A.5 AVS-RDD 中定义的元素： 93
A.5.1 声明为 rightElement 的元素 *93*
A.5.2 声明为 constraintElement 的元素 *94*
A.5.3 声明为 dutyElement 的元素 *97*
A.5.4 声明为 context 的元素 *99*

附录 B AVS-DREL 实践指南 (参考性资料) **101**

B.1 规范一致性描述 101
B.1.1 严格一致性的实现 *101*
B.1.2 一致性的实现 *101*
B.2 应用样例 101
B.2.1 样例一 *101*
B.2.2 样例二 *103*

附录 C M 序列发生器（参考性资料） **106**

前　言

GB/T 20090 在《信息技术　先进音视频编码》的总标题下，包括以下九个部分：

第 1 部分：系统－广播；

第 2 部分：视频；

第 3 部分：音频；

第 4 部分：一致性测试；

第 5 部分：参考软件；

第 6 部分：数字媒体版权管理；

第 7 部分：移动视频；

第 8 部分：IP 网络传输；

第 9 部分：系统－文件格式。

本部分为 GB/T 20090 的第 6 部分。

本部分由中华人民共和国信息产业部提出。

本部分由全国信息技术标准化技术委员会归口。

本部分由信息产业部数字音视频编解码技术标准工作组组织起草。本部分起草单位包括中国科学院计算技术研究所，清华大学，华中科技大学，松下电器研究开发（中国）有限公司，华为技术有限公司，中国科学院微电子研究所，北京大学，UT 斯达康（中国）有限公司等。

本部分主要起草人：

黄铁军，黄仲阳，陈大港，赵黎，林辉，周莉，申省梅，霍龙社，王媛，蓝娟，屈彤，李彬，沙瀛，邓婉婷，冯雯洁，吴砥，刘晓云，邹远志，张江，刘清堂，刘永亮,石国欣，刘绍辉，马涛，殷毅波等。

引　言

GB/T 20090 的技术部分包括系统、视频、音频、版权管理等四个部分，其中系统、视频、音频规定的是解码器应当支持的技术要求，版权管理部分是为了适应数字电视广播、网络流媒体以及数字存储媒体等应用中对数字媒体版权管理的需要而制定的共性基础标准。

0.1　基本理念

制定 GB/T 20090.6 的基本理念是：版权管理标准是贯穿数字媒体整个生命周期的技术基础设施，可鉴别、可信任的解码器（称为可信解码器）是版权管理信任链的关键。因此，GB/T 20090.6 首要任务是确定可信解码器的技术要求，消费终端制造商根据此要求开发的各种媒体消费终端能够在多种数字媒体应用（例如广播应用、交互应用和存储应用）中以可信任的方式获取、消费有权利要求的数字媒体。在此基础上，GB/T 20090.6 根据不同应用的特点，制定专用的消息、协议来支持可信解码器和内容提供、授权认证等前端服务系统之间的版权管理信息传送和交换。这样，GB/T 20090.6 就不仅能够作为多种数字媒体应用系统的共性基础标准，也能够通过可信解码器实现这些应用和系统之间的互操作。

0.2　档次划分

GB/T 20090 的本部分目前包括三个档：核心档、网络电视档、广播档，其中核心档定义可信解码器技术要求，网络电视档和广播档分别定义可信解码器支持这两种应用需要满足的技术要求。同时，定义了各应用档用于描述权利的数字权利描述语言。有可能针对其他应用基于核心档定义新的应用档。

0.3　如何阅读本部分

本部分的编写按照档次划分。核心档是所有应用都必须支持的，建议先阅读。网络电视档和广播档具有应用针对性，可选择阅读。网络电视档和广播档的应用者需要阅读数字权利描述语言。

0.4　相关专利情况说明

本标准的发布机构提请注意如下事实，声明符合本标准时，可以使用涉及 x.y，…，x.y.z 条和附录 x 中有关内容的相关专利。

本标准的发布机构对于专利的范围、有效性和验证资料不提出任何看法。

专利持有人已向本标准的发布机构保证，他愿意同任何申请人在合理和非歧视的条款和条件下，就使用授权许可证进行谈判。这方面，该专利持有人的声明已在本标准的发布机构备案。

在本标准起草过程中，起草组织者数字音视频编解码技术标准工作组根据会员签署同意的工作组章程和有关知识产权规定以及会员在提案、审阅等期间提出的专利披露与许可声明等对标准可能涉及的专利进行了识别。已经确知下表列出的专利权人持有本标准的本部分的专利：

专利持有人	联系地址
中国科学院计算技术研究所	北京市海淀区中关村科学院南路6号（100080）

上述专利权人同意对所持有的本标准的本部分的必要专利在合理和非歧视的条款和条件基础上，通过 AVS 专利池进行许可。由数字音视频编解码技术标准工作组推动成立的 AVS 专利池管理委员会是决定专利池具体许可条款的独立机构。对于专利池中的所有专利，标准实施者可通过专利池管理委

员会认可的授权机构获得许可。有关资料可从数字音视频编解码技术标准工作组秘书处获得，联系方法如下：

联 系 人：黄铁军 （数字音视频编解码技术标准工作组秘书长）

通讯地址：北京 2704 信箱 31 分箱

邮政编码：100080

电子邮件：tjhuang@ict.ac.cn

电　　话：+10-58858303，+10-58858300-303

传　　真：+10-58858301

网　　址：http://www.avs.org.cn

请注意除上述已经识别出的专利外，本标准的某些内容有可能涉及专利。本标准的发布机构不应承担识别这些专利的责任。

信息技术　先进音视频编码　第 6 部分：数字媒体版权管理

1.　范围

GB/T 200090 规定了数字音视频的压缩、解压缩、处理和表示的技术方案，适用于高分辨率和标准分辨率数字电视广播、激光数字存储媒体、互联网宽带流媒体、多媒体通信等应用。

标准的本部分规定的技术要求适用于以上所述应用中的数字媒体版权管理需要。

2.　规范性引用文件

下列文件中的条款通过本部分的引用而成为本部分的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本部分，然而，鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本部分。

GB/T 200090.1—YYYY 信息技术 先进音视频编码 第 1 部分:系统

注：本规范第七章中以下定义的

AVS DRM 控制信息表（ADT）的 PID 值 0x03

AVS DRM 流类型的值 Ox1A （表 2－29）

AVS DRM 描述子的值 41 （表 2-39）需由本部分最终确认。

GB/T 200090.2—2006 信息技术 先进音视频编码 第 2 部分:视频

GB/T 200090.3—YYYY 信息技术 先进音视频编码 第 3 部分:音频

GB/T 200090.7—YYYY 信息技术 先进音视频编码 第 7 部分:移动视频

GB/T 200090.8—YYYY 信息技术 先进音视频编码 第 8 部分:在 IP 网络上传输 AVS

GB/T 200090.9—YYYY 信息技术 先进音视频编码 第 9 部分:文件格式

ISO/IEC 9594-8:2001 信息技术－开放系统互联－目录：公钥和属性证书框架

XML “Extensible Markup Language (XML) 1.0 (Second Edition)”, W3C Recommendation 6 October 2000, URL:http://www.w3c.org/TR/2000/REC-xml-20001006/

XMLSchema “XML Schema Part 2: Datatypes”, W3C Recommendation 2 May 2001, URL:http://www.w3.org/TR/xmlschema-2/

XMLSIG “XML Signature Syntax and Processing”, W3C Recommendation 12 February 2002, URL:http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/

3.　术语和定义

下列术语和定义适用于 GB/T 200090 的本部分。另外，GB/T 200090.1，GB/T 200090.8 和 GB/T 200090.9 中的定义和缩写也适用于本部分。

3.1.　定义

许可证

数字权利描述语言基本单位，即某一个权力发布者对另一权利接受者的权利声明。

许可单元

许可证的基本单元。许可单元针对某种资源向权利接受者发布某种权利，依次包含主体，权利，资源，约束和义务 5 种元素。

主体

针对价值传递链中主体所处的不同位置，主体分为权利发布者和权利接受者。

权利

泛指主体对资源所拥有的操作。

资源

资源包括视频，音频等数字资源。

约束

定义了主体对资源使用相应权利时应满足的条件。

义务

义务是主体在行使一定权利的同时承担的各种责任。

适配

一种解析或转换过程，实现可信解码器和外围系统之间的互联互操作。

密码单元

一些硬件、软件、固件或一些组合的集合，这些组合实现了包括密码算法和密钥产生（可选择）在内的密码功能或过程。

M序列发生器

根据生成多项式和初相位，可以产生长周期的伪随机序列的非线性反馈的移位寄存器。生成多项式决定了反馈移位寄存器的结构；初相位指出反馈移位寄存器从什么状态开始运转。

AVS可信解码器

符合本标准的 AVS 解码器

DRM权利容器

一个包含有受保护的内容相关联的数字权利描述（如使用规则等）信息的基本容器。

DRM控制信息表

一个特别定义的用于展现节目所必要的保护信息表，类似与其他携带于传输流中的 PSI 表(如节目关联表,节目映射表,条件访问表等)。

DRM描述子

所描述的信息标识与被保护的基本流的相关联系。

注册服务器

用于处理客户端的注册身份请求。

许可证服务器

服务器用于管理客户端的许可证的获取和撤销。

3.2. 缩写

AVS: 数字音视频编解码技术标准工作组(Audio and Video Coding Standard Workgroup)
XML：可扩展标记语言 (EXtensible Markup Language)
SDP: 会话描述协议 (Session Description Protocol)
RTP: 实时传输协议 (Realtime Transport Protocol)
URI: 通用资源标志 (Uniform Resource Identifier)
DRM: 数字版权管理 (Digital Rights Managment)
CPF：内容打包格式(Content Packaging Format)
DREL:数字权利描述语言(Digital Rights Expression Language)
RDD: 权利数据字典 (Rights Data Dictionary）

3.3. 数据类型

字节阵列ByteArray

通用的字符串（string）或字节（Bytes）阵列容器。

实型变量Decimal

一个带符号的允许小数点的实型数，在前面没有“－”号时为正。如：“2.0”，“－3.0”和“4.5”。

整型变量Integer

一个整型数。

标识Identifier

不包括空格、逗号和非可视字符的数字－字符串，最大长度 255。

短字符串String255

由 ASCII 码组成的长度小于 255 的字符序列。

长字符串String4096

由 ASCII 码组成的长度小于 4096 的字符序列。

日期时间Datetime

ISO8601 格式中的字符串。

标准时间间隔Timespan

遵从时间日期格式标准[ISO8601]。国际标准记录法中时间记录格式是由小时、分和秒表示的时间段，彼此用冒号分隔，形如 hh:mm:ss。小时由两到四位数字组成；分由两位数字组成，可以是 00 到 59 之间的任意值；秒由两位数字和额外的两个小数位组成，可以是 00 到 59.99 之间的任意值。三个由冒号分隔的数字都是必要的，只显示分和秒的情况也不能例外。

通用资源标志符 URI

符合 RFC2396 格式的字符串。

4. AVS 数字版权管理概要参考模型与档次划分

4.1. AVS 数字版权管理概要参考模型

AVS 数字版权管理的概要参考模型如图 1，自内而外包括可信解码器、适配层和外围环境。

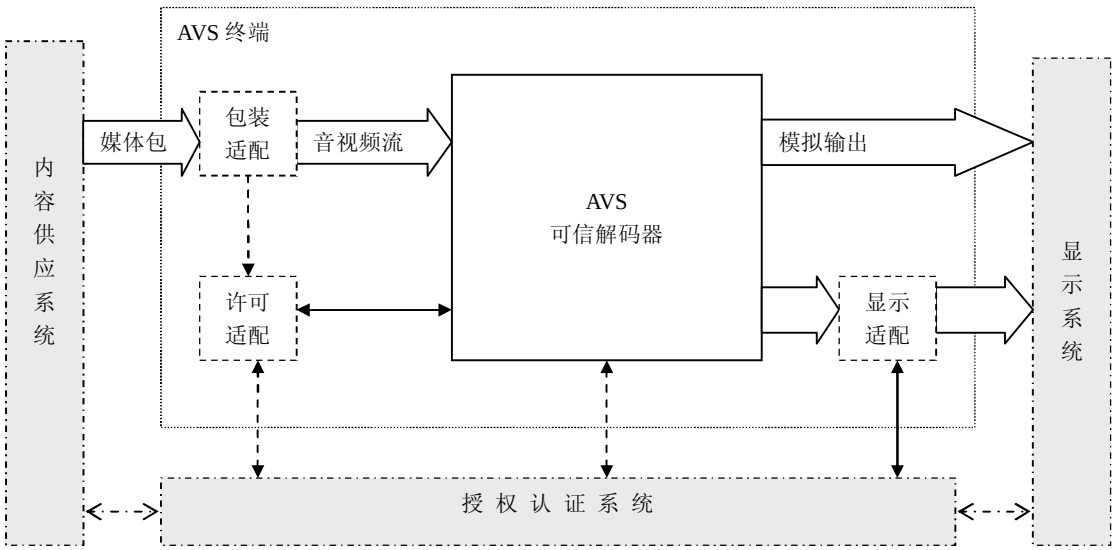


图 1 AVS 数字版权管理的概要参考模型

可信解码器是普通解码器的扩展，增加了认证、解密、明文重构等单元和输出加密等可选单元。

可信解码器的外围环境包括内容供应系统、授权认证系统和显示系统，标准的本部分定义可信解码器与这些系统之间的接口。

适配层是可信解码器和外围环境的连接层，解决可信解码器和外围环境之间的互连互操作问题。包括：适应内容供应系统的包装适配层，适应授权认证系统的许可适配层和适应显示系统的显示适配层。本标准没有具体定义各适配层所应有的操作。

4.2. 档

GB/T 20090.6-YYYY 目前分为三个档：核心档、广播档、网络电视档。

核心档定义可信解码器的构成及其各构成单元的功能和性能要求。符合 GB/T 20090.6-YYYY 的实现必须支持核心档。

广播档针对数字电视广播等单向广播应用，基于 GB/T 20090.1-YYYY 所定义的传输流，定义 AVS 接收终端能够解析的版权描述信息的语法。广播档建立在本部分定义的核心档基础上。

网络电视档针对 IP 传输环境下音视频节目广播、点播和下载等服务，基于 GB/T 20090.8-YYYY 所定义的网络封装格式和 GB/T 20090.9-YYYY 所定义的文件封装格式，定义 AVS 接收终端能够解析的版权描述信息的语法和内容封装格式。网络电视档建立在本部分定义的核心档基础上。

本部分还可能面向其它 AVS 应用定义相应的档以及支持多种数字版权管理系统互操作的互操作档。

5. 核心档

可信解码器在传统音视频解码器基础上增加了一些安全单元（图 2）。认证单元是采用公钥算法的密码单元，用于可信解码器的身份认证，支持与其它设备建立安全信道，支持数字签名。解密单元利用从认证单元获得的密钥对加密过的 AVS 内容进行解密。明文重构单元把含有加密数据的媒体流恢复成符合 AVS 标准的媒体流，从而使得重构结果能够直接送入符合 AVS 标准的解码器。

可信解码器还可包含解码输出流加密单元，支持把解码后的音频、视频流以加密方式输出给显示设备。

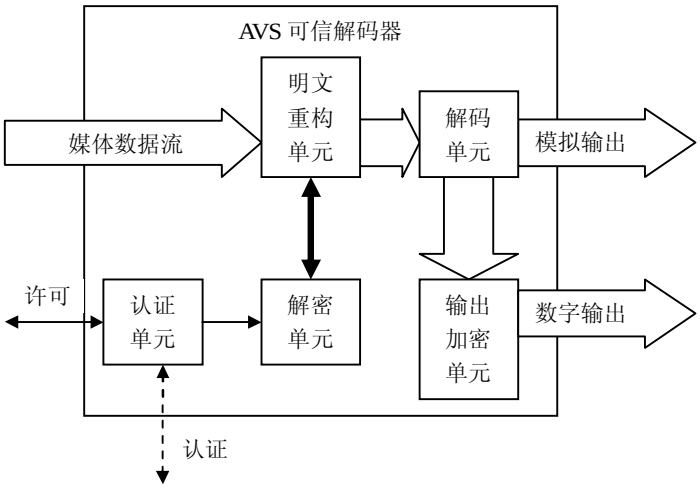


图 2 AVS 可信解码器结构示意图

5.1. 密码算法

在标准的本部分用到的或可能用到的密码算法及其命名和编号如表 1：

表 1 GB/T 20090.6-yyyy 可能用到的密码算法表

算法类别	类别名称	类别编号	算法名称	算法编号
杂凑算法	HashAlgorithm	0000	HashAlgorithm:SCH-192	0000
			HashAlgorithm:SHA-256	0001
			HashAlgorithm:SHA-1	0010
			保留	0011-1111
公钥加密算法	PublicKeyAlgorithm	0001	PublicKeyAlgorithm:SMA-192	0000
			PublicKeyAlgorithm:RSA-1024	0001
			保留	0010-1111
分组密码算法	BlockCipherAlgorithm	0010	BlockCipherAlgorithm:SMB-128-128	0000
			BlockCipherAlgorithm:AES-128-128	0001
			BlockCipherAlgorithm:3DES-64-112	0010
			BlockCipherAlgorithm:RC4	0011
			保留	0100-1111
流密码算法	StreamCipherAlgorithm	0011	StreamCipherAlgorithm:SMS-128	0000
			保留	0001-1111
签名算法	SignatureAlgorithm	0100	SignatureAlgorithm:SMA-SCH-192	0000

			SignatureAlgorithm:RSA-SHA-1024	0001
随机数生成算法	RNGAlgorithm	0101	RNGAlgorithm:SMR	0000
			保留	0001
保留	保留	0110-1111	保留	0000-1111

表 1 中，算法编号为 0000 的算法为国家密码应用主管部门指定的算法，所有符合标准本部分的实现都应实现这些算法。其它算法根据有关规定选择。

在本部分中在涉及到算法引用时，文本描述可引用算法名称或者编号，二者等价，例如“HashAlgorithm: SCH-192”与“0x00000000”均指国家密码管理局指定的杂凑算法，但在消息描述等数据定义中只能采用编号方式指代。

5.2. 内容解密单元

内容解密算法为对称算法，其中 StreamCipherAlgorithm:SMS-128 为 AVS 解码器必须支持的算法，此算法支持每次 1 比特的密码序列。

对内容进行加解密的密钥称为 CK，CK 长度记为 CK_Length。

实现内容解密算法的部件称为解密单元。对解密单元的性能要求为解密速度不小于 40Mbps。

5.3. 认证单元

认证单元是采用公钥算法的密码单元，用于可信解码器的身份认证，支持与其它设备建立安全信道，支持数字签名。

认证单元采用 PublicKeyAlgorithm:SMA-192，此算法为必须实现的缺省算法，其它算法可选。

可信解码器的公钥（PK）通过可靠方式提交给权利发布方，用于权利发布方机密信息（例如业务密钥 SK）进行加密并安全地发送给用户。

公私钥对的生成和管理方法不在本标准定义范围内，由配套的密钥算法及应用技术体系规范定义。

若采用证书方式对可信解码器进行身份认证，则应采用本认证单元的公私钥对生成证书，并用本认证单元验证获得的证书。

可信解码器需对数据进行签名或验证时，必须使用本认证单元完成。

认证单元支持数字签名验证功能，以对获得的许可证等关键数据的完整性和来源进行鉴别。

可信解码器支持通过数字签名验证功能对内容发布者的身份进行鉴别，这种情况下，认证单元验证签名的能力要求为每秒至少 30 次。

认证单元负责对许可证的存储和解析，并根据许可证中的权利要求和限制对解密解码单元进行相应的管理和控制。

5.4. 解码输出流加密单元

若可信解码器支持对解码后的音频、视频数据流以加密方式输出给显示设备，则应实现本节定义的解码输出流加密单元。

解码输出流加密单元的使用依赖于可信解码器和显示设备之间的相互认证，可信解码器采用 5.2 定义的认证单元进行这种认证。完成认证后可信解码器与显示设备通过协商获得加密密钥。

解码输出流加密单元采用的算法为 StreamCipherAlgorithm:SMS-128。

解码输出流加密单元的加密速度应不低于 1.2Gbps。

5.5. 带保护的压缩流输出功能

若可信解码器支持对获得的压缩流进行本地存储或转发给其它设备，则应根据权利要求决定直接输出解密后的码流还是以密文方式输出。

压缩流以密文方式转发时，应保持压缩流的加密状态，同时按照权利要求生成新的许可。

5.6. 明文重构单元

5.6.1. 内容保护模式

内容保护模式定义媒体流中哪些部分是受保护的，记号为 AVSEncryptionMode，这是一个 5bit 的整数。其中模式 0 和 1 为必需支持，其余模式为可选。

5.6.1.1. 无保护模式（模式 0）

在其作用范围内的全部内容数据均不保护。

5.6.1.2. 全保护模式（模式 1）

全保护模式，在其作用范围内全部内容数据均保护。

5.6.1.3. 视频帧基本流模式（模式 2）

除序列头、图像头、条带头之外实体数据进行保护。

5.6.1.4. se(v)和 ce(0)保护模式（模式 3）

仅对 P 帧中的 se(v)和 ce(0)符号加密。

5.6.1.5. 视频帧内 M 序列选择保护模式（模式 4）

本模式的保护对象为一帧。

本模式的可选保护比特为帧内预测模式、运动向量和每个块的 DCT 系数的熵编码结果的最低位，其它比特均不保护。

保护比特按次序从可选保护比特中挑选，挑选规则如下：根据附录 C 规定的 m 序列发生器，生成伪随机序列，该随机序列和可选保护比特的位置一一对应，若当前随机值为 1，且随机序列中当前比特之前值为 1 的比特个数为加密间隔 N 的整数倍，则此位置对应的可选保护比特被挑选出来。当挑选得到的比特数构成一个分组时，对齐进行解密，然后放回码流的原位替换掉原来比特。当一帧内的码流最后不足以挑选出一个完整分组时，则不再执行解密和替换操作。如果一帧不足以挑出一个完整分组时，则此帧不加密/解密。

如果不指定，则 m 序列发生器初相位取值为 01011010010110100101101101，加密间隔 N 为 15。

初相位取值和加密间隔也可通过约定方式指定，这种约定方式可以在应用档中规定。

5.6.1.6. 音频帧增强层保护模式（模式 5）

保护作用范围内的增强层数据。根据保护强度，调整需加密的增强层层数，加密方向为从高层往低层方向。加密层数的指示方法由应用档具体规定。若未指定层数，则表示所有增强层均为密文。

5.6.2. 明文重构过程

明文重构单元根据内容保护模式，将一个独立加解密的媒体数据流中受保护部分按先后顺序以比特为单位抽取出比特流送给解密单元，解密单元解密后将结果送回明文重构单元，重构单元将解密后的比特流按先后次序以比特为单位放置回原来位置，从而实现媒体数据的重构，然后输出给解码单元解码。

5.7. 密钥管理

5.7.1. 内容密钥

对内容进行解密的密钥称为 CK。

CK 由内容主控密钥 MK 和与内容流携带的辅助密钥（即 SaltKey，简称 AK）采用下面的推导算法得到。

辅助密钥 AK 与受保护内容的同步对应关系由各应用档规定，

AK 随着内容流的变化按照一定周期更新，本标准不规定更新策略。

5.7.1.1. 内容密钥推导算法

推导算法需要使用杂凑算法，这里采用 HashAlgorithm:SCH-192。

CK 的长度 CK_Length 根据所采用的内容解密算法确定，HashAlgorithm 的输出比特数记为 Hash_Length。

CK 的计算方法如下：

1. 如果 CK_Length 小于或等于 Hash_Length，则：

CK ＝ LEFT（hash（MK||AK），CK_Length）

2. 如果 CK_Length 大于 Hash_Length，则：

CK ＝ LEFT（hash（MK||AK||＂0001＂）|| hash（MK||SK||＂0002＂）||...|| hash（MK||SK||＂000n＂）,CK_Length）

其中：

(1) LEFT(s,i)：取字符串 s 的左 i 个字符；

(2) s||t：字符串 s 和 t 串联操作；

(3) n 等于 CK_Length / Hash_Length 向上取整。

5.7.2. 内容主控密钥 MK

内容主控密钥（简称 MK）是针对特定内容、节目或服务的密钥。

MK 可以采用分组密码算法进行保护，这种情况下，BlockCipherAlgorithm:SMB-128-128 必须支持，保护 MK 的密钥称为业务密钥（简称 SK），长度为 128bit。

5.7.3. 认证单元公私钥

认证单元中的公钥（简称 PK）和私钥是本 DRM 模型的信任根，但其管理不在本标准的范围之内。

权利拥有者使用目标可信解码器的公钥 PK 分发业务密钥 SK，再用 SK 分发内容主控密钥 MK 的方式分发许可。

权利拥有者也使用目标可信解码器的公钥 PK 直接分发内容主控密钥 MK。

5.8. 可信解码器的安全保护要求

可信解码器应作为一个整体进行保护，例如一颗芯片或经过专门保护的软件内核，应符合以下安全技术功能要求：

(1) 保护可信解码器免受未授权的操作或使用的影响。

(2) 保护可信解码器中各密码单元免受未授权的操作或使用的影响。

(3) 防止可信解码器中各单元之间所交换数据的未经授权的泄露。

(4) 防止密码单元内容的未经授权的泄露，包括明文密钥和其它关键安全参数。

(5) 防止密码单元的未经授权的修改，包括未经授权的修改、替代、插入和密钥及其它的关键安全参数的删除。

(6) 使用安全措施保护敏感信息。

(7) 使用安全措施检测密码单元操作中的错误，防止由于这些错误引起的对敏感数据和关键安全参数造成的危害。

(8) 可信解码器安全模块应支持安全升级。

6. 网络电视档

6.1. 网络电视档数字版权管理参考模型

网络电视档数字版权管理参考模型如图 3，它是 AVS 数字版权管理概要参考模型的一种具体化和扩展。

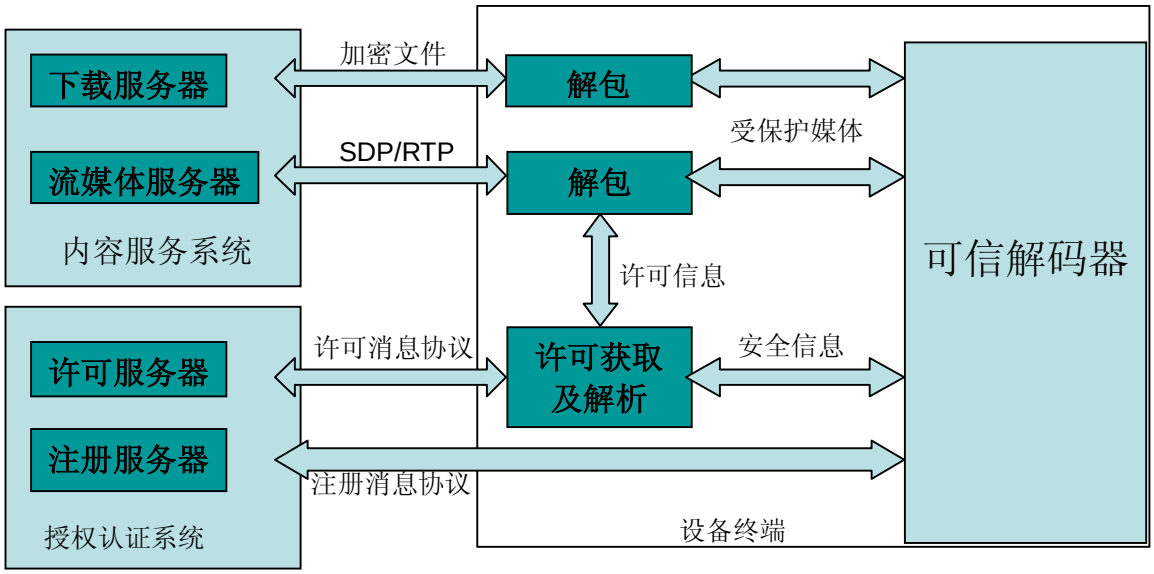


图 3 网络电视档数字版权管理参考模型

网络电视档的技术要求包括：系统协议消息、媒体内容打包（包括文件格式和 RTP 封装）和权利描述语言。

权利描述语言部分中所使用的基于 XML 的许可证以及在 AVS DRM 系统协议机制中所使用的基于 XML 的消息，推荐使用二进制的信息格式以减少 AVS 解码器所需的负荷。

6.2. 系统协议消息

不同数字版权管理系统涉及的协议消息由于应用不同而有所不同，但是几乎所有的数字版权管理都需要涉及以下两大类协议消息：（认证媒体终端身份的）注册管理和（管理媒体许可的）许可证获取/撤销管理。

本节定义了双向通信网络环境下的媒体终端和注册服务器、许可服务器之间的协议和消息格式，并假定通信双方均支持公钥体系来辅助双方之间的相关事务。并且还假定：

(1) 假定通信通道是双向的；

(2) 各种服务器，例如注册服务器，许可证服务器等都只是为了说明方便，它们代表的是在 DRM 系统中的一种角色，是逻辑服务器，在具体实现中，可能是一个物理服务器承担多种角色，对应着多个逻辑服务器，也可能是多个物理服务器共同承担一个角色，对应着一个逻辑服务器。

6.2.1. 概述

本规范定义的协议主要包括：1.注册协议，2. 许可证获取/撤销协议。注册协议是用来用户向注册服务器注册身份。许可证获取/撤销协议是用户用来向许可证服务器获取和撤销许可证。

6.2.1.1. 注册协议

注册协议是用来在注册服务器和客户端之间进行完整的安全信息交互和握手，通常只在

最开始的时候执行一次。本协议包括协议参数和协议版本、加密算法等的协商。

注册协议的成功完成，将会在服务器上建立用户的帐户信息，以及返回给用户一个 UUID 和密码（可选）。以后用户就可以使用他的 UUID 和密码获取及撤销许可证。

6.2.1.2. 许可证获取协议

许可证获取协议是用来支持客户端向许可证服务器请求许可证。本协议包括协议参数和协议版本、加密算法等的协商、权利信息的传递等。

6.2.1.3. 许可证撤销协议

许可证撤销协议是用于撤销客户端先前获取的许可证。本协议既可以由服务器端发起，也可以由客户端自行发起。

6.2.2. 消息定义

在以下消息参数表中，”M”代表必需，”O”代表可选，”-”表示未定义。

在下述协议中使用的密码算法应符合核心档中关于密码算法所做的规定，消息协议中涉及：

(1) 客户端支持的加密算法
(2) 用户个人信息的加密算法
(3) 通信密钥的加密算法
(4) 杂凑算法
(5) 签名算法

下述所有消息中的签名应用于 Signature 项以外所有数据的签名。

6.2.2.1. 注册管理协议

注册管理协议主要包括四个消息：ClientRegHello，ServerRegHello，RegisterRequest 和 RegisterResponse。

6.2.2.1.1. ClientRegHello

ClientRegHello 消息是从客户端送往注册服务器的，是用来初始化注册过程的，主要是协商协议版本、加密算法等。

参数	
Version	M
DeviceID	M
UserPubKey	M
SupportedAlgorithms	M
Extensions	O

Version：使用<大版本号.小版本号>格式来表示客户端支持的最高的版本号。在这一版本的协议中，Version 这个字段必须设置成“1.0”。小版本号的升级必须能够后向兼容。

DeviceID：用来向服务器唯一标识设备。

UserPubKey：用户公钥信息，例如数字证书等。

SupportedAlgorithms：是用来描述客户端支持的加密算法。

Extensions：是用来描述一些扩展信息，对扩展信息的使用和定义在本协议中没有明确的规定，可以根据不同的情况自己定义。

6.2.2.1.2. ServerRegHello

ServerRegHello 消息是从注册服务器送往客户端的，是用来回应客户端送往注册服务器的 ClientRegHello 消息的，是用来告诉客户端决定采用的协议版本等。

参数		
	成功，即 Status=1	失败，即 Status=其他
Status	M	M

ErrorReason	-	O
SessionID	M	-
SelectedVersion	M	-
SelectedAlgorithm	M	-
ServerPubKey	M	-
ServerInfo	O	-
Extensions	O	-

Status：指明 ClientRegHello 是否成功处理。1 表示成功处理，其他表示处理失败。

ErrorReason：指明处理失败的原因。

SessionID：是由注册服务器设置的唯一的会话 ID。这允许同时有几个客户端采用短连接的方式与注册服务器通信。

SelectedVersion：表示选择的协议版本。

SelectedAlgorithm：表示服务器选择的算法，这些算法是从 ClientRegHello 消息中的 SupportedAlgorithm 的子集。

ServerPubKey：服务器的公钥信息，如数字证书等。

ServerInfo：是用来描述服务器的一些信息。

Extensions：是用来描述一些扩展信息，对扩展信息的使用和定义在本协议中没有明确的规定，可以根据不同的情况自己定义。

6.2.2.1.3. RegisterRequest

RegisterRequest 消息是从客户端送往注册服务器端的，具体发起注册请求。

参数	
SessionID	M
CommunicationKey	M
CommunicationKeyEncryptAlgorithm	M
UserInfo	M
UserInfoEncryptAlgorithm	M
Extensions	O
HashAlgorithm	M
Signature	M
SignatureAlgorithm	M

SessionID：与前一个 ServerRegHello 消息中的 SessionID 相同，否则服务器将终止注册协议。

CommunicationKey：由客户端产生的用于通信的密钥的密文。

CommunicationKeyEncryptAlgorithm：通信密钥的加密算法，通信密钥一般使用服务器的公钥进行加密。

UserInfo：用户的个人信息的密文。

UserInfoEncryptAlgorithm：用户个人信息的加密算法。

Extensions：是用来描述一些扩展信息，对扩展信息的使用和定义在本协议中没有明确的规定，可以根据不同的情况自己定义。

HashAlgorithm：杂凑算法，指签名算法使用的杂凑算法。

Signature：客户端对他发送消息的签名。

SignatureAlgorithm：签名算法。

6.2.2.1.4. RegisterResponse

RegisterResponse 消息是从注册服务器端送往客户端的，响应客户端的注册请求。

参数	

	成功，即 Status=”Success”	失败，即 Status=”Failure”
Status	M	M
ErrorReason	-	O
SessionID	M	-
UserUUID	M	-
UserPassword	O	-
Extensions	O	-
HashAlgorithm	M	-
Signature	M	-
SignatureAlgorithm	M	-

Status：指明 RegisterRequest 是否成功处理。1 表示成功处理，其他表示处理失败。

ErrorReason：指明处理失败的原因。

SessionID：与前一个 RegisterRequest（和 ServerRegHello）消息中的 SessionID 相同，否则服务器将终止注册协议。

UserUUID：注册服务器为用户产生的独一无二的用户身份标识，用户身份标识的组成由注册服务器依照它的规则产生。

UserPassword：用户密码。

HashAlgorithm：杂凑算法，指签名算法使用的杂凑算法。

Extensions：是用来描述一些扩展信息，对扩展信息的使用和定义在本协议中没有明确的规定，可以根据不同的情况自己定义。

Signature：服务器对他发送消息的签名。

SignatureAlgorithm：签名算法。

6.2.2.2. 许可证获取协议

许可证获取协议主要包括四个消息：ClientLicHello，ServerLicHello，LicenseRequest 和 LicenseResponse。

6.2.2.2.1. ClientLicHello

ClientLicHello 消息是从客户端送往许可证服务器端的，是用来初始化许可证获取过程的，主要是协商协议版本，加密算法等。

参数	
Version	M
DeviceID	M
UserPubKey	M
SupportedAlgorithms	M
Extensions	O

Version：使用<大版本号.小版本号>格式来表示客户端支持的最高的版本号。在这一版本的协议中，Version 这个字段必须设置成“1.0”。小版本号的升级必须能够后向兼容。

DeviceID：用来向服务器唯一标识设备。

UserPubKey：是用户的公钥信息，如数字证书等。

SupportedAlgorithms：是用来描述客户端支持的加密算法。

Extensions：是用来描述一些扩展信息，对扩展信息的使用和定义在本协议中没有明确的规定，可以根据不同的情况自己定义。

6.2.2.2.2. ServerLicHello

ServerLicHello 消息是从许可证服务器送往客户端的，是用来回应客户端送往许可证服务器的 ClientLicHello 消息的，是用来告诉客户端决定采用的协议版本等。

参数		
	成功，即 Status=”1”	失败，即 Status=”0”
Status	M	M
ErrorReason	-	O
SessionID	M	-
SelectedVersion	M	-
SelectedAlgorithm	M	-
ServerPubKey	M	-
ServerInfo	O	-
Extensions	O	-

Status：指明 ClientLicHello 是否成功处理。1 表示成功处理，其他表示处理失败。

ErrorReason：指明处理失败的原因。

SessionID：是由许可证服务器设置的唯一的会话 ID。这允许同时有几个客户端采用短连接的方式与许可证服务器通信。

SelectedVersion：表示选择的协议版本。

SelectedAlgorithm：表示服务器选择的算法，这些算法是从 ClientLicHello 消息中的 SupportedAlgorithm 的子集。

ServerPubKey：服务器的公钥信息，如数字证书等。

ServerInfo：是用来描述服务器的一些信息。

Extensions：是用来描述一些扩展信息，对扩展信息的使用和定义在本协议中没有明确的规定，可以根据不同的情况自己定义。

6.2.2.2.3. LicenseRequest

LicenseRequestt 消息是从客户端送往许可证服务器端的，具体发起获取许可证的请求。

参数	
SessionID	M
CommunicationKey	M
CommunicationKeyEncryptAlgorithm	M
CapabilityInfo	M
UserUUID	M
UserPassword	O
RightsInfo	M
ServerInfo	O
Extensions	O
HashAlgorithm	M
Signature	M
SignatureAlgorithm	M

SessionID：与前一个 ServerLicHello 消息中的 SessionID 相同，否则服务器将终止许可证获取协议。

CommunicationKey：由客户端产生的用于通信的密钥的密文。

CommunicationKeyEncryptAlgorithm：通信密钥的加密算法，通信密钥一般使用服务器的公钥进行加密。

CapabilityInfo：设备的能力描述，如支不支持 Secure Clock 和 Secure Timer 等，当设备不支持 Secure Clock 和 Secure Timer 时，与时间相关的权利描述就得进行转换。比如，许可证服务器在发行许可证前，可能需要参考 CapabilityInfo，在必要的时候将原始的许可证转换为客户端

可处理的许可证。

UserUUID：用户的身份标识。

UserPassword：用户密码，应采用密文方式传输。

RightsInfo：权利信息，应符合 DREL 规范的 XML 权利信息子集，必需包括内容标识， 也可携带一些与该内容相关的权利、约束及义务请求信息。

ServerInfo：是用来描述服务器的一些信息。

Extensions：是用来描述一些扩展信息，对扩展信息的使用和定义在本协议中没有明确的规定，可以根据不同的情况自己定义。

HashAlgorithm：杂凑算法，指签名算法使用的杂凑算法。

Signature：客户端对他发送消息的签名。

SignatureAlgorithm：签名算法。

6.2.2.2.4. LicenseResponse

LicenseResponse 消息是从许可证服务器端送往客户端的，响应客户端的许可证获取请求。

参数		
	成功，即 Status=”Success”	失败，即 Status=”Failure”
Status	M	M
ErrorReason	-	O
SessionID	M	-
LicenseInfo	M	-
Extensions	O	-
HashAlgorithm	M	-
Signature	M	-
SignatureAlgorithm	M	-

Status：指明 LicenseRequest 是否成功处理。1 表示成功处理，其他表示处理失败。

ErrorReason：指明处理失败的原因。

SessionID：与前一个 LicenseRequest（和 ServerLicHello）消息中的 SessionID 相同，否则服务器将终止许可证获取协议。

LicenseInfo：许可证信息，应符合 DREL 规范的 XML 权利信息，可包括权利、约束及义务信息，及相关的密钥信息。

Extensions：是用来描述一些扩展信息，对扩展信息的使用和定义在本协议中没有明确的规定，可以根据不同的情况自己定义。

HashAlgorithm：杂凑算法，指签名算法使用的杂凑算法。

Signature：服务器对他发送消息的签名。

SignatureAlgorithm：签名算法。

6.2.2.3. 许可证撤销协议

许可证撤销协议主要包括四个消息：ClientLicHello， ServerLicHello，LicenseRevokeRequest，LicenseRevokeResponse 和 LicenseRevokeInstruct。

ClientLicHello 和 ServerLicHello 消息保持与已有许可证获取协议中的同名消息一样，即重用已有的消息，包括其功能和格式。

6.2.2.3.1. LicenseRevokeRequest

LicenseRevokeRequest 消息是从客户端送往许可证服务器端的，具体发起撤销许可证的请求。

参数	
SessionID	M

CommunicationKey	M
CommunicationKeyEncryptAlgorithm	M
UserUUID	M
UserPassword	O
LicenseID	M
RightsInfo	O
Extensions	O
HashAlgorithm	M
Signature	M
SignatureAlgorithm	M

SessionID：与前一个 ServerLicHello 消息中的 SessionID 相同，否则服务器将终止许可证撤销协议。

CommunicationKey：由客户端产生的用于通信的密钥的密文。

CommunicationKeyEncryptAlgorithm：通信密钥的加密算法，通信密钥一般使用服务器的公钥进行加密。

UserUUID：用户的身份标识。

UserPassword：用户密码，应采用密文方式传输。

LicenseID：许可证号。

RightsInfo：许可证中的权利信息。（可以将许可证传送给许可证服务器，防止许可证服务器没有许可证发布记录）

Extensions：是用来描述一些扩展信息，对扩展信息的使用和定义在本协议中没有明确的规定，可以根据不同的情况自己定义。

HashAlgorithm：杂凑算法，指签名算法使用的杂凑算法。

Signature：客户端对他发送消息的签名。

SignatureAlgorithm：签名算法。

6.2.2.3.2. LicenseRevokeResponse

LicenseRevokeResponse 消息是从许可证服务器端送往客户端的，响应客户端的许可证撤销请求。

参数		
	成功，即 Status=”Success”	失败，即 Status=”Failure”
Status	M	M
ErrorReason	-	O
SessionID	M	-
Extensions	O	-
HashAlgorithm	M	-
Signature	M	-
SignatureAlgorithm	M	-

Status：指明 LicenseRevokeRequest 是否成功处理。1 表示成功处理，允许客户端撤销许可证，其他表示处理失败。

ErrorReason：指明处理失败的原因。

SessionID：与前一个 LicenseRevokeRequest（和 ServerLicHello）消息中的 SessionID 相同，否则服务器将终止许可证撤销协议。

Extensions：是用来描述一些扩展信息，对扩展信息的使用和定义在本协议中没有明确的规定，可以根据不同的情况自己定义。

HashAlgorithm：杂凑算法，指签名算法使用的杂凑算法。

Signature：服务器对他发送消息的签名。

SignatureAlgorithm：签名算法。

6.2.2.3.3. LicenseRevokeInstruct

LicenseRevokeInstruct 消息是从服务器端送往许可证客户端的，具体通知客户端发起撤销许可证的请求。

参数	
SessionID	M
LicenseID	M
UserUUID	M
Extensions	O
HashAlgorithm	M
Signature	M
SignatureAlgorithm	M

SessionID：与前一个 ServerLicHello 消息中的 SessionID 相同，否则服务器将终止许可证撤销协议。

LicenseID：许可证号。

UserUUID：注册服务器为用户产生的独一无二的用户身份标识，用户身份标识的组成由注册服务器依照它的规则产生。

ServerInfo：是用来描述服务器的一些信息。

Extensions：是用来描述一些扩展信息，对扩展信息的使用和定义在本协议中没有明确的规定，可以根据不同的情况自己定义。

HashAlgorithm：杂凑算法，指签名算法使用的杂凑算法。

Signature：客户端对他发送消息的签名。

SignatureAlgorithm：签名算法。

6.2.2.3.4. LicenseRevokeConfirm

LicenseRevokeConfirm 消息是从客户端送往许可证服务器端的，确认服务器的许可证撤销响应。

参数		
	成功，即 Status=”Success”	失败，即 Status=”Failure”
Status	M	M
ErrorReason	-	O
SessionID	M	-
Extensions	O	-
HashAlgorithm	M	-
Signature	M	-
SignatureAlgorithm	M	-

Status：指明 LicenseRequest 是否成功处理。1 表示成功处理，其他表示处理失败。

ErrorReason：指明处理失败的原因。

SessionID：与前一个 LicenseRevokeResponse（和 LicenseRevokeRequest、ServerLicHello）消息中的 SessionID 相同，否则服务器将终止许可证撤销协议。

Extensions：是用来描述一些扩展信息，对扩展信息的使用和定义在本协议中没有明确的规定，可以根据不同的情况自己定义。

HashAlgorithm：杂凑算法，指签名算法使用的杂凑算法。

Signature：客户端对他发送消息的签名。

SignatureAlgorithm：签名算法。

6.2.3. 消息命令描述形式

6.2.2 节定义了各消息的参数及各个参数的意义。本节定义各消息命令描述格式。

命令描述形式的格式为：命令头+命令体，其中命令头的格式和长度固定，命令体的格式和长度随命令的不同而不同。

命令头的格式如下：

Command_ID	int（四个字节）	用来标志命令
Body_Length	int（四个字节）	用来指明命令体的长度
Reserved_1	int（四个字节）	保留
Reserved_2	int（四个字节）	保留

6.2.3.1.1. 注册协议

6.2.3.1.1.1. ClientRegHello

字段	类型
VersionLen	int (四个字节)
Version	String (VersionLen 个字节)
DeviceIDLen	int (四个字节)
DeviceID	String (DeviceIDLen 个字节)
UserPubKeyLen	int (四个字节)
UserPubKey	String (UserPubKeyLen 个字节)
SupportedAlgorithmsLen	int (四个字节)
SupportedAlgorithms	String (SupportedAlgorithmsLen 个字节)
ExtensionsLen	int (四个字节)
Extensions	String (ExtensionsLen 个字节)

6.2.3.1.1.2. ServerRegHello

字段	类型
StatusLen	int (四个字节)
Status	String (StatusLen 个字节)
ErrorReasonLen	int (四个字节)
ErrorReason	String (ErrorReasonLen 个字节)
SessionIDLen	int (四个字节)
SessionID	String (SessionIDLen 个字节)
SelectedVersionLen	int (四个字节)
SelectedVersion	String (SelectedVersionLen 个字节)
SelectedAlgorithmLen	int (四个字节)
SelectedAlgorithm	String (SelectedAlgorithmLen 个字节)
ServerPubKeyLen	int (四个字节)
ServerPubKey	String (ServerPubKeyLen 个字节)
ServerInfoLen	int (四个字节)

ServerInfo	String (ServerInfoLen 个字节)
ExtensionsLen	int (四个字节)
Extensions	String (ExtensionsLen 个字节)

6.2.3.1.1.3. RegisterRequest

字段	类型
SessionIDLen	int (四个字节)
SessionID	String (SessionIDLen 个字节)
CommunicationKeyLen	int (四个字节)
CommunicationKey	String (CommunicationKeyLen 个字节)
CommunicationKeyEncryptAlgorithmLen	int (四个字节)
CommunicationKeyEncryptAlgorithm	String (CommunicationKeyEncryptAlgorithmLen 个字节)
UserInfoLen	int (四个字节)
UserInfo	String (UserInfoLen 个字节)
UserInfoEncryptAlgorithmLen	int (四个字节)
UserInfoEncryptAlgorithm	String (UserInfoEncryptAlgorithmLen 个字节)
ExtensionsLen	int (四个字节)
Extensions	String (ExtensionsLen 个字节)
HashAlgorithmLen	int (四个字节)
HashAlgorithm	String (HashAlgorithmLen 个字节)
SignatureLen	int (四个字节)
Signature	String (SignatureLen 个字节)
SignatureAlgorithmLen	int (四个字节)
SignatureAlgorithm	String (SignatureAlgorithmLen 个字节)

6.2.3.1.1.4. RegisterResponse

字段	类型
StatusLen	int (四个字节)
Status	String (StatusLen 个字节)
ErrorReasonLen	int (四个字节)
ErrorReason	String (ErrorReasonLen 个字节)
SessionIDLen	int (四个字节)
SessionID	String (SessionIDLen 个字节)
UserUUIDLen	int (四个字节)
UserUUID	String (UserUUIDLen 个字节)
UserPasswordLen	int (四个字节)
UserPassword	String (UserPasswordLen 个字节)
ExtensionsLen	int (四个字节)
Extensions	String (ExtensionsLen 个字节)
HashAlgorithmLen	int (四个字节)

HashAlgorithm	String (HashAlgorithmLen 个字节)
SignatureLen	int (四个字节)
Signature	String (SignatureLen 个字节)
SignatureAlgorithmLen	int (四个字节)
SignatureAlgorithm	String (SignatureAlgorithmLen 个字节)

6.2.3.1.2. 许可证获取协议

6.2.3.1.2.1. ClientLicHello

字段	类型
VersionLen	int (四个字节)
Version	String (VersionLen 个字节)
DeviceIDLen	int (四个字节)
DeviceID	String (DeviceIDLen 个字节)
UserPubKeyLen	int (四个字节)
UserPubKey	String (UserPubKeyLen 个字节)
SupportedAlgorithmsLen	int (四个字节)
SupportedAlgorithms	String (SupportedAlgorithmsLen 个字节)
ExtensionsLen	int (四个字节)
Extensions	String (ExtensionsLen 个字节)

6.2.3.1.2.2. ServerLicHello

字段	类型
StatusLen	int (四个字节)
Status	String (StatusLen 个字节)
ErrorReasonLen	int (四个字节)
ErrorReason	String (ErrorReasonLen 个字节)
SessionIDLen	int (四个字节)
SessionID	String (SessionIDLen 个字节)
SelectedVersionLen	int (四个字节)
SelectedVersion	String (SelectedVersionLen 个字节)
SelectedAlgorithmLen	int (四个字节)
SelectedAlgorithm	String (SelectedAlgorithmLen 个字节)
ServerPubKeyLen	int (四个字节)
ServerPubKey	String (ServerPubKeyLen 个字节)
ServerInfoLen	int (四个字节)
ServerInfo	String (ServerInfoLen 个字节)
ExtensionsLen	int (四个字节)
Extensions	String (ExtensionsLen 个字节)

6.2.3.1.2.3. LicenseRequest

字段	类型
SessionIDLen	int (四个字节)
SessionID	String (SessionIDLen 个字节)
CommunicationKeyLen	int (四个字节)

CommunicationKey	String (CommunicationKeyLen 个字节)
CommunicationKeyencryptAlgorithmLen	int (四个字节)
CommunicationKeyEncryptAlgorithm	String (CommunicationKeyEncryptAlgorithmLen 个字节)
CapabilityInfoLen	int (四个字节)
CapabilityInfo	String(CapabilityInfoLen 个字节)
UserUUIDLen	int (四个字节)
UserUUID	String (UserUUIDLen 个字节)
UserPasswordLen	int (四个字节)
UserPassword	String (UserPasswordLen 个字节)
RightsInfoLen	int (四个字节)
RightsInfo	String (RightsInfoLen 个字节)
ServerInfoLen	int (四个字节)
ServerInfo	String (ServerInfoLen 个字节)
ExtensionsLen	int (四个字节)
Extensions	String (ExtensionsLen 个字节)
HashAlgorithmLen	int (四个字节)
HashAlgorithm	String (HashAlgorithmLen 个字节)
SignatureLen	int (四个字节)
Signature	String (SignatureLen 个字节)
SignatureAlgorithmLen	int (四个字节)
SignatureAlgorithm	String (SignatureAlgorithmLen 个字节)

6.2.3.1.2.4. LicenseResponse

字段	类型
StatusLen	int (四个字节)
Status	String (StatusLen 个字节)
ErrorReasonLen	int (四个字节)
ErrorReason	String (ErrorReasonLen 个字节)
SessionIDLen	int (四个字节)
SessionID	String (SessionIDLen 个字节)
LicenseInfoLen	int (四个字节)
LicenseInfo	String (LicenseInfoLen 个字节)
ExtensionsLen	int (四个字节)
Extensions	String (ExtensionsLen 个字节)
HashAlgorithmLen	int (四个字节)
HashAlgorithm	String (HashAlgorithmLen 个字节)
SignatureLen	int (四个字节)
Signature	String (SignatureLen 个字节)
SignatureAlgorithmLen	int (四个字节)
SignatureAlgorithm	String (SignatureAlgorithmLen 个字节)

6.2.3.1.3. 许可证撤销协议

6.2.3.1.3.1. LicenseRevokeRequest：

字段	类型
SessionIDLen	int (四个字节)
SessionID	String (SessionIDLen 个字节)
CommunicationKeyLen	int (四个字节)
CommunicationKey	String (CommunicationKeyLen 个字节)
CommunicationKeyencryptAlgorithmLen	int (四个字节)
CommunicationKeyEncryptAlgorithm	String (CommunicationKeyEncryptAlgorithmLen 个字节)
UserUUIDLen	int (四个字节)
UserUUID	String (UserUUIDLen 个字节)
UserPasswordLen	int (四个字节)
UserPassword	String (UserPasswordLen 个字节)
LicenseIDLen	int (四个字节)
LicenseID	String (LicenseIDLen 个字节)
RightsInfoLen	int (四个字节)
RightsInfo	String (RightsInfoLen 个字节)
ExtensionsLen	int (四个字节)
Extensions	String (ExtensionsLen 个字节)
HashAlgorithmLen	int (四个字节)
HashAlgorithm	String (HashAlgorithmLen 个字节)
SignatureLen	int (四个字节)
Signature	String (SignatureLen 个字节)
SignatureAlgorithmLen	int (四个字节)
SignatureAlgorithm	String (SignatureAlgorithmLen 个字节)

6.2.3.1.3.2. LicenseRevokeResponse：

字段	类型
StatusLen	int (四个字节)
Status	String (StatusLen 个字节)
ErrorReasonLen	int (四个字节)
ErrorReason	String (ErrorReasonLen 个字节)
SessionIDLen	int (四个字节)
SessionID	String (SessionIDLen 个字节)
ExtensionsLen	int (四个字节)
Extensions	String (ExtensionsLen 个字节)
HashAlgorithmLen	int (四个字节)
HashAlgorithm	String (HashAlgorithmLen 个字节)
SignatureLen	int (四个字节)
Signature	String (SignatureLen 个字节)
SignatureAlgorithmLen	int (四个字节)
SignatureAlgorithm	String (SignatureAlgorithmLen 个字节)

6.2.3.1.3.3. LicenseRevokeInstruct：

字段	类型
SessionIDLen	int (四个字节)
SessionID	String (SessionIDLen 个字节)
LicenseIDLen	int (四个字节)
LicenseID	String (LicenseIDLen 个字节)
UserUUIDLen	int (四个字节)
UserUUID	String (UserUUIDLen 个字节)
ExtensionsLen	int (四个字节)
Extensions	String (ExtensionsLen 个字节)
HashAlgorithmLen	int (四个字节)
HashAlgorithm	String (HashAlgorithmLen 个字节)
SignatureLen	int (四个字节)
Signature	String (SignatureLen 个字节)
SignatureAlgorithmLen	int (四个字节)
SignatureAlgorithm	String (SignatureAlgorithmLen 个字节)

6.2.3.1.3.4. LicenseRevokeConfirm：

字段	类型
StatusLen	int (四个字节)
Status	String (StatusLen 个字节)
ErrorReasonLen	int (四个字节)
ErrorReason	String (ErrorReasonLen 个字节)
SessionIDLen	int (四个字节)
SessionID	String (SessionIDLen 个字节)
ExtensionsLen	int (四个字节)
Extensions	String (ExtensionsLen 个字节)
HashAlgorithmLen	int (四个字节)
HashAlgorithm	String (HashAlgorithmLen 个字节)
SignatureLen	int (四个字节)
Signature	String (SignatureLen 个字节)
SignatureAlgorithmLen	int (四个字节)
SignatureAlgorithm	String (SignatureAlgorithmLen 个字节)

6.2.4. 流程

6.2.4.1. 注册协议

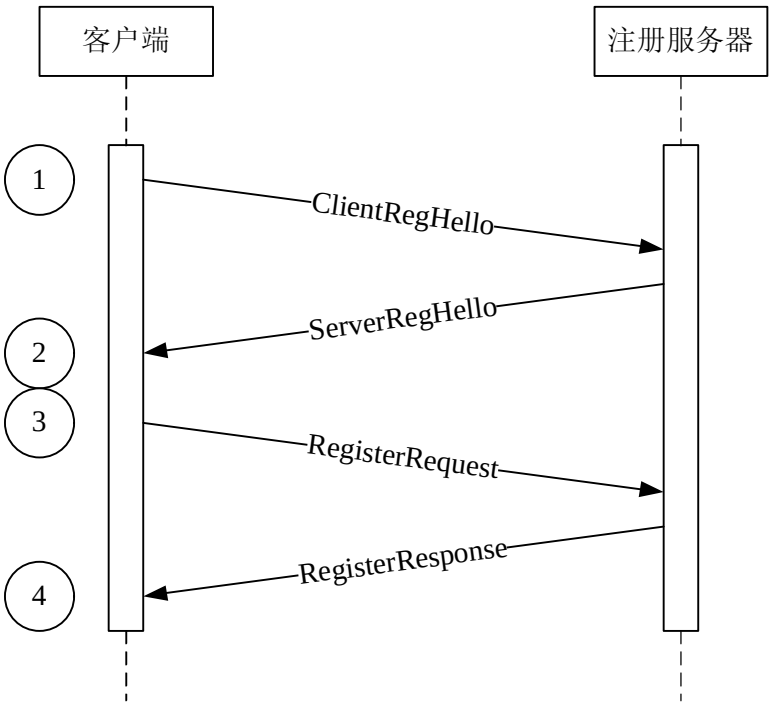


图 4 注册协议流程

(1) 客户端向注册服务器发起注册协商请求 ClientRegHello，希望就协议版本和算法等与注册服务器进行协商；

(2) 注册服务器回应客户端的请求，通过 ServerRegHello 告知客户端选定的协议版本和算法等；

(3) 客户端向注册服务器发起注册请求 RegisterRequest，按照协商的版本的协议和算法，把相关信息发送给注册服务器；

(4) 注册服务器回应客户端的请求，通过 RegisterResponse 告知客户端注册结果等。

6.2.4.1.1. 许可证获取协议

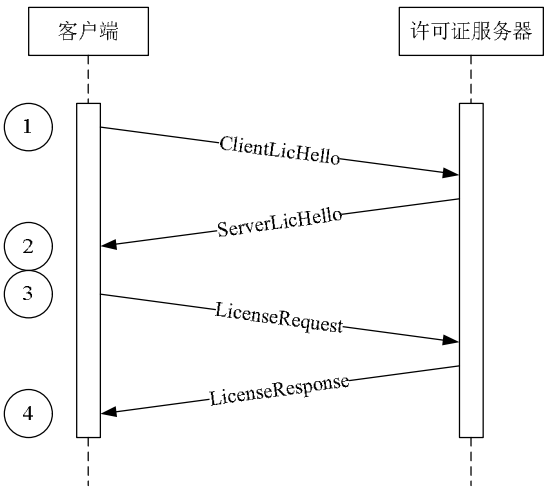


图 5 许可证获取协议流程

(1) 客户端向许可证服务器发起获取许可证协商请求 ClientLicHello，希望就协议版本和算法等与许可证服务器进行协商；

(2) 许可证服务器回应客户端的请求，通过 ServerLicHello 告知客户端选定的协议版本和算法等；

(3) 客户端向许可证服务器发起获取许可证请求 LicenseRequest，按照协商的版本的协议和算法，把相关信息发送给许可证服务器；

(4) 许可证服务器回应客户端的请求，通过 LicenseResponse 告知客户端许可证等。

6.2.4.1.2. 许可证撤销协议

6.2.4.1.2.1. 客户端发起的许可证撤销协议

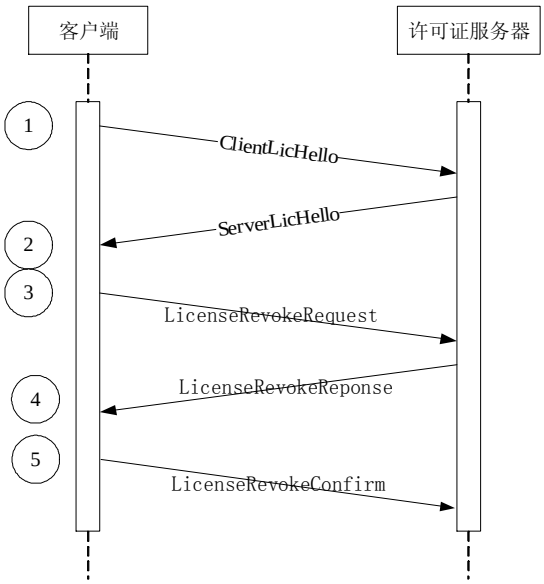


图 6 许可证撤销协议流程

(1) 客户端向许可证服务器发起获取许可证协商请求 ClientLicHello，希望就协议版本和算法等与许可证服务器进行协商；

(2) 许可证服务器回应客户端的请求，通过 ServerLicHello 告知客户端选定的协议版本和算法等；

(3) 客户端向许可证服务器发起许可证撤销请求 LicenseRevokeRequest，按照协商的版本的协议和算法，把相关信息发送给许可证服务器；

(4) 许可证服务器回应客户端的请求，通过 LicenseRevokeResponse 告知客户端许可证撤销结果等。

(5) 客户端向许可证服务器发起许可证撤销确认消息 LicenseRevokeConfirm，把终端处理的最后结果发送给许可证服务器；许可服务器收到 LicenseRevokeConfirm 后，进行后续处理，如完成计费处理，清除本地撤销记录等。如果许可服务器没有收到 LicenseRevokeConfirm 消息，需要根据撤销记录进行回滚，即将准备撤销的许可证恢复为可用，清除设置的撤销记录。

6.2.4.1.2.2. 服务器发起的许可证撤销协议

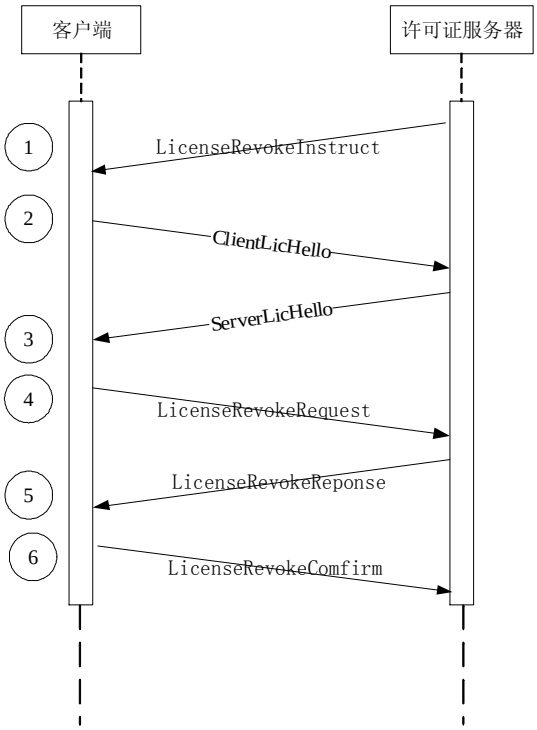


图 7 服务器发起的许可证撤销协议流程

(1) 许可证服务器向客户端发起许可证撤销通知 LicenseRevokeInstruct，把相关信息发送给客户端。

(2) 客户端向许可证服务器发起获取许可证协商请求 ClientLicHello，希望就协议版本和算法等与许可证服务器进行协商；

(3) 许可证服务器回应客户端的请求，通过 ServerLicHello 告知客户端选定的协议版本和算法等；

(4) 客户端向许可证服务器发起获取许可证请求 LicenseRevokeRequest，按照协商的版本的协议和算法，把相关信息发送给许可证服务器；

(5) 许可证服务器回应客户端的请求，通过 LicenseRevokeResponse 告知客户端许可证撤销结果等。

(6) 客户端向许可证服务器发起获取许可证请求 LicenseRevokeConfirm，把终端处理的最后结果发送给许可证服务器；

6.3. 对 AVS DREL 数字权利描述语言的采用及限制

网络电视档采用 AVS DREL 数字权利描述语言（参见第 8 章），但是必须满足以下限制。

6.3.1. 权利接受者

不支持权利接受者组。

6.3.2. 权利

普通权利中的使用类型中只支持：输出、播放。

重用权利中只支持：修改、分割、打包。

管理类型中只支持: 移动、复制、备份、保存。

高级权利中只支持: 转让类型和撤销类型。

6.3.3. 约束

时间的约束中只支持：次数、累加时间、时间段、使用周期。

6.3.4. 义务

不对义务元素进行定义。

6.4. 内容打包格式

以下阐述在内容被保护情况下数字媒体文件的封装格式和在IP网络上传输时的RTP打包格式，即 RTP 净载动态保护格式和支持版权管理的 AVS 文件格式。

6.4.1. RTP 净载动态保护格式

本节定义流媒体服务器在实时发送已打包好的明文 RTP 包时，对其净载部分进行动态选择性加密的传输保护机制。在流媒体的传输过程中，流媒体服务器可以动态地更换密钥，增强对流媒体内容的保护强度。

本机制需要配合 DREL 和 SDP 来实现，共同提供解密 RTP 载荷内容所需要的信息，包括：

(1) 内容加密密钥的主控密钥长度及密钥值（密文）

(2) 内容加密密钥的子密钥长度及密钥值

(3) 主控密钥的加密信息，如加密算法（通常为非对称算法）

(4) 内容加密密钥的推导算法

(5) 内容的加密信息，如加密算法（通常为对称算法）、IV 长度及 IV 值、Padding 类型

其中 RTP 包、DREL 和 SDP 通过密钥 ID 等信息相互关联。DREL 中应包含上述 1、3 两项。剩余信息分别包含在 RTP 包和 SDP 中。通常情况下，RTP 包的 AVSRTPDRMHeader 包含一些变化频率高的内容，SDP 则包含相对稳定的信息，根据这个原则，前者描述的内容为：

(1) 内容加密密钥的子密钥值

(2) 内容加密信息中的 IV 值

(3) Padding 的长度

SDP 描述的内容为：

(1) 内容加密密钥的子密钥长度

(2) 内容加密密钥的推导算法

(3) 内容加密信息中的加密算法、IV 长度以及 Padding 类型

6.4.1.1. 支持内容保护的 AVS RTP 净载格式

GB/T 20090.8-YYYY 基于 RFC3550 定义了 AVS 媒体数据封装为 RTP 包的方法，格式如下：

RTP Header	AVS 净载数据

图 13 AVS 数据的 RTP 封装格式

图中，RTP Header 格式符合 RFC3550，AVS 净载数据格式符合 GB/T 20090.8-YYYY。

支持内容保护的 AVS RTP 格式如下：

RTP Header	AVSRTPDRMHeader	加密的或明文的 AVS 净载数据

图 14 支持内容保护的 AVS RTP 格式

图中的净载可以是原净载的密文或明文形式，AVSRTPDRMHeader 的具体格式下面定义。

6.4.1.1.1. AVSRTPDRMHeader

AVSRTPDRMHeader 必须紧跟在 RTP 净载之后，提供关于 AVS DRM 的相关信息。具体定义如下：

```
aligned(8) class AVSRTPDRMHeader {
    int(32)    HeadLen;
    bit(1)      DataIsEncrypted;              // 加密指示符
    bit(7)      AVSEncryptionMode;            // (如果 DataIsEncrypted =0 skip;
                                                  否则，参见核心档表 XX)
    if (DataIsEncrypted == 1) {
        unsigned int(8) KeyID;
        unsigned int(8 * SaltKeyLength) SaltKey;
        unsigned    int(8 * IVLength) IVValue;
        unsigned    int(8) PaddingLength;
    }
}
```

其中

HeadLen：	AVSRTPDRMHeader 长度
DataIsEncrypted：	加密指示符
EncrytptionMode：	加密模式
KeyID：	主控密钥序号
SaltKeyLength：	辅助密钥长度（参见 SDP 定义）
SaltKey：	辅助密钥（随机数）
IVLength：	IV 长度（参见 SDP 定义）
IVValue：	IV 值
PaddingLength：	补位的长度

如果 AVSRTPDRMHeader 长度不满足 32bit 整数倍，以全 0 补足。

这样，被授权的播放设备可以利用 AVSRTPDRMHeader 中的信息来恢复 RTP 载荷数据。

6.4.1.2. SDP 格式

SDP 媒体描述格式如下：

m = <media> <port>/<number of ports> <transport> <fmt list>

a = rtpmap:<payload type> <encoding name>/<clock rate>[/<encoding parameters>]

a = fmtp:<payload type> mode=<mode>; <AVS-GENERIC-PARMS> <ENC-AVS-GENERICPARAMS>

其中：

<media> = "audio"|"video"

<transport> = "RTP/AVP"

<encoding name> = "enc-avs2" //用于标识加密的 AVS1-P2 视频编码格式

<encoding name> = "enc-avs7" //用于标识加密的 AVS1-P7 视频编码格式

<encoding name> = "enc-avs3" //用于标识加密的 AVS1-P3 音频编码格式

<mode> = "avs-video"|"avs-audio"

<AVS-GENERIC-PARMS>参见 AVS－P8 部分的定义

ENC-AVS-GENERICPARAMS 定义如下：

描述字	取值范围	默认值
CryptoIVLength	0－64（Bytes）	8（Bytes）
SaltkeyLength	0－64（Bytes）	8（Bytes）
CryptoAlg		参见核心档
CryptoPaddingType		0
CryptoKMSVersion	1.X	

说明：

CryptoIVLength： 内容加密过程采用的初始向量的长度

SaltkeyLength： 内容加密密钥的辅助密钥的长度

CryptoAlg： 内容加密算法名称

CryptoPaddingType： 加密分组补足方法。当值为 0 时，如果采用分组加密算法，对于最后一个分组，如果长度 n 小于选定算法的分组长度 m，补充 m-n 个值为(m-n)的字节；否则，补充 m 个值为 m 的字节。

CrytoKMSVersion： 密钥管理版本，当前为 1.0

如果前三个参数为默认值，可以不出现在 SDP 信息中，以节省空间。

6.4.1.2.1. 举例 （资料性）

m = audio 0 RTP/AVP 96

a = rtpmap:96 enc-avs-generic

a = fmtp:96; mode = avs-video; profile-level-id= 2; CryptoIVLength =128; SaltkeyLength =4; CryptoAlg = AES-CBC128; CryptoPaddingType =0; CryptoKMSVersion = 1.0

6.4.1.3. 应用过程 (资料性)

注册用户登录到系统后，使用许可证获取协议获得指定节目的许可证。

许可证服务器在许可中携带多个主控密钥，各个主控密钥对应的序号和推导内容密钥的算法。通过许可证，将传输过程中的所有可能使用的主控密钥及对应的序号一次性全部传给用户终端。

在 RTP 包中增加当前内容加密使用的主控密钥的序号以及辅助密钥。终端收到 RTP 包之后，取得主控密钥序号和辅助密钥，与目前的序号和辅助密钥比较，如果一致，继续使用目前内容密钥进行解密；如果不一致，根据主控密钥序号，从许可证中取得对应的主控密钥，根据标准规定的密钥推导算法，通过对主控密钥和辅助密钥进行计算，得到更换后的内容解密密钥，进行解密。

6.4.2. 支持版权管理的 AVS 文件格式

支持版权管理的 AVS 文件格式是对 GB/T 20090.9-xxxx 的扩展，GB/T 20090.9-xxxx 基于并兼容 ISO 基媒体文件格式(ISO base media file format，ISO/IEC 14496-12 | 15444-12)。支持 GB/T 20090.9-xxxx 的设备（例如流媒体服务器和播放器）可通过支持本部分定义的扩展功能而实现本部分的要求。

6.4.2.1. 文件的完整性验证

在使用受保护的 AVS 文件前必须对受保护内容进行完整性验证。为了减少用户播放媒体内容时的等待时间，设备可以在文件下载过程中或首次使用文件前计算杂凑值，并将其存储在设备中相应的存储区。

完整性验证的对象是除 AVS DRM 不保证自由空间盒的完整性，因此在计算 AVS 内容杂

凑值时，应跳过该盒。

验证受保护内容的完整性时，将存储的杂凑值与该文件关联的权限对象所携带的杂凑值进行比较，如果一致，则允许播放，否则拒绝用户的进一步操作。

这里杂凑计算的算法由许可证中的杂凑算法指示符指定，该杂凑算法必须采用核心档规定的杂凑算法。

6.4.2.2. 文件格式中的 AVS DRM 扩展信息

图 8 表示了 AVS 文件的基本结构，包括四种基本盒类型：文件类型盒(File Type Box)、媒体描述盒(Movie Box)、媒体数据盒(Media Data Box)、自由空间盒(Free Box)。其中，前三个盒必须出现，自由空间盒为可选盒。

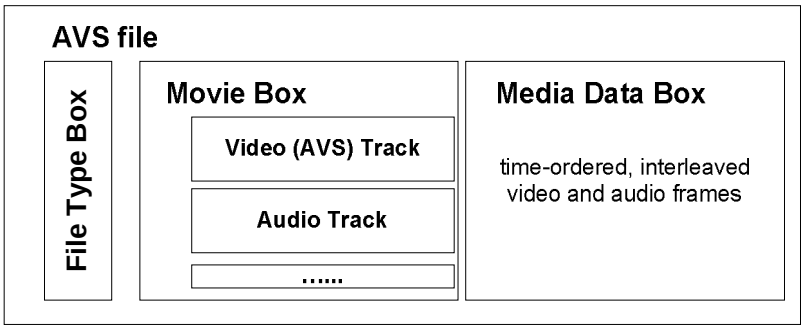


图 8 基于 ISO 基媒体文件格式的 AVS 文件格式概要模型

ISO 基媒体文件格式定义了通用的保护方案，参见图 9。以视频为例，sample decription box 的盒类型用 4CC 字符'encv'表示，其中定义 ProtectionSchemeInfo 盒用于内容保护。

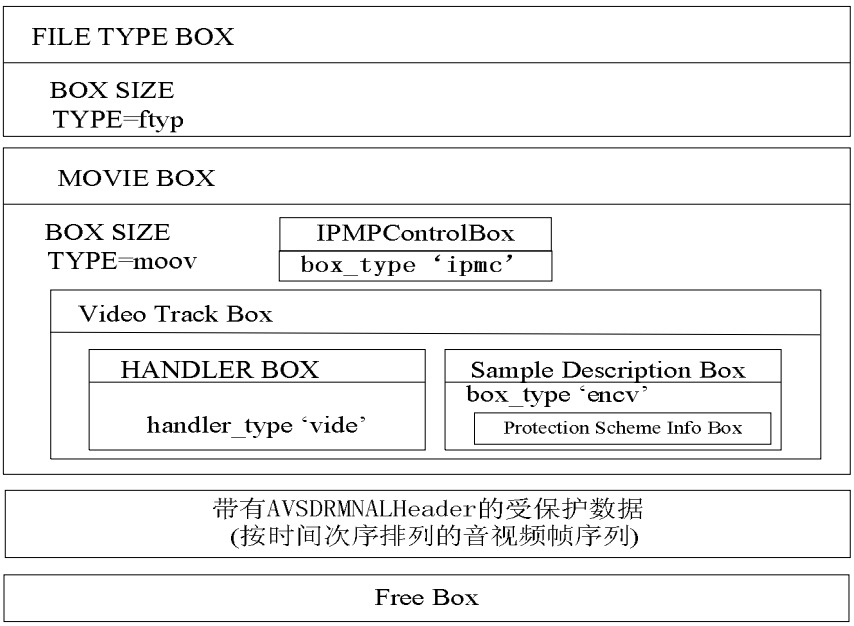


图 9 AVS 受保护媒体数据及 DRM 扩展信息的位置

6.4.2.2.1. 文件类型盒

文件类型盒是由 ISO 定义的强制盒，必须出现在文件的开头。

```
aligned(8) class FileTypeBox extends Box (‘ftyp’)  {
   unsigned int(32)    major-brand;
   unsigned int(32)    minor-version;
   unsigned int(32)    compatible-brands[];
}
```

major-brand：适合该文件的最佳解析器的标识，满足本规范的 major-brand 值(参见 GB/T 20090.9—xxxx)。

minor-version：上述解析器的版本号。

compatible-brands：兼容的其它文件解析器标识。

6.4.2.2.2. IPMP Control Box

IPMP Control Box 属于容器盒，用于存放 AVS DRM 通用保护方案有关的所有数据，应置于文件 MOOV 盒下一层。本部分规范规定该盒中扩展定义一个 AVSDRMGeneralInfo 盒，其中定义相关的通用 AVS DRM 数据。

```
aligned(8) class IPMPControlBox extends FullBox('ipmc', 0, flags) {
   IPMP_ToolListDescriptor    toollist;
   int(8)                     no_of_IPMPDescriptors;
   IPMP_Descriptor            ipmp_desc[no_of_IPMPDescriptors];
   AVSDRMGeneralInfoBox       GeneralInfo;
}
```

toollist 是一个 IPMP_ToolList 描述子；

no_of_IPMPDescriptors 是一组 IPMP 描述子阵列数目；

ipmp_desc[]是具体 IPMP 描述子阵列。

6.4.2.2.2.1. AVS DRM General Information Box

AVSDRMGeneralInfo 盒在 AVS 文件内容打包中必须出现，提供 DRM 的必要信息，如全局唯一的内容 ID 等。

```
aligned(8) class AVSDRMGeneralInfoBox extends FullBox (‘adgi’,0,flags){
   unsigned int (16)   ContentIDLength;          // 内容 ID 的长度
   char                ContentID[ ];             // 内容 ID, 全局唯一
   bit (1)             isAVSLicenseURL;          // 许可证获取方式，“0”直接
                                                 // 在本盒中携带，“1”URL 获取
   if（isAVSLicenseURL = = 1）{
      unsigned int (16)   LicenseURLLength;      // 许可证获取 URL 长度
      char                LicenseURL;            // 许可证获取 URL
   } else {
      ByteArray           LicenseData;           // 许可证 （语法参见 DREL License）
      ByteArray           ResourceData;          // 语法参见 DREL Resource。一些可以
                                                 // 包含播放设备直接采集到的环境信息
                                                 //（如网络参数，硬件参数等）
```

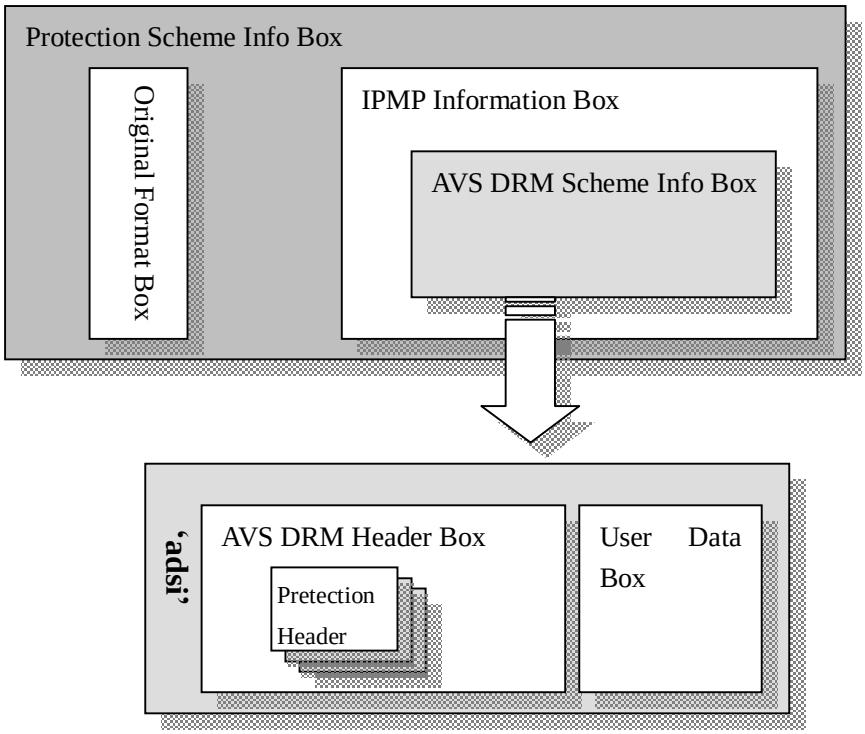
```
    }
}
```

ContentID: 用户用来绑定许可证的全局唯一内容标识。

LicenseURL：用户用来获得许可证的 URL 地址。

6.4.2.2.3. Protection Scheme Info Box

所有与视频/音频内容保护相关的数据都必须定义在 Video/Audio Track 的 ProtectSchemeInfo 盒中，包括：原始视频/音频内容的格式信息、内容保护方案及参数等。ProtectSchemeInfo 属于容器盒，其中含有 OriginalFormat 盒、IPMPInfo 盒。



6.4.2.2.3.1. Original Format Box

```
aligned(8) class OriginalFormatBox(codingname) extends Box ('frma') {
   unsigned int(32)     data_format = codingname;   // 加密前编码数据的原始格式
```

data_format 为原始码流格式的 4CC 表示，对于满足本规范的文件，取值参见 GB/T 20090.9—xxxx。

6.4.2.2.3.2. IPMP Information Box

IPMP Information Box 属于容器盒，用于存放 AVS DRM 保护方案有关的所有数据。本部分规范规定该盒中扩展定义一个 AVSDRMSchemeInfo 盒，其中定义相关的 AVS DRM 数据。

```
aligned (8) class IPMPInfoBox extends FullBox('imif', 0, 0){
   IPMP_Descriptor              ipmp_desc[];
   AVSDRMSchemeInfoBox     SchemeInfo;
}
```

ipmp_desc[]是一组 IPMP 描述子。

6.4.2.2.3.2.1. AVS DRM Scheme Information Box

```
aligned(8) class AVSDRMSchemeInfoBox extends FullBox ('adsi',0,flags){
```

```
    AVSDRMHeader    drmHeader;
    If ( flags & 0x000001 ){
        UserDataBox        UserData;
    }
}
```

AVSDRMHeader：DRM 相关数据及属性。其中包括获得许可证的 URL、对内容采取的保护方案、加密算法等必须信息及其它扩展信息。

UserDataBox：内容相关数据及属性。包括内容作者、主题、版权声明等。UserDataBox 为可选盒，当需要该盒时，flags 值应置为 1。

6.4.2.2.3.2.1.1. AVS DRM Header Box

AVSDRMHeader 盒在 AVS 内容打包中必须出现，提供 DRM 的必要信息，如内容保护方式等。此外还提供了可扩展的机制，用于实现附加功能。

```
aligned (8) class AVSDRMHeader extends FullBox (‘adhd’, version, flag )   {
    unsigned int (8)     ProtectionBoxNum;          // 内容保护方案盒的个数
    ProtectionHeader     prtHeader[];               // 内容保护方案盒
    AVSDRMFormatBox          AVSDRMSmpFormat;       // Sample 格式盒
    Box        ExtendedHeaders[];                   // 扩展盒
}


aligned (8) ProtectionHeader extends FullBox (‘prth’, 0, flags) {
    unsigned int(8)      ProtectionType;            // 内容保护类型，如：加密、水印等
    unsigned int (8)     ProtectionMethod;          // 内容保护方法
    unsigned int (16)    ParameterLength;           // 说明性参数的字节数
    char                 Parameter[];               // 内容保护方法的说明性参数
}
```

6.4.2.2.3.2.1.1.1. ProtectionHeader

为提供可扩展性和兼容性，定义了 Protection Header 盒用于包含具体的内容保护信息，各字段具体语义如下：

ProtectionType：内容保护的类型。虽然当前的 AVS DRM 规范中只对内容使用了加密保护，但该字段为未来新增的保护类型预留了空间。ProtectionType 的取值见下表：

Protection Type	取值	语义
未保护	0x00	未对内容进行保护。这种情况下，用户不需要获得相应的许可证，即可使用内容。
加密方式 1	0x01	以样本为单位加密
加密方式 2	0x02	以 NAL 单元为单位加密
其它	0x02~0xFF	为新增保护类型（如水印等）预留空间

ProtectionMethod：在 Protection Type 所指定的类型下，具体的保护方法。以加密为例，该字

段可以表示采用的加密算法，取值参见本规范核心档。

Parameter[]：用于包含内容保护方法的参数信息，这是指除核心档关于密码算法和保护模式规定的有关参数外，需要扩展的其他参数。这些参数信息以“名称-取值”对的形式出现，名称和取值之间用‘:’隔开，每个名称-取值对以字符‘\0’结束。规定名称中不得含有‘:’字符，以免被解析器错认为分隔符。取值中可以包含‘:’，因为解析器总是以‘\0’为取值结束的判断标识。Paremeter[]的长度由 ParameterLength 确定或一直延伸至所在 Protection Header 盒的末尾。

6.4.2.2.3.2.1.1.2. AVSDRMFormatBox

```
aligned(8) class AVSDRMFormatBox extends FullBox('samp', 0, 0) {
   bit(5)      EncryptionMode;
   bit(1)      FixedKeyParameters;
   bit(2)      reserved;
   unsigned int(8)      KeyID;
   unsigned int(8)      SaltKeyLength;
   unsigned int(8)      IVLength;
   if(FixedKeyParameters = = 1){
        unsigned int(8 * IVLength)          IVValue;
        unsigned int(8 * SaltKeyLength)   SaltKey;
   }
}
```

EncryptionMode 的取值见本规范核心档定义；

FixedKeyParameters 用来表明是否在本轨道中所有的 NAL 单元采用同样固定的密钥参数，如果固定，其后的 IVValue 和 SalkKey 对于所有 NAL 单元来说都是一样的，这样就不需要每一 NAL 单元都提供该信息。

6.4.2.2.3.2.1.1.3. Extended Headers

扩展盒放在 AVS DRM Header Box 的末尾，可用于定义一些附加功能。

6.4.2.2.3.2.1.2. User Data Box

User Data Box 是一个容器盒(container box)，详细定义参照 ISO 14496-12，用于存放说明性的用户数据。这些用户数据以子盒（sub-box）的形式出现，每个子盒明确地定义了各自的功能。

```
aligned (8) class UserDataBox extends Box(‘udta’)  {
}
```

本盒可包含版权信息、标题（Title）、主题（Theme）、作者（Author）、ContentLevel（内容分级）、图标地址（IconURL）、说明信息地址（InfoURL）等。

6.4.2.2.4. 自由空间盒

自由空间盒(free space box)是 ISO 定义的一个可选盒，可以包含任何类型的数据。

```
aligned (8) class FreeSpaceBox extends Box(free-type){
   unsigned int(8)      data[];
}
```

free-type 的取值为’free’或’skip’。

AVS DRM 不保证自由空间盒的完整性，因此在计算 AVS 内容杂凑值时，应跳过该盒。

6.4.2.3. 媒体数据的保护

本小节定义媒体数据的加解密方法。

AVS 文件格式（GB/T 20090.9-xxxx）定义了封装 AVS 音视频内容的方法和格式，它基于并兼容 ISO 基媒体文件格式(ISO base media file format，ISO/IEC 14496-12 | 15444-12)。

NAL 单元保护模式建立在 AVS 文件格式的基础上，这种方式下保护对象是 NAL 单元的净载。

一个样本中的各 NAL 单元的加解密相互独立，共用该样本携带的密钥和初始向量。

受保护的 NAL 单元的结构如下：

	ß－－－－－－－－－－－－ NAL 净载－－－－－－－－－－－－à	
NAL	AVSDRMNALHeader	加密数据

AVSDRMNALHeader 的结构见如下：

```
aligned(8) class AVSDRMNALHeader {
   bit(1)      IsEncrypted ;
      bit(1)     FixedKeyFlag;
   bit(6)      reserved;
   if (IsEncrypted = = 1 ) {
        if (FixedKeyFlag = = 0) {          // 对未固定密钥参数的 NAL，进一步读取 IVValue
                                           // 和 SaltKey
            unsigned int(8 * IV_length)        IVValue;
            unsigned int(8 * SaltKeyLength)   SalkKey;
        }else{                  //从 AVSDRMFormatBox 中获取 IVValue 和 SalkKey 值
        }
   }
}
```

如果 AVSDRMFormatBox 中的 EncryptionMode 取值 0，IsEncrypted 一定设为 0。

FixedKeyFlag 指示本 NAL 是否自带密钥参数，应等于 AVSDRMFormatBox 中的 FixedKeyParameters。对于加密的样本，如果 FixedKeyFlag 为 0，进一步读取 IVValue 和 SaltKey。否则可直接从 AVSDRMFormatBox 中获取 IVValue 和 SalkKey 值。

6.4.2.4. 文件的流化

本部分定义的文件可按照 GB/T 20090.8-xxxx 进行 RTP 流化。

SDP 会话信令同 6.4.1.2。

7. 广播档

7.1. 参考模型

AVS DRM 广播档参考模型如下：

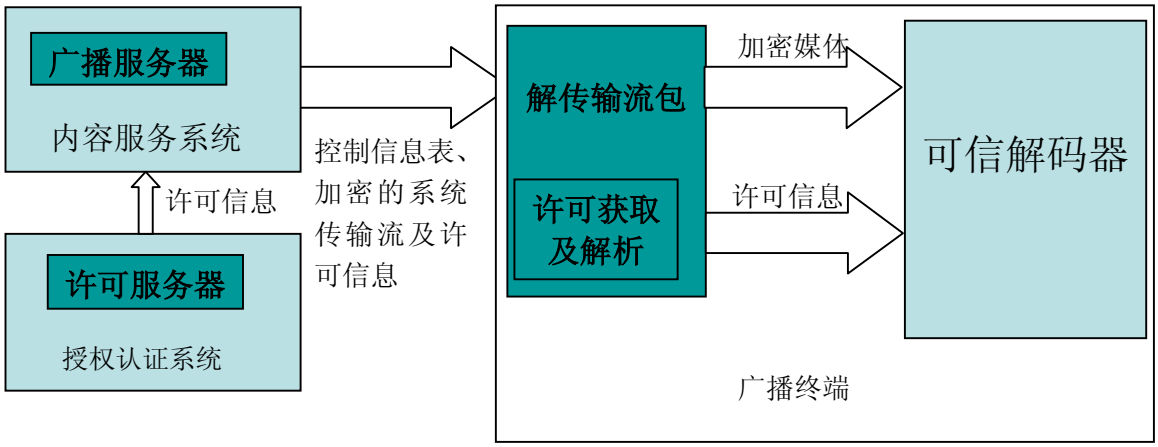


图 15　AVS DRM 广播档参考模型

7.1.1.　所需技术规范

为满足上述体系结构中各模块的实现，本标准草案对广播环境下权利描述语言及 数字广播媒体内容打包方案等技术做了进一步的定义（见本草案第 7.2，7.3 节）。其中对于权利描述语言部分中所使用的基于 XML 的许可证以及在 AVS DRM 系统协议机制中所使用的基于 XML 的消息，本标准推荐使用二进制的信息格式以减少 AVS 解码器所需的负荷处理。

7.2. 对 AVS DREL 数字权利描述语言的采用及限制

广播档采用 AVS DREL 数字权利描述语言（参见第 8 章），但是必须满足以下限制。

通过对权利和约束的组合可以实现资源无限制复制，复制一次，不能再复制，不可复制以及有限时间，有限次数的播放。

7.2.1.　权利接受者

不支持：权利接受者组。

7.2.2.　权利

普通权利中的使用类型只支持：输出、播放。

管理类型只支持：复制。

7.2.3.　约束

时间的约束中只支持：次数，时间段

7.2.4.　义务

不对义务元素进行定义

7.3. 内容打包格式

以下分别阐述了在内容被保护情况下进行数字媒体打包的格式，包括数字媒体在 AVS 系统层上传输时的内容打包格式。

7.3.1.　在 AVS 系统传输流上的控制信息表及保护标记

AVS 广播系统标准（参见 AVS 系统规范［GB/T 20090.1—xxxx］）给出系统层编码规范。它主要被设计用于支持把本标准的音视频编码数据组合起来并支持以下五个基本功能：解码时多条压缩流的同步；多条压缩流交织为一个单一流；为启动解码而对缓冲区进行初始化；连续的缓冲区管理；时间标识。该部分不支持对所传输数字内容的版权保护。本部分在 AVS 广播系统基础上，定义了对传输的数字内容实施保护的机理以及在传输流上对该保护的标记。

7.3.1.1.　AVS 数字媒体版权保护控制信息表

AVS DRM 控制信息表包括有基本的 DRM 权利容器，DRM 单元容器以及其它一些 DRM

控制信息。在内容流中也可使用 DRM 单元容器来传输二进制 DRM 单元。与受保护的内容相关联的数字权利描述（如使用规则等）将放在 DRM 权利容器。整个 AVS DRM 控制信息表可以选择性的被签名以保证它的完整性。

在传输流上，它以 AVS DRM 控制信息表的形式存在。AVS DRM 控制信息表是一个特别定义的用于展现节目所必要的保护信息表，类似于其他携带于传输流中的 PSI 表 (如节目关联表,节目映射表,条件访问表，网络信息表等)。如果在系统流中的任何部分使用 AVS DRM 安全保护标准, AVS DRM 控制信息表应该存在。

7.3.1.1.1. 在传输流上的 AVS DRM 控制信息

AVS 系统传输流包以一个 4 字节前缀开始，内含一个 13 位的包标识 (PID)。PID 通过节目特定信息 (PSI) 表指定包含在传输流包中的数据内容。具有相同 PID 值的传输流包仅携带来自同一个基本流的数据。 AVS DRM 控制信息表被放置在 PSI. 分配给 AVS DRM 控制信息表的 GB/T 20090.1－ xxxx 包标识符 PID 为 0x03. 在被插入传输流包前，AVS DRM 控制信息表可根据以下的语法分割成为一个或多个段。

7.3.1.1.2. 在传输流上的 AVS DRM 控制信息段语法

语法	位数	助记符
AVSDRM_Control_Info_section() {		
table_id	8	uimsbf
Section_syntax_indicator	1	Bslbf
'0'	1	Bslbf
Reserved	2	Bslbf
Section_length	12	uimsbf
Reserved	2	bslbf
AVSDRM_control_info_version	5	uimsbf
Current_next_indicator	1	bslbf
Section_number	8	uimsbf
last_section_number	8	uimsbf
AVSDRM_control_info_classes_total_len	16	uimsbf
AVSDRM_control_info_classes_len_in_this_sect	16	uimsbf
For (i=0; i<N;i++) {		
AVSDRM_control_info_class()		
}		
If (section_number==last_section_number) {		
IsSigned	1	bslbf
Reserved	7	bslbf
If (isSigned)		
Signature		ByteArray
NumCerts	8	uimsbf
for (i=0; i<numCerts;i++) {		
CertType	8	uimsbf
Certificate		ByteArray
}		
}		
}		
CRC_32	32	rpchof
}		

7.3.1.1.3. 语义定义

表标识字段（table_id）： 8 位字段，值为 0x07.

段语法指示符字段（section_syntax_indicator）：1 位字段，值为'1'.

段长度字段（section_length）：12 位字段，规定紧跟在该字段之后且包括 CRC 的该段的

字节数目。其值不能超过 4093。

AVS DRM 版本号字段（AVSDRM_control_info_version）：5 位字段。它是整个 AVS DRM 控制信息表的版本号码。一旦 AVS DRM 控制信息表发生变化，该字段将递增 1，并对 32 取模。在 current_next_indicator 为'1'时，版本号应该是当前适用的 AVS DRM 控制信息表的版本号；在 current_next_indicator 为'0'时，版本号应该是下一个适用的 AVS DRM 控制信息表的版本号。

当前下一个指示符字段（current_next_indicator）：1 位指示符。置'1'时表示传送的 AVS DRM 控制信息表是当前适用的；置'0'时表示传送的 AVS DRM 控制信息表还不适用，它是下一个生效的表。

段号字段（section_number）：8 位字段，给出了该段的号码。AVS DRM 控制信息表中第一个段的 section_number 应为 0x00。AVS DRM 控制信息表每增加一个段，它将递增 1，并对 256 取模。

末段号字段（last_section_number）：8 位字段，给出整个 AVS DRM 控制信息表中最后一段(即有最高 section_number)的号码。

AVS DRM 控制信息类总长字段（AVSDRM_Control_Info_classes_total_len）：16 位字段，给出 AVS DRM 控制信息类的字节总长，该类可被携带于超过一个 AVS DRM 控制信息段上。

AVS DRM 控制信息类在本段的字节长（AVSDRM_Control_Info_classes_len_in_this_section）：16 位字段，给出 AVS DRM 控制信息在本段中紧接着本字段的字节长。

AVS DRM 控制信息类字段（AVSDRM_Control_Info_class）：AVS DRM 控制信息类包括 AVS DRM 单元容器，权利容器等。因为它的长度可能超过 4093，因此它可以被拆分成多个 AVS DRM 控制信息段。当 AVS DRM 控制信息段从完全的 AVS DRM 控制信息表中被重新组合时，AVS DRM 控制信息类通过 AVSDRM_Control_Info_classes_total_len 合理地组合回来。

是否签名字段（isSigned）：1 位字段表明是否对 AVS DRM 控制信息表进行签名。

签名（Signature）：整个 AVS DRM 控制信息类的签名。

证书类型（CertType）：使用的证书类型，它的取值不在本部分范围。

证书数目（NumCerts）：包括的证书数目。

证书（Certificate）：证书链。

CRC 32 字段（CRC_32）：32 位字段，它包含 CRC 值以在处理完整个 AVS DRM 控制信息段后在 ISO/IEC 13818-1 附录 B 中定义的解码器寄存器产生 0 输出值。

7.3.1.1.4. AVS DRM 控制信息类

AVS DRM 控制信息包括以下使用不同标签的 AVS DRM 控制信息类。

AVS DRM 控制信息类标签	类名
0x00	Forbidden
0x01	AVSDRM_Module_Container Class
0x02	AVSDRM_Rights_Container Class
0x03	AVSDRM_VariantKey_Container class
0x04-0x05	AVSDRM Reserved
0x06-0xC0	Reserved
0xC1-0xFE	User private
0xFF	Forbidden

7.3.1.1.4.1. AVS DRM 单元容器类

AVS DRM 单元容器类包含为获取 DRM 单元所需描述信息,所有这些信息提供了识别某特定加载 DRM 单元的来源和目的地所需的方式。该容器也可直接用来传输二进制 DRM 单元本身。DRM 保护单元中的对称加密算法应采用核心档中定义的加密算法。

表 X AVS DRM 单元容器类中字段语法定义

语法	位数	助记符
AVSDRM_Module_Container() {		
Control_info_class_tag	8	uimsbf
Length	16	uimsbf
Flag	1	bslbf
ManufacturerID	8	uimsbf
Reboot	1	bslbf
Add_on	1	bslbf
Download_id	32	uimsbf
Device_No	8	uimsbf
Group_No	8	uimsbf
ModuleVerNo	8	uimsbf
DRMModuleBody		ByteArra
}		

说明：

控制信息类标签（Control_info_class_tag）：8 位字段，值设置为 0x01 表明这是 AVS DRM 权利容器。

长度（Length）：紧接着本字段之后 AVS DRM 单元容器类的字节数。

标记位 (Flag)：1 位字段，该位为“1”表明携带的 DRM 单元可更新，为“0”表明不可更新。

厂商标识符（ManufacturerID）：该字段表征实际 DRM 单元生产厂商的标识。

重新启动 （Reboot）：1 位字段，该字段表明实际加载 DRM 单元是否需要重新启动用户设备，该位为“1”表明需重启，为“0”表明不需重启(继续)。

附加（Add_on）：1 位字段，该字段表明实际加载 DRM 单元是否追加或取代原有存在的 DRM 单元，该位为“1”表明为追加模式，为“0”表明取代模式。

下载标识符（Download_id）：该字段表征下载 DRM 单元的标识以区别原有的或将来可能发生的 DRM 单元下载。

设备编号（Device_No.）：8 位字段，只有当与设备编号相匹配的设备才下载这个 DRM 单元。

组编号（Group_No.）：8 位字段，DRM 单元所首选的接受组或域编号。

DRM 单元版本号（Module version No.）：8 位字段，表明具体 DRM 单元的版本号。

DRM 单元体（DRMModuleBody）：该字段包含 DRM 二进制单元的具体数据。

7.3.1.1.4.2. AVS DRM 权利容器类

AVS DRM 权利容器类存放有与 AVS DRM 保护内容相关的使用规则和状态。

表 X AVS DRM 权利容器类中字段语法定义

语法	位数	助记符
AVSDRM Rights Container ()		
Control_info_class_tag	8	uimsbf
Length	16	uimsbf
REL_ID	2	uimsbf
REL_Format	1	uimsbf
RightsData		ByteArray
}		

说明：

控制信息类标签（Control_info_class_tag）：8 位字段，值设置为 0x02 表明这是 AVS DRM 权利容器。

长度（Length）：紧接着本字段之后 AVS DRM 权利容器类的字节数。

权利描述语言标识（REL_ID）：2 位字段，该字段用来识别在 DRM 系统中实际使用的权利描述语言，例如 AVS 的数字权利描述语言 DREL 或其他数字权利描述语言。

语言格式（Language_Format）：1 位字段，该字段用来识别实际携带权利描述语言的格式是二进制或可扩展标记语言 XML，该位为“1”表明实际携带权利描述语言的格式是二进制，为“0”表明实际携带权利描述语言的格式是可扩展标记语言 XML。

权利数据（rights_data）：该字段包含具体的权利信息，即许可证。

7.3.1.1.4.3. 可变密钥容器类

AVS DRM 可变密钥容器类存放有广播流加密密钥。

表 X AVS DRM 可变密钥容器类中字段语法定义

语法	位数	助记符
AVSDRM_VariantKey_Container () {		
Control_info_class_tag	8	uimsbf
Length	16	uimsbf
VariantKeyData		ByteArra
}		

说明：

可变密钥数据（VariantKeyData）：广播流加密密钥数据。这里的密钥数据是指辅助密钥，该辅助密钥与 AVS DRM 权利容器类中许可证中的主密钥按照 5.7.1 中定义的密钥推导算法生成内容密钥。该内容密钥用于媒体数据的解密。

7.3.1.2. AVS DRM 保护标记

在传输流的节目映射表 (PMT) 中嵌入 AVS DRM 描述子来完整的表述 AVS DRM 保护计划。AVS DRM 描述子在 7.3.1.2.2 节中有详述。

7.3.1.2.1. 在传输流中 AVS DRM 保护标记

下表是在 GB/T 20090.1—xxxx 系统中的传输流节目映射段（参见 GB/T 20090.1—xxxx 表 2-28）。

表 X 传输流节目映射段语法定义

语法	位数	助记符
TS program map section() {		
table_id	8	Uimsbf
section_syntax_indicator	1	Bslbf
'0'	1	Bslbf
Reserved	2	Bslbf
section_length	12	Uimsbf
program_number	16	Uimsbf
Reserved	2	Bslbf
version_number	5	Uimsbf
current_next_indicator	1	Bslbf
section_number	8	Uimsbf
last_section_number	8	Uimsbf

Reserved	3	Bslbf
PCR_PID	13	uimsbf
Reserved	4	bslbf
program_info_length	12	uimsbf
for (i=0; i<N; i++) {		
descriptor()		
}		
for (i=0;i<N1;i++) {		
stream_type	8	uimsbf
Reserved	3	bslbf
elementary_PID	13	uimsnf
Reserved	4	bslbf
ES_info_length	12	uimsbf
for (i=0; i<N2; i++) {		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

如果 AVS DRM 描述子存在于 PMT 中的基本流循环外的 descriptor( )，表明在 AVS DRM 描述子中描述的信息应用于该节目中的所有被保护的基本流。如果 AVS DRM 描述子存在于 PMT 中的基本流循环内的，与单独一个基本流相联的 descriptor( )，表明在 AVS DRM 描述子中描述的信息只应用于该特别的基本流。

在 GB/T 20090.1—xxxx 表 2-29 中加入 AVS DRM 流类型:其值为 Ox1A 并且调整保留值范围。

7.3.1.2.2. AVS DRM 描述子

在传输流 PMT 中 AVS DRM 描述子的存在表明在 AVS DRM 描述子中描述的信息与被保护的基本流的相关联系。AVS DRM 数据（AVSDRM_Data）提供了基本流如何被保护等的信息如使用规则，加密参数等等。

在 GB/T 20090.1—xxxx 表 2-39 中加入 AVS DRM 描述子标签:其值为 41 并且调整保留值范围。

7.3.1.2.2.1. AVS DRM 描述子的语法

表 X AVS DRM 描述子语法定义

语法	位数	助记符
AVSDRM descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
AVSDRM_Descriptor_ID	32	uimsbf
AVSDRM_Data_length	16	uimsbf
for (i=0; i< N; i++) {		
AVSDRM_Data		
}		
isSigned	8	uimsbf
if (isSigned)		
Signature		ByteArray
NumCerts	8	uimsbf
for (i=0; i<numCerts;i++) {		
CertType	8	uimsbf

```
                    Certificate                                    ByteArray
                }
            }
        }
```

说明：

AVS DRM 描述子标识字段 （AVSDRM_Descriptor_ID）：AVS DRM 描述子独一无二的标识。它可被用作参照这个特别的描述子。0x00000000 和 0xFFFFFFFF 禁止使用。

AVS DRM 数据字段（AVSDRM_Data）：AVS DRM 数据描述保护机理以及特别的使用规则。

是否签名字段（isSigned）：1 位字段表明是否对 AVS　DRM 描述子进行签名。

签名（Signature）：整个 AVS　DRM 描述子的签名。

证书类型（CertType）：使用的证书类型，它的取值不在本部分范围。

证书数目（NumCerts）：包括的证书数目。

证书（Certificate）：证书序列。

7.3.2. 媒体内容保护

根据 GB/T 20090.1 定义的传输流（TS 流）适合于广播应用。对内容的保护，既可以针对 TS 流，也可以针对 PES 流，在 TS 包头和 PES 包的包头均有专门的字段定义。

不允许在 TS 流和 PES 流同时加绕。

7.3.2.1. TS 层的保护

TS 包头中表示保护方式的字段为 **transport_scrambling_control**，是一个 2 位字段，指出传输流包有效负载数据的加扰方式。传输流包头和适应字段不应该被加扰。对空的分组而言，该字段值应设定为'00'。此字段取值为 00 表示未加扰，即明文状态，其它三个取值由用户定义。这里取值'11'表示秘文状态，其它取值保留。

表 X　TS 包加扰控制值

值	描　述
00	明文状态
01	用户定义
10	用户定义
11	加密状态

7.3.2.2. PES 层的保护

PES 包的 PES_scrambling_control 字段表示 PES 分组包有效负载的加扰方式。当加扰发生在 PES 层，PES 分组包头—如果有可选字段的话也包括在内—不应被加扰。PES_scrambling_control 取值规定如下:

表 X　GB/T 20090.1 规定的 PES 加扰控制值

值	描　述
00	非加扰
01	AVSEncryptionMode=1
10	AVSEncryptionMode=2

11	AVSEncryptionMode=4（若负载为视频）或5（若负载为音频）

7.3.3. **广播档与原有 CA 系统的关系 （参考性资料）**

原有的基于 AVS 系统（GB/T 200090.1）的私有 CA 系统也可用于保护以传输流传输的数字广播内容。它提供以下的功能：对特别保护传输流包的标记，ECM/EMM 的信息传输，以及表明使用的特殊 CA 系统。

尽管 CA 系统可以与 AVS 音视频流集成以达到保护的目的，但并没有规定不同 CA 系统之间的互操作性问题。而且众所周知，CA 系统并不灵活，也就是无法应用多个 CA 系统来保护单一传输流。而上述定义的广播档通过 AVSDRM 控制信息表、不同的控制数据类（也就是 AVSDRM 单元容器，权利容器，可变密钥容器）可在一定程度上提供互操作性及灵活性。

广播档可与原有的 CA 系统反向，前向兼容并共存。

7.3.3.1. 反向兼容

为达致 CA 系统识别广播档中定义的保护信息的存在，广播档中定义的保护信息必须被 CA 系统识别，也就是该信息必须在 CA 系统中被标记。这样支持广播档的设备可以在 CA 系统标记表中选择它所需的保护信息与数据。

那些完全用原有 CA 系统设计的设备只可以确定得到的传输流被一种保护机理所保护并通知用户，但不能完全解析该保护机理，并解密内容。注：该设备并不会死机。

7.3.3.2. 前向兼容

如果一个已经设计好的 CA 系统希望使用广播中定义的原理进行传输并标记保护信息，该 CA 系统可定义为一个特别的 AVSDRM 单元（AVSDRMModule）并将相关的信息包含与 AVSDRM 控制信息表中。另外该 CA 系统所拥有的特别信息以及 ECM/EMM 信息都可以在与用该 CA 系统保护子流的 AVSDRM 描述子中给予定义和描述。

这样，实现广播档的设备也能够完全解析该 CA 系统相关的标记信息，并解密用该 CA 系统进行保护的内容。

7.3.3.3. 共存

广播档与原有的 CA 系统共存是可行的，这两个系统保护可以独立或互相合作以达到保护内容的目的。也就是说，混合的传输流中将包含有 AVSDRM 描述子、CA 描述子、ECMs、EMMs、AVSDRM 数据、AVSDRM 控制信息表、CA 条件访问表。处理这种混合的传输流要求终端设备必须实现广播档及原有的 CA 系统。设计这类设备是可行的，但不在本标准范围内。

8. AVS 数字权利描述语言

8.1. 概述

AVS 数字权利描述语言（简称 AVS-DREL）定义了开放和可信任环境中进行数字版权管理的语法和语义规则，旨在为数字音视频资源的出版、交易、分配和消费等的操作提供一种具有灵活性和互操作性的权利描述机制，通过这种机制描述受保护的数字资源、使用权利和主体之间的关系。

AVS-DREL 定义的语法和语义规则用于：1）描述数字版权所涉及的要素；2）描述主体间的授权关系；3）描述对内容访问或使用的控制机制。

AVS-DREL 基于本部分核心档所定义的密码算法和安全协议，实现身份认证，维护内容机密性和完整性，支持隐私保护。

AVS-DREL 采用 XML 绑定的形式，支持机器读取和执行。

AVS-DREL 分三个部分，分别是信息模型、数据字典和 XML 绑定。信息模型对数字权利描述语言的语法结构和框架模型进行描述；数据字典给出了 AVS-DREL 语法框架下的元素语义定义；XML 绑定作为附录，描述了如何以 XML 文档的形式对数字权利进行语法描述，通过 XML

Schema 即可对 XML 文档的正确性进行验证。

文档的附录也给出了 AVS-DREL 实践指南，列举了 AVS-DREL 的几种实践应用样例。

8.2. AVS 数字权利描述语言信息模型

AVS-DREL 信息模型定义的许可证结构由主体、权利、资源、约束、义务等五类元素组成。

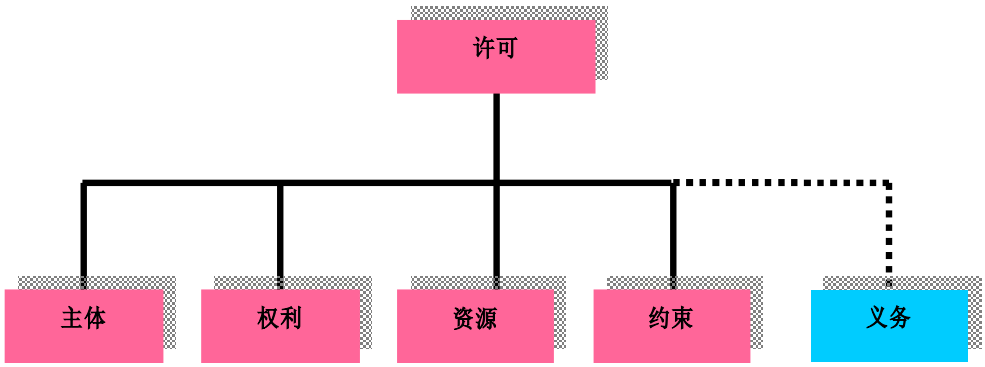


图 16 AVS-DREL 信息模型概要结构图

图 16 表示 AVS-DREL 信息模型的概要结构，它用于描述许可认证的基本要素，并不表示各元素之间的层次关系。此概要模型包括：

(1) 许可：一个包含了权利发布者向权利接受者授权的描述集合的基本容器

(2) 主体：一种强制性的对象元素，也是抽象类型元素，它本身不代表任何具体的实体元素，针对价值传递链中主体所处的不同位置，主体分为权利发布者和权利接受者；虽然权利发布者和权利接受者的应用形式及其在许可证中所处的结构位置不同，但都属于主体的范畴；当存在多个主体共同行使某一权利时，可通过主体组对该主体集合进行定义

(3) 资源：包括视频、音频等数字资源

(4) 权利：泛指权利接受者对资源所拥有的操作

(5) 约束：定义了权利接受者对资源使用相应权利时应满足的条件

(6) 义务：权利接受者在行使一定权利的同时承担的各种责任，义务是可选项

8.3 节将对 AVS-DREL 的信息模型的语义进行详细描述。

8.3. 权利描述语言模型

8.3.1. 定义和结构

数字权利描述语言基本单位为一个许可证，即某一个权利发布者对另一个权利接受者发布的针对某项数字资源使用权利的授权声明。

许可证包含以下含义：

(1) 版权拥有者和分销商之间的权利授权声明

(2) 分销商与用户之间的权利授权声明

(3) 用户与用户之间的权利授权声明

通过对许可证中不同权利的定义，不同权利发布者和权利接受者可以实现以上三种权利授权声明。

许可证由以下三部分组成：

(1) 权利发布者：描述权利发布者的基本信息；许可证必须包括权利发布者

(2) 许可单元：许可证的基本单元。许可单元在许可证中是必须出现的，但是出现的个数没有限制。在同一个许可证下的许可单元必须具有相同的权利接受者。许可单元针对个资源向权利接受者发布某种权利，依次包含主体，权利，资源，约束和义务 5 种元

素。许可单元与权利接受者、权利和资源是一一对应的，但是许可单元可以对定义的权利设置多个约束条件和义务条件

(3) 许可证签名：用于保证许可证的完整性和不可否认性

许可证基本结构如图所示：

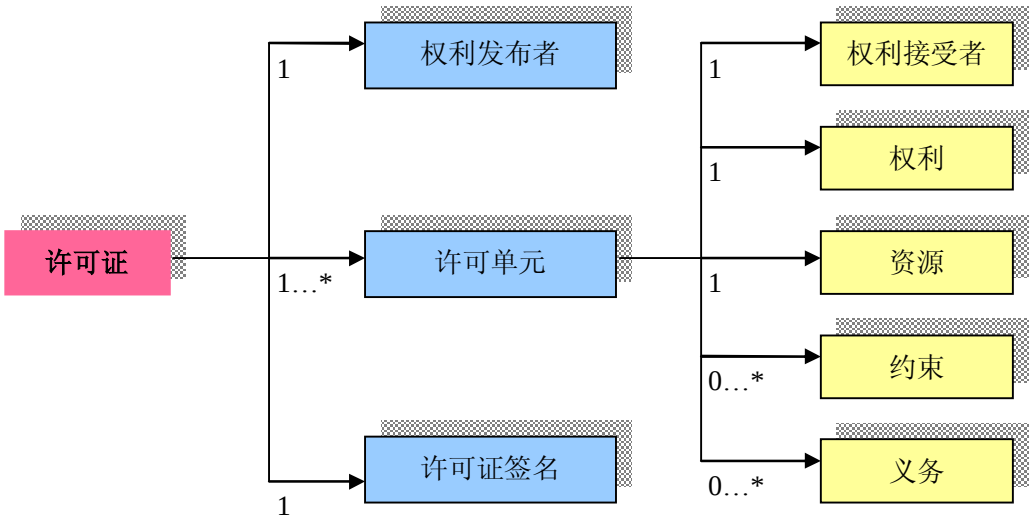


图 17　许可证模型结构图

8.3.2.　**标识**

为了能够快速和准确的对许可证的子元素进行定位，每一个许可证元素都需要定义一个唯一标识，包括：

(1) licenseID：许可证唯一标识，用于许可证定位和存储

(2) elementID：每一个许可证中的元素都用一个 ID 标识，便于定位和存储

(3) elementIDRef：许可证子元素之间往往会互相引用，以减少重复定义造成的冗余，引用元素采用 IDRef 进行唯一的标识，与被引用元素的 ID 值相对应

8.3.3.　**属性**

许可证中每个元素都具有共同的特征，这些共同的特征组成了许可证的基本属性，许可证中各部分元素可选择性对下列基本属性进行定义描述。包括：

(1) 名称：用来描述当前元素的名称

(2) 版本：用来指定当前元素的版本

(3) 日期：指定元素发生或有效的日期

(4) 物理地址：当前元素可定位的物理地址

(5) 数字地址：当前元素可定位的数字地址（URI）

(6) 引用：关于当前元素附加信息的一个链接（URI）

(7) 注释：解释、描述等附加信息

(8) 语言：当前元素所采用的语言

说明：在“AVS 数据字典”中详细的定义了以上基本属性。

8.3.4.　**元素**

8.3.4.1.　主体

主体元素包括权利发布者和权利接受者。主体元素定义了不同环境下的角色，可以表示为具体的人、组织、机器设备、应用程序、网络终端或某类角色等。

8.3.4.1.1.　权利发布者

权利发布者指在资源的创作、生产、分发时可以向权利接受者发布权利的实体，包括版权

拥有者、中间商、代理商、分销商以及用户本身等等。权利发布者具有发布许可证、分配资源使用权的权利，也可以是许可证的签发者。

当权利发布者以主体组的形式为单位进行权利发布的时候，可以通过定义权利发布者组对其进行描述。

8.3.4.1.2. 权利接受者

权利接受者指资源的消费者，包括价值传递链中的中间环节或终端用户，对应为各种身份和性质的人，组织或机器设备等。权利接受者仅具有行使资源使用权利的权利，可以是许可证的接受者。

当权利接受者作为一个群体进行接受并使用权利的时候，可以通过定义权利发布者组对其进行描述。

8.3.4.1.3. 主体属性

主体的共同的特征由主体属性元素描述。主体属性元素通过引用元素基本属性中的单元对主体的名称，关于主体附加信息的链接和对主体进行解释和描述的附加信息等信息进行选择性的描述。

8.3.4.2. 权利

在数字权利描述语言中，权利代表对资源或许可的处置方式，使权利接受者在授权认证的环境下，对某种资源或许可本身行使相关的操作，在语义上会与某种资源相关联。

权利是一种强制性的对象元素，也是抽象类型元素，权利元素本身不代表任何具体的操作，权利类型的元素在许可单元中出现且仅出现一次。根据 AVS 视音频标准中的需求，并针对不同的权利发布者和不同的使用环境，权利可以分成普通权利和高级权利，如图：

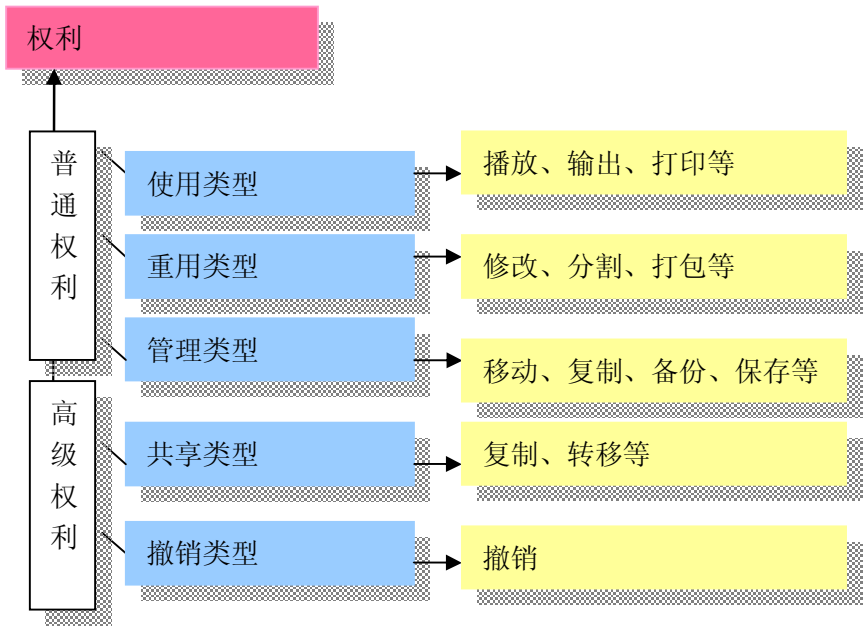


图 18　权利模型结构图

8.3.4.2.1. 普通权利

对应于价值链中的中间环节或最末端的权利接受者，定义允许直接使用和处理资源的行为或动作。共分为三类：使用类型、重用类型和管理类型。

8.3.4.2.1.1. 使用类型

定义使用资源时的动作。有以下三种类型：

(1)　播放：通过解码，在本地设备上观看资源

备注：针对不同的资源内容，与不同的应用相联系，播放会有相应的不同动作。

(2) 输出：先在本地设备解码，随后将码流输出到外部设备

(3) 打印：将本地设备上进行播放的图片资源，在输出设备上进行打印

8.3.4.2.1.2. 重用类型

表示权利接受者获得到资源后，可以再次使用的权利。

(1) 修改：对资源进行一定程度的编辑和修改，形成新的资源

(2) 分割：对资源进行分割，形成新的资源

(3) 打包：对多个资源进行整合，形成新的资源

8.3.4.2.1.3. 管理类型

针对资源的管理操作。

(1) 移动：资源在本地设备进行移动，移动之后的资源可以被直接使用，处于原位置的资源将被删除

(2) 复制：资源在本地设备上进行复制，复制之后的资源可以被直接使用，处于原位置的资源同样可以被直接使用

(3) 备份：防止由于设备或人为的因素导致资源丢失而采取的一种对资源进行存储的手段，备份之后的资源不能被直接使用，而只有在原有的资源被破坏不能使用的时候，经过备份的资源可以通过相应的操作进行恢复，恢复之后的资源才可以被直接使用

(4) 保存：将在线观看的资源存储到本地设备

8.3.4.2.2. 高级权利

定义对现有普通权利的行为或动作。高级权利只有当许可证中已经定义了普通权利之后才能在许可证中被定义。

8.3.4.2.2.1. 共享类型

通过某种过程使权利接受者对资源的普通权利及其相应的约束和义务信息进行复制或转移。

(1) 转移：权利接受者将所具有的对资源的使用权利转移给下一级用户，转移之后原来的权利接受者就不可以使用该权利。该权利接受者在转移了资源的使用权利之后，有权撤销权利接受者所拥有的对该资源的权利

(2) 复制：权利接受者将所具有的对资源的使用权利复制给下一级用户，复制之后原来的权利接受者还可以继续使用权利。该权利接受者在复制资源的使用权利之后，有权撤销权利接受者所拥有的对该资源的权利

8.3.4.2.2.2. 撤销类型

撤销：权利发布者取消权利接受者使用资源的权利

8.3.4.3. 资源

资源：权利发布者可在其上授予权利的视频，音频等数字资源。资源可以是一个数字作品（电子书、音频、视频或图象等），可以是服务（如邮件服务等），也可以是某实体所拥有的信息（如某人电子邮件地址等）。本标准中的资源主要针对 AVS 音视频，资源元素包括资源属性、资源的密钥信息和资源的摘要信息。

资源是一种强制性的对象元素，也是抽象类型元素，资源元素本身不代表任何具体的实体，资源类型的元素在许可单元中出现且仅出现一次。当许可单元中存在多种相关资源时，可通过资源组对该相关资源集合进行定义。

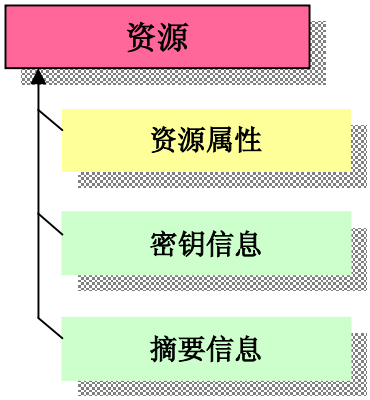


图 19　资源模型结构图

8.3.4.3.1.　资源属性

资源的共同特征通过属性元素描述，资源属性元素通过引用元素基本属性中的单元对资源的名字，解释或描述资源的附加信息或附加信息的链接等进行选择性的描述。

8.3.4.4.　约束

定义权利接受者对资源使用相应权利时应满足的条件。

约束是一种强制性的对象元素，也是抽象类型元素，约束元素本身不代表任何具体的实体，约束类型的元素在许可单元中可以不出现或出现多次。根据 AVS 视音频标准中的需求，针对约束的不同范围，可把约束分成空间、时间等约束类型。当许可单元中存在多种约束时，可采用容器的方式将多种约束加载到容器中，并通过定义逻辑关系属性标明不同约束之间的关系。

约束类型可以分为：

(1)　空间的约束

(2)　范围的约束

(3)　时间的约束

(4)　使用环境的约束

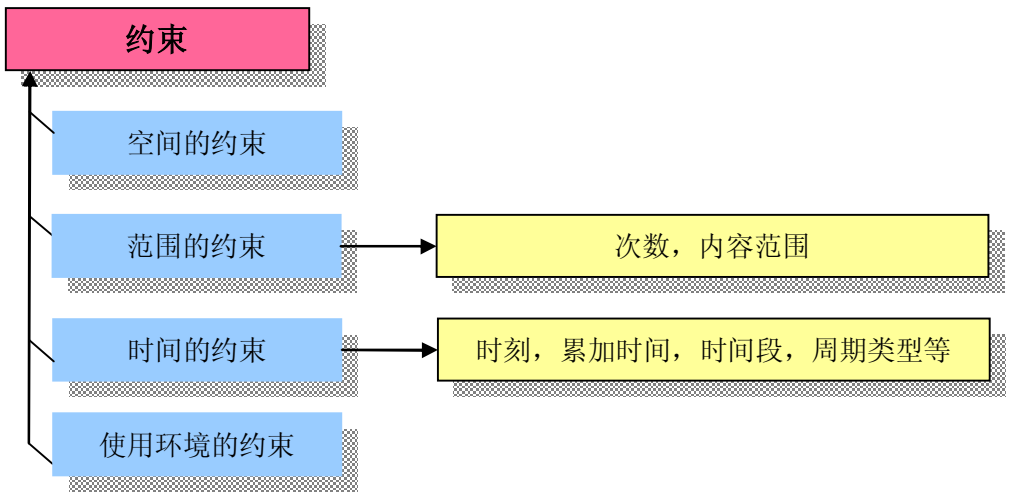


图 20　约束模型结构图

8.3.4.4.1.　空间的约束

实现对地理范围的或者网络划分的限制。例如不同国家、地区或是网络地址段的限制。

8.3.4.4.2.　范围的约束

实现对资源使用范围的约束，可以分为：

(1) 次数：约束执行权利的次数

(2) 内容范围：约束执行权利的资源的范围

8.3.4.4.3. 时间的约束

实现对时间方面的约束限制，可以分为：

(1) 时刻：约束在某一个特定的时刻执行权利

(2) 累加时间：约束总共执行权利的时间

(3) 时间段：约束连续执行权利的一段时间

(4) 周期类型：周期性执行权利的类型

8.3.4.4.4. 使用环境的约束

要求在使用某一项资源的时候必须实现对物理设备或软件系统的约束限制。例如资源的质量品质，资源的表现形式，资源的存储形式，资源的传输形式以及资源的粒度，系统是否具有安全的环境以及系统或其中某单元是否符合相应的版本等约束。

8.3.4.5. 义务

权利接受者在行使一定权利的同时应该承担的责任。

例如权利接受者使用资源，就必须付费，这就是权利接受者的义务。在“义务”项中就可以定义支付方式，例如，使用前、使用后、按次付费等。在义务项中也可以要求在资源中添加版权说明和版权使用说明等说明性信息，以便资源的接受者了解资源使用的情况。

义务是一种非强制性的对象元素，也是抽象类型元素，义务元素本身不代表任何具体的实体，义务类型的元素在许可单元中可不出现。根据 AVS 视音频标准中的需求，针对义务的不同内容，可把义务分成付费方式，交互性要求等义务形式。当许可单元中存在多种义务时，可采用容器的方式将多种义务加载到容器中，并通过定义逻辑关系属性标明不同义务之间的关系。

义务类型可以分为：

(1) 付费方式

(2) 交互性的要求

(3) 对用户的要求

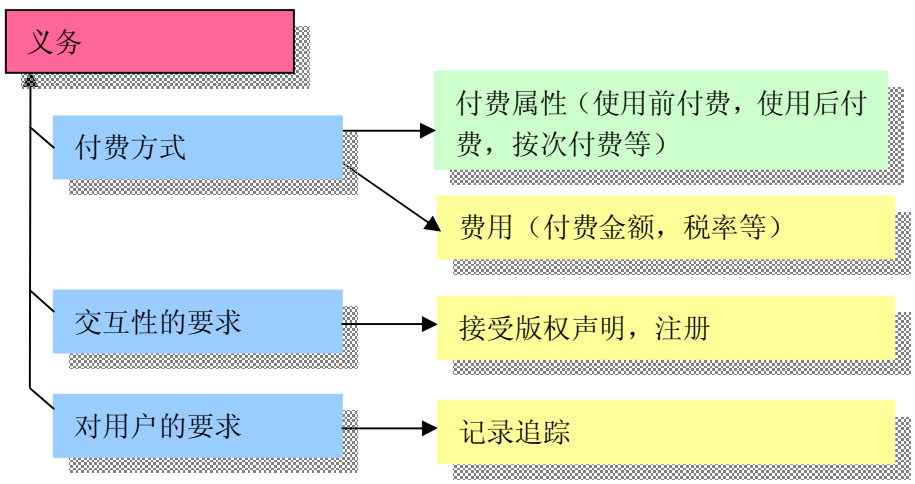


图 21 义务模型结构图

8.3.4.5.1. 付费方式

8.3.4.5.1.1. 付费属性

用户对资源行使权利时，需要支付一定费用。付费属性描述了用户需遵从的付费方式，标识付费义务与权利的关系，具有以下三种付费属性值：

使用前：先行使付费义务，后授予并行使权利

(1) 使用后：先授予并行使权利，后行使付费义务

(2) 按次付费：每次行使权利时都需行使付费义务

8.3.4.5.1.2. 费用

(1) 付费金额：费用额，该元素还包括用来标识付费货币种类的属性

(2) 税率：税率

8.3.4.5.2. 交互性的要求

(1) 接受版权声明：被授权资源权利前，用户必须浏览并且同意版权声明

(2) 注册：授权并行使资源权利前，用户必须通过服务提供商注册自己的详细信息

8.3.4.5.3. 对用户的要求

记录追踪：用户对某资源行使权利后，记录用户对该资源行使权利的行为

8.3.5. AVS-DREL 安全模型

AVS-DREL 采用数字签名保证整个许可证的完整性，并描述许可证中的机密信息的存放和加密方法。在下述 AVS-DREL 安全模型中使用的密码算法应符合核心档中关于密码算法所做的规定，这些密码算法包括：

(1) 摘要算法

(2) 密钥加密算法

(3) 签名算法

8.3.5.1. 资源加密信息描述

AVS-DREL 资源加密信息包括资源摘要及所采用摘要算法、资源加密主控密钥及其加密算法。具体构成如下图：

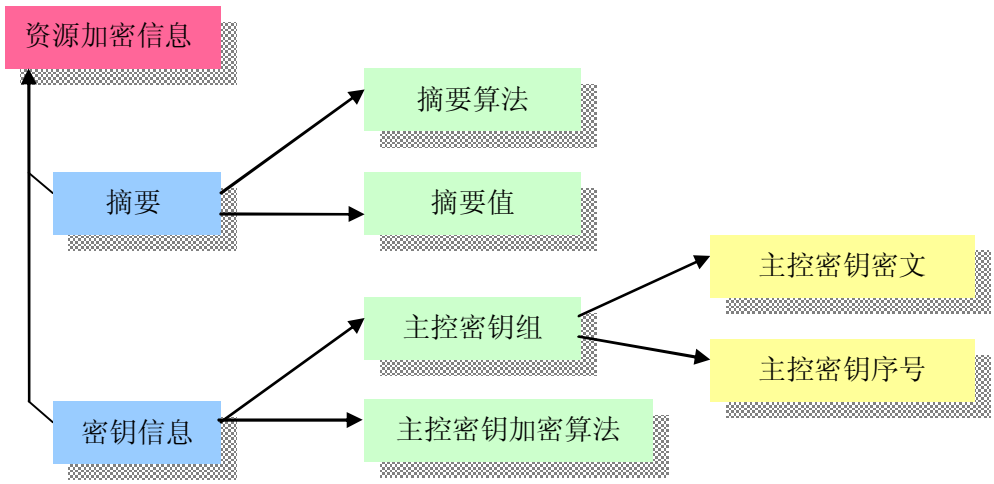


图 22 加密实体结构图

8.3.5.1.1. 摘要

摘要是资源的下属元素。

摘要用来验证相关联的内容的整体性，它包括下列两个元素：

(1) 摘要算法（Digest Method）：用来计算摘要值的算法

(2) 摘要值（Digest Value）：计算出来的内容的摘要的值

8.3.5.1.2. 密钥信息

密钥信息元素是资源的下属元素，它包含：

(1) 主控密钥组（Key Array）：是主控密钥密文及其序号的列表

(2) 主控密钥加密算法（Masterkey Encryption Algorithm）

主控密钥密文是用主控密钥加密算法对主控密钥加密得到的结果。主控密钥序号与主控密钥密文一一对应，可用于指示内容加密时所使用的密钥。当主控密钥组中只包含一个主控密钥时，则表明此主控密钥即是内容加密密钥。

8.3.5.2. 数字签名

数字签名是许可证的下属元素，签名对象为整个许可证的中除本元素外的所有其他数据，采用 W3C XML 数字签名规范[X]。签名所包括的元素如下图：

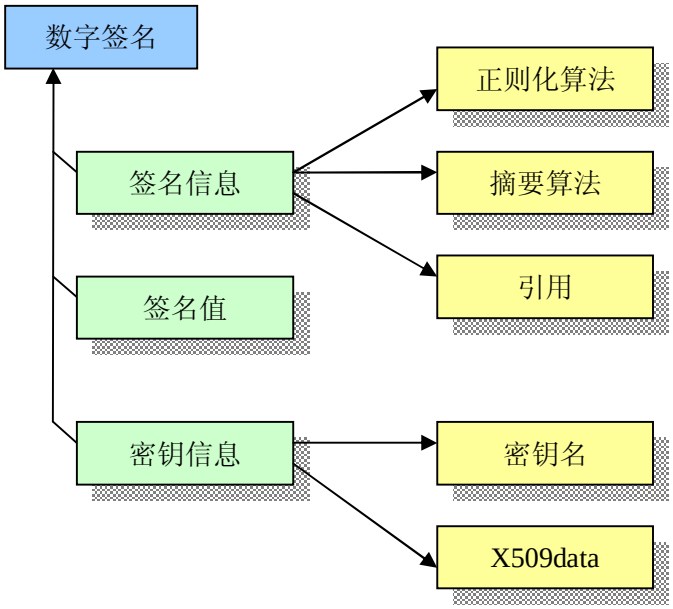


图 23 数字签名实体结构图

8.4. AVS 数字权利描述语言数据字典

本标准为 AVS 权利描述语言的元素定义了数据字典。数据字典用词汇表的方式表达，是一个推荐使用的取值列表，列表支持扩展类型。但是，使用推荐值的元素将具有最大程度的语义互操作性，也就是说，这些元素将最大可能地被别的终端用户所理解。

8.4.1. 数据结构概述

8.4.1.1. 元素的基本结构

权利许可证通过许可证元素进行描述。许可证元素主要由以下五类子元素组成，分别是：

(1) 主体
(2) 权利
(3) 资源
(4) 约束
(5) 义务

其他数据元素通过继承这五类数据元素从而对其性质进行扩展，以实现对许可证的完整描述。

8.4.1.2. 元素属性

数据元素一般有九个属性。数据元素是一个层次结构，包括聚合数据元素和简单数据元素（其中，简单数据元素即层次结构中的叶节点）。只有简单数据元素才有数据类型和值域。对于每个数据元素，其基本框架定义如下：

(1) 名称：数据元素的名称
(2) 标识符：数据元素的唯一标识
(3) 定义：对数据元素的定义
(4) 注释：附加说明，包括举例说明
(5) 大小：数据元素所允许的最大取值个数
(6) 次序：值的排列次序是否有意义（只适用于具有列表值的数据元素，参见 8.4.1.3 列表

值）

(7) 数据类型：描述数据元素所取数值的数据类型（仅简单数据元素才有，参见**错误！未找到引用源。**数据类型）

(8) 值域：数据元素的取值范围——一般以词汇表或者引用另一个标准/规范的形式出现（仅简单数据元素才有）

(9) 约束性：对数据元素约束属性的描述，包括必需性数据元素和可选性数据元素（参见8.4.1.4 约束性），“M”表示必需性数据元素，“O”表示可选性数据元素

8.4.1.3. 列表值

在元数据实例中，某些数据元素的值可以不是一个单一的值，而是一个列表。列表应该是下面两种类型之一。

(1) 有序的：值的先后次序有意义

(2) 无序的：值的先后次序无意义

8.4.1.4. 约束性

数据元素的约束性。约束性有二个层次：

(1) 必需的：在数据结构中定义，并且必须在数据结构的实例中出现的数据元素；“必需”属性属于数据元素的约束属性，用“M”表示

(2) 可选的：在数据结构中定义，但不一定要求在数据结构的实例中出现的数据元素；“可选”属性属于数据元素的约束属性，用“O”表示

8.4.2. 数据元素列表

8.4.2.1. 许可证

编号	名称	标识符	定义	约束	大小	次序	值域	数据类型	注释
1	许可证	license	某一个权利发布者对另一权利接受者发布的针对某项数字资源的使用权利的授权声明	M	1	无			许可证是数字权利描述语言的一个基本单元
1.1	许可单元	licenseUnit	针对某种资源向权利接受者发布某种权利	M	*	无			许可证基本组成单元

8.4.2.2. 主体

编号	名称	标识符	定义	约束	大小	次序	值域	数据类型	注释和举例
2	主体	subjectType	权利发布者和权利接受者						表1中所有类型均为subjectType
2.1	权利发布者	distributor	描述权利发布者的数字签名等信息	M	1	有			许可证价值传递链中的非终端用户许可证组成单元

编号	名称	标识符	定义	约束	大小	次序	值域	数据类型	注释和举例
2.2	权利接受者	receiver	描述权利接受者的数字签名等信息	M	1	有		String4096	许可证价值传递链中的中间环节或终端用户 许可证组成单元
2.3	主体组	group	可识别的个体的集合	O	1	有			权利接受者为多人时通过 group 进行标识

8.4.2.3. 权利

编号	名称	标识符	定义	约束	大小	次序	值域	数据类型	注释
3	权利	right	代表一种动作，使权利接受者在权威认证环境下，被授权对某种资源行使的操作	M	1	有			在语义上与某种资源相关联(至少应选择一项权限)
3.1	普通权利	normalRight	定义允许直接使用和处理资源的行为或动作	O	1	无			
3.1.1	使用类型	useActionType	使用资源时的动作						
3.1.1.1	输出	output	通过在本地设备解码，将码流输出到外部设备	O	1	无			
3.1.1.2	播放	play	通过解码，在本地设备上观看资源	O	1	无			针对具体不同的内容，与不同的应用相联系的“播放”会有相应的不同动作
3.1.2	重用类型	reuseActionType	表示权利接受者获得到资源后，可以再次使用的权利						

编号	名称	标识符	定义	约束	大小	次序	值域	数据类型	注释
3.1.2.1	修改	modify	对资源进行一定程度的编辑，形成新的资源	O	1	无			
3.1.2.2	分割	split	对资源进行分割，形成新的资源	O	1	无			
3.1.2.3	打包	package	对多个资源进行整合，形成新的资源	O	1	无			
3.1.3	管理类型	managementType	针对资源的管理操作						
3.1.3.1	移动	move	在本地的数字设备上移动资源	O	1	无			
3.1.3.2	复制	copy	资源在本地数字设备上的拷贝	O	1	无			
3.1.3.3	备份	backup	防止由于设备或人为的因素导致资源的丢失而采取的一种存储手段	O	1	无			
3.1.3.4	保存	save	在本地设备上存储该资源	O	1	无			
3.2	高级权利	advancedRight	定义在现有普通权利之上的相关行为或动作，在价值链中为转让或撤销下一级许可证	O	1	无			
3.2.1	共享类型	transferType	通过某种过程使权利接受者对资源的普通权利进行转移						
3.2.1.1	转移	moveRight	当前权利接	O	1	无			

编号	名称	标识符	定义	约束	大小	次序	值域	数据类型	注释
			受者将所具有的对资源的普通权利、约束和义务信息进行转移，转移后当前权利接受者不能执行原有的权利						
3.2.1.2	复制	copyRight	当前权利接受者将所具有的对资源的普通权利、约束和义务信息进行复制后转给其它用户，复制后当前权利接受者还可以执行原有的权利	O	1	无			
3.2.2	撤销类型	revokeType							
3.2.2.1	撤销	revoke	撤销权利接受者使用资源的权利	O	1	无			

表 8-1

8.4.2.4. 资源

编号	名称	标识符	定义	约束	大小	次序	值域	数据类型	注释
4	资源	resource	资源	M	1	有			
4.1	资源属性	resourceAttribute	对资源属性的描述	O	1	无			
4.2	密钥信息	keyinfo	与密钥相关的信息	O	1	无			
4.3	摘要信息	digest	与摘要相关的信息	O	1	无			

表 8-2

8.4.2.5. 约束

编号	名称	标识符	定义	约束	大小	次序	值域	数据类型	注释
5	约束	constraint	对资源权限的使用情景进行约束	O	1	无			在语义上与某种权利相关联
5.1	空间的约束	spatial	在指定空间范围内可以使用资源	O	*	无			
5.2	范围的约束	bounds	对资源使用范围的约束	O	*	无			
5.2.1	次数	count	内容使用的次数	O	*	无		Integer	一个正整数，例如：一段视频资源的播放次数被约束在 10 次，就说明该资源可以被播放 0 至 10 次
5.2.2	内容范围	range	指出内容可使用的范围：（min,max）	O	*	无		Integer	支持正的和负的十进制数，如果没有指定最小或最大值，那么这个范围就是开放的，例如，最小值为“1”（没有最大值）意味着这个范围的最大值是无穷大；注：最小值必须总是不大于最大值，而且，其中之一必须存在；例如：一段音频资源代表的是一盘音乐专辑，而权利发布者限制用户只能播放其中的第 2

编号	名称	标识符	定义	约束	大小	次序	值域	数据类型	注释
									首歌到第5首歌的内容
5.3	时间的约束	temporal	在指定时间内行使权利	O	*	无			
5.3.1	时刻	dateTime	可以是开始的时刻，结束的时刻，也可以是开始和结束时刻所决定的时间段，因此这个元素可以包括用来描述起始时间或是固定时间的实体	O	*	无		Datetime	天或者时间的值必须符合 [ISO8601]标准；如果没有开始值或者结束值，则该元素表示开放式的范围，注意：开始的值必须小于或者等于结束的值，如果没有出现固定时间值，则开始值必须出现；固定值只能单独出现
5.3.2	累加时间	accumulated	累加的时间	O	*	无		Timespan	累加时间用时间段来表示，时间段的值必须符合[ISO8601] 的标准；比如，“P30H”意味着一个 30 小时的时间段，也就是约束一段视频资源的累加播放时间是 30 小时
5.3.3	时间段	interval	时间间隔	O	*	无		Timespan	时间间隔用时间段来表示，时间段的值必须符合[ISO8601] 的标准；比如，

编号	名称	标识符	定义	约束	大小	次序	值域	数据类型	注释
									“P7D”意味着连续七天的使用时间，也就是一段视频资源可以被连续的播放 7 天
5.3.4	使用周期	period	周期性时间约束的属性	O	*	无			
5.4	使用环境的约束	System	对处理资源的计算机系统等硬件和软件方面的要求	O	*	无			内容执行装置将许可限制中的某项系统限制信息（即系统子元素信息）与其相应实际系统信息进行比较，判断是否符合要求，符合则可以执行所述内容对象，否则，放弃执行所述内容对象。
5.4.1	处理器	CPU	CPU 型号	O	*	无			
5.4.2	显示设备	Screen	显示设备类型	O	*	无			
5.4.3	存储设备	storeDevice	存储设备类型	O	*	无			
5.4.4	内存	memory	内存方式	O	*	无			
5.4.5	打印设备	printer	打印设备类型	O	*	无			
5.4.6	DRM 版本	DRM version	DRM 版本号	O	*	无			
5.4.7	解码器版本	DEC version	解码单元版本号	O	*	无			

表 8-3

8.4.2.6. 义务

编号	名称	标识符	定义	约束	大小	次序	值域	数据类型	注释
6	义务	duty		O	1	有			
6.1	付费方式	paymentMethod	定义用户使用资源时付费方式和费用等信息	O	*	无			
6.1.1	付费属性	parmentAttribute	描述了用户需遵从的付费方式,标识付费义务与权利的关系						
6.1.1.1	使用前	prePay	授权或者使用权限之前支付金额	O	1	无			
6.1.1.2	使用后	postPay	授权或者使用权限之后支付金额	O	1	无			
6.1.1.3	按次付费	perUse	每次使用授权权限时支付金额	O	1	无			
6.1.2	费用	fee	定义用户使用资源时所付费用的信息	O	1	有			
6.1.2.1	付费金额	amount	费用额	O	1	有		Decimal	必须是正的十进制数到两位十进制数;托管的货币必须使用[ISO4217]编码
6.1.2.2	税率	taxPercent	税率	O	1	有	0—100	Decimal	必须是0到100（包括）之间的

编号	名称	标识符	定义	约束	大小	次序	值域	数据类型	注释
							之间的数		数
6.2	交互性要求	interactive	权利发布者和接受者之间的一些交互活动						
6.2.1	接受版权声明	accept	用户使用资源前必须阅读并同意版权声明	O	*	无		String4096	
6.2.2	注册	register	用户在使用资源前必须先登记注册	O	*	无		String4096	
6.3	对用户的要求	requirementForUser	对用户的一些要求						
6.3.1	可追踪	tracked	用户对某资源行使权利后，记录用户对该资源行使权利的行为	O	*	无		String4096	用户必须清权利发布者的保密政策

表 8-4

8.4.2.7. 元素基本属性

编号	名称	标识符	定义	约束	大小	次序	值域	数据类型	注释
7	元素基本属性	context	定义许可证中各个元素的基本属性	M	1	无			许可证中各部分都可选择性的对其进行定义
7.1	名称	name	用来描述某实体的名字	O	1	无		String255	
7.2	版本	edition	指定实体的版本	O	1	无		String255	
7.3	日期	dateTime	指定实体发生或有效的日期	O	1	无		Datetime	
7.4	物理地址	physicalLocation	可定位的物理地址	O	1	无		String255	
7.5	数字地址	url	可定位的数字地址	O	1	无		URI	
7.6	引用	reference	一个指向附加信息的连接（URI）	O	1	无		URI	
7.7	注释	note	解释、描述等附加信息	O	1	无		String4096	
7.8	语言	language	采用的语言	O	1	无		String255	

表 8-5

8.4.2.8. 许可证标识

编号	名称	标识符	定义	约束	大小	次序	值域	数据类型	注释
8	许可证标识	characterGroup	用来对许可证中的元素进行定义和存储的标志	M	1	无			
8.1	licenseID	licenseID	许可证唯一标识，用于许可证定位和存储	M	1	无			
8.2	elementID	id	用来描述许可证中元素的唯一标识	O	1	无			
8.3	elementIDref	idref	引用某个许可证元素时的唯一标识	O	1	无			

表 8-6

参考文献

[1]

附录 A 数字权利描述语言 XML 绑定

A.1 名称空间前缀

avs-drel：表示数字权利描述语言中的权利描述实体。
avs-rdd：表示数字权利描述语言中的数据字典实体。

A.2 AVS-DREL schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:avs-drel="urn:avs:drm:2005:03-AVS-DREL-NS"
    xmlns:xsd=http://www.w3.org/ 2001/XMLSchema
    xmlns:ds=http://www.w3.org/2000/09/xmldsig#
    xmlns:enc=http://www.w3.org/2001/04/xmlenc#
    targetNamespace="urn:avs:drm:2005:03-AVS-DREL-NS"
    elementFormDefault="qualified" attributeFormDefault="qualified" version="1.1">
<xsd:import namespace=http://www.w3.org/2000/09/xmldsig#
    schemaLocation="http://localhost:8080/relSchema/xmldsig-core-schema.xsd"/>
<xsd:import namespace=http://www.w3.org/2001/04/xmlenc#
    schemaLocation="http://localhost:8080/relSchema/xenc-schema.xsd"/>
<!--define license element-->
    <xsd:element name="license" type="avs-drel:licenseType"/>
    <xsd:element name="licenseUnit" type="avs-drel:licenseUnitType"/>
    <xsd:complexType name="licenseType">
        <xsd:sequence>
            <xsd:element name="title" type="xsd:string" minOccurs="0"/>
            <xsd:element ref="avs-drel:context" minOccurs="0"/>
            <xsd:element name="distributer" type="avs-drel:subjectType" minOccurs="0"/>
            <xsd:element ref="avs-drel:licenseUnit" maxOccurs="unbounded"/>
            <xsd:element ref="ds:Signature" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="avs-drel:characterGroup"/>
    </xsd:complexType>
    <xsd:complexType name="licenseUnitType">
        <xsd:sequence>
            <xsd:element name="unitTitle" type="xsd:string" minOccurs="0"/>
            <xsd:element ref="avs-drel:context" minOccurs="0"/>
            <xsd:element name="receiver" type="avs-drel:subjectType"/>
            <xsd:element ref="avs-drel:right"/>
            <xsd:element ref="avs-drel:resource"/>
            <xsd:element ref="avs-drel:constraint" minOccurs="0"/>
            <xsd:element ref="avs-drel:duty" minOccurs="0"/>
        </xsd:sequence>
```

```
            <xsd:attributeGroup ref="avs-drel:characterGroup"/>
        </xsd:complexType>
    <!--define subjectType-->
        <xsd:element name="unit">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element ref="avs-drel:context" minOccurs="0"/>
                </xsd:sequence>
                <xsd:attribute name="loginName" type="xsd:string"/>
            </xsd:complexType>
        </xsd:element>
        <xsd:complexType name="subjectType">
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element ref="avs-drel:unit"/>
                <xsd:element name="group">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element ref="avs-drel:unit" minOccurs="0"
                                maxOccurs="unbounded"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
            </xsd:choice>
            <xsd:attributeGroup ref="avs-drel:characterGroup"/>
        </xsd:complexType>
    <!--define resource element-->
        <xsd:element name="resource">
            <xsd:complexType>
                <xsd:choice minOccurs="0" maxOccurs="unbounded">
                    <xsd:element ref="avs-drel:resourceUnit"/>
                    <xsd:element name="resourceGroup">
                        <xsd:complexType>
                            <xsd:sequence>
                                <xsd:element ref="avs-drel:resourceUnit"/>
                                <xsd:element name="unitInfo" minOccurs="0"
                                    maxOccurs="unbounded">
                                    <xsd:complexType>
                                        <xsd:attribute name="name" type="xsd:string"/>
                                    </xsd:complexType>
                                </xsd:element>
                            </xsd:sequence>
                        </xsd:complexType>
                    </xsd:element>
                </xsd:choice>
                <xsd:attributeGroup ref="avs-drel:characterGroup"/>
```

```
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="resourceUnit" type="avs-drel:resourceType"/>
        <xsd:element name="resourceElement" abstract="true"/>
        <xsd:complexType name="resourceType">
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element ref="avs-drel:context"/>
                <xsd:element ref="avs-drel:extends"/>
                <xsd:element name="digest">
                    <xsd:complexType>
                        <xsd:choice minOccurs="0" maxOccurs="unbounded">
                            <xsd:element ref="ds:DigestMethod"/>
                            <xsd:element ref="ds:DigestValue"/>
                        </xsd:choice>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element ref="avs-drel:keyinfo"/>
                <xsd:element ref="avs-drel:resourceElement"/>
            </xsd:choice>
            <xsd:attributeGroup ref="avs-drel:characterGroup"/>
        </xsd:complexType>
        <xsd:element name="extends" type="avs-drel:extendsType"/>
        <xsd:complexType name="extendsType">
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element ref="avs-drel:context"/>
            </xsd:choice>
            <xsd:attribute name="override" type="xsd:boolean" default="false"/>
            <xsd:attribute name="default" type="xsd:boolean" default="false"/>
        </xsd:complexType>
    <!--define right element-->
        <xsd:element name="right" type="avs-drel:rightType"/>
        <xsd:element name="rightElement" abstract="true"/>
        <xsd:complexType name="rightType">
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element ref="avs-drel:rightElement"/>
            </xsd:choice>
            <xsd:attributeGroup ref="avs-drel:characterGroup"/>
        </xsd:complexType>
    <!--define constraint element-->
        <xsd:element name="constraint" type="avs-drel:constraintType"/>
        <xsd:element name="constraintElement" abstract="true"/>
        <xsd:complexType name="constraintType">
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element ref="avs-drel:context"/>
                <xsd:element ref="avs-drel:container"/>
```

```
                <xsd:element ref="avs-drel:sequence"/>
            </xsd:choice>
            <xsd:attributeGroup ref="avs-drel:characterGroup"/>
        </xsd:complexType>
<!--define duty element-->
        <xsd:element name="duty" type="avs-drel:dutyType"/>
        <xsd:element name="dutyElement" abstract="true"/>
        <xsd:complexType name="dutyType">
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element ref="avs-drel:context"/>
                <xsd:element ref="avs-drel:container"/>
                <xsd:element ref="avs-drel:sequence"/>
            </xsd:choice>
            <xsd:attributeGroup ref="avs-drel:characterGroup"/>
        </xsd:complexType>
<!--define context element-->
        <xsd:element name="context" type="avs-drel:contextType"/>
        <xsd:element name="contextElement" abstract="true"/>
        <xsd:complexType name="contextType">
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element ref="avs-drel:contextElement"/>
            </xsd:choice>
        </xsd:complexType>
<!--define    IdGroup attributeGroup-->
        <xsd:attributeGroup name="characterGroup">
            <xsd:attribute name="id" type="xsd:ID"/>
            <xsd:attribute name="idref" type="xsd:IDREF"/>
        </xsd:attributeGroup>
<!--define container element-->
        <xsd:element name="container" type="avs-drel:containerType"/>
        <xsd:complexType name="containerType">
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element ref="avs-drel:constraintElement"/>
                <xsd:element ref="avs-drel:dutyElement"/>
                <xsd:element ref="avs-drel:constraint"/>
                <xsd:element ref="avs-drel:duty"/>
            </xsd:choice>
            <xsd:attribute name="type" default="and">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:NMTOKEN">
                        <xsd:enumeration value="and"/>
                        <xsd:enumeration value="in-or"/>
                        <xsd:enumeration value="ex-or"/>
                    </xsd:restriction>
                </xsd:simpleType>
```

```
            </xsd:attribute>
            <xsd:attributeGroup ref="avs-drel:characterGroup"/>
        </xsd:complexType>
    <!--define sequence element-->
        <xsd:element name="sequence" type="avs-drel:sequenceType"/>
        <xsd:complexType name="sequenceType">
            <xsd:sequence>
                <xsd:element ref="avs-drel:seqItem" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute name="order" default="total">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:NMTOKEN">
                        <xsd:enumeration value="total"/>
                        <xsd:enumeration value="partial"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
        </xsd:complexType>
        <xsd:element name="seqItem" type="avs-drel:seqItemType"/>
        <xsd:complexType name="seqItemType">
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element ref="avs-drel:constraintElement"/>
                <xsd:element ref="avs-drel:dutyElement"/>
                <xsd:element ref="avs-drel:container"/>
                <xsd:element ref="avs-drel:sequence"/>
            </xsd:choice>
            <xsd:attribute name="number" type="xsd:integer" use="required"/>
        </xsd:complexType>
        <xsd:element name="keyinfo">
            <xsd:complexType>
                <xsd:choice>
                    <xsd:element name="KeyArray">
                        <xsd:complexType>
                            <xsd:choice maxOccurs="unbounded">
                                <xsd:element name="EncryptedKey"
                                    type="avs-drel:EncryptedKeyType"/>
                                <xsd:element ref="ds:KeyInfo"/>
                            </xsd:choice>
                        </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="ControlKeyDeducedAlgorithm" type="xsd:string"/>
                </xsd:choice>
            </xsd:complexType>
        </xsd:element>
        <xsd:complexType name="EncryptedKeyType">
```

```
            <xsd:complexContent>
                <xsd:extension base="enc:EncryptedKeyType">
                    <xsd:attribute name="KeyID" type="xsd:integer"/>
                </xsd:extension>
            </xsd:complexContent>
        </xsd:complexType>
</xsd:schema>
```

A.3 AVS-RDD schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 2 U (http://www.xmlspy.com) by jinjing (elwg) -->
<xsd:schema xmlns:avs-drel="urn:avs:drm:2005:03-AVS-DREL-NS"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:avs-rdd="urn:avs:drm:2005:03-AVS-RDD-NS"
    targetNamespace="urn:avs:drm:2005:03-AVS-RDD-NS"
    elementFormDefault="qualified"
    attributeFormDefault="qualified" version="1.1">
<xsd:import namespace="http://www.w3.org/2000/09/xmldsig#"
    schemaLocation="http://localhost:8080/relSchema/xmldsig-core-schema.xsd"/>
<xsd:import namespace="urn:avs:drm:2005:03-AVS-DREL-NS"
    schemaLocation="http://localhost:8080/relSchema/avs-drel.xsd"/>
<!-- Declare all the right Elements -->
    <xsd:element name="normalRight" substitutionGroup="avs-drel:rightElement">
        <xsd:complexType>
            <xsd:choice>
                <xsd:element name="output" type="avs-drel:rightType"/>
                <xsd:element name="play" type="avs-drel:rightType"/>
                <xsd:element name="print" type="avs-drel:rightType"/>
                <xsd:element name="modify" type="avs-drel:rightType"/>
                <xsd:element name="split" type="avs-drel:rightType"/>
                <xsd:element name="package" type="avs-drel:rightType"/>
                <xsd:element name="move" type="avs-drel:rightType"/>
                <xsd:element name="copy" type="avs-drel:rightType"/>
                <xsd:element name="save" type="avs-drel:rightType"/>
                <xsd:element name="backup" type="avs-drel:rightType"/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="advancedRight" substitutionGroup="avs-drel:rightElement">
        <xsd:complexType>
            <xsd:choice>
                <xsd:element name="moveRight" type="avs-drel:rightType"/>
                <xsd:element name="copyRight" type="avs-drel:rightType"/>
                <xsd:element name="revoke" type="avs-drel:rightType"/>
```

```
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
<!-- Declare the fee Element (used in duty) -->
    <xsd:element name="fee">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="amount">
                    <xsd:complexType>
                        <xsd:simpleContent>
                            <xsd:extension base="xsd:decimal">
                                <xsd:attribute name="currency" type="xsd:NMTOKEN"
                                    use="required"/>
                            </xsd:extension>
                        </xsd:simpleContent>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element name="taxPercent" minOccurs="0">
                    <xsd:complexType>
                        <xsd:simpleContent>
                            <xsd:extension base="xsd:decimal">
                                <xsd:attribute name="code" type="xsd:NMTOKEN"
                                    use="required"/>
                            </xsd:extension>
                        </xsd:simpleContent>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
<!-- Define the dataTypes used for duty that use fee element -->
    <xsd:complexType name="paymentMethodType">
        <xsd:complexContent>
            <xsd:extension base="avs-drel:dutyType">
                <xsd:sequence>
                    <xsd:element ref="avs-rdd:fee"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
<!-- Declare all the duty Elements -->
    <xsd:element name="paymentMethod" substitutionGroup="avs-drel:dutyElement">
        <xsd:complexType>
            <xsd:choice>
                <xsd:element name="prepay" type="avs-rdd:paymentMethodType"
```

```
                    minOccurs="0"/>
                <xsd:element name="postpay" type="avs-rdd:paymentMethodType"
                    minOccurs="0"/>
                <xsd:element name="peruse" type="avs-rdd:paymentMethodType"
                    minOccurs="0"/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="accept" type="avs-drel:dutyType"
        substitutionGroup="avs-drel:dutyElement"/>
    <xsd:element name="register" type="avs-drel:dutyType"
        substitutionGroup="avs-drel:dutyElement"/>
    <xsd:element name="tracked" type="avs-drel:dutyType"
        substitutionGroup="avs-drel:dutyElement"/>
<!-- Declare all the Constraint Elements -->
    <xsd:simpleType name="dateAndOrTime">
        <xsd:union memberTypes="xsd:date xsd:dateTime"/>
    </xsd:simpleType>
    <xsd:complexType name="dateType">
        <xsd:complexContent>
            <xsd:extension base="avs-drel:constraintType">
                <xsd:choice>
                    <xsd:sequence>
                        <xsd:element name="start" type="avs-rdd:dateAndOrTime"
                            minOccurs="0"/>
                        <xsd:element name="end" type="avs-rdd:dateAndOrTime"
                            minOccurs="0"/>
                    </xsd:sequence>
                    <xsd:element name="fixed" type="avs-rdd:dateAndOrTime"
                        minOccurs="0"/>
                </xsd:choice>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="spatial" type="xsd:string"
        substitutionGroup="avs-drel:constraintElement"/>
    <xsd:element name="bounds" substitutionGroup="avs-drel:constraintElement">
        <xsd:complexType>
            <xsd:choice maxOccurs="unbounded">
                <xsd:element name="count" type="xsd:positiveInteger"/>
                <xsd:element name="range">
                    <xsd:complexType>
                        <xsd:complexContent>
                            <xsd:extension base="avs-drel:constraintType">
                                <xsd:sequence>
```

```
                                    <xsd:element name="min" type="xsd:decimal"
                                        minOccurs="0"/>
                                    <xsd:element name="max" type="xsd:decimal"
                                        minOccurs="0"/>
                                </xsd:sequence>
                            </xsd:extension>
                        </xsd:complexContent>
                    </xsd:complexType>
                </xsd:element>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="temporal" substitutionGroup="avs-drel:constraintElement">
        <xsd:complexType>
            <xsd:choice maxOccurs="unbounded">
                <xsd:element name="dateTime" type="avs-rdd:dateType"/>
                <xsd:element name="accumulated" type="xsd:duration"/>
                <xsd:element name="interval" type="xsd:duration"/>
                <xsd:element name="period">
                    <xsd:complexType>
                        <xsd:complexContent>
                            <xsd:extension base="avs-rdd:dateType">
                                <xsd:attribute name="periodType" use="required">
                                    <xsd:simpleType>
                                        <xsd:restriction base="xsd:NMTOKEN">
                                            <xsd:enumeration value="daily"/>
                                            <xsd:enumeration value="weekly"/>
                                            <xsd:enumeration value="monthly"/>
                                        </xsd:restriction>
                                    </xsd:simpleType>
                                </xsd:attribute>
                            </xsd:extension>
                        </xsd:complexContent>
                    </xsd:complexType>
                </xsd:element>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="system" substitutionGroup="avs-drel:constraintElement">
    <xsd:complexContent>
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name=" CPU " type=" xsd:string " maxOccurs=" unbounded "/>
            <xsd:element name=" screen " type=" xsd:string " maxOccurs=" unbounded "/>
            <xsd:element name=" storeDevice " type=" xsd:string " maxOccurs="unbounded "/>
            <xsd:element name=" memory " type=" xsd:string " maxOccurs=" unbounded "/>
```

```
                <xsd:element name=" printer " type=" xsd:string " maxOccurs=" unbounded "/>
                <xsd:element name="DRMversion " type=" xsd:string " maxOccurs=" unbounded "/>
                <xsd:element name=" DECversion " type=" xsd:string " maxOccurs=" unbounded "/>
            </xsd:choice>
        </xsd:complexContent>
</xsd:element>
<!-- Declare all the context Elements -->
    <xsd:element name="name" type="xsd:string" substitutionGroup="avs-drel:contextElement"/>
    <xsd:element name="edition" type="xsd:string" substitutionGroup="avs-drel:contextElement"/>
    <xsd:element name="datetime" type="xsd:string"
        substitutionGroup="avs-drel:contextElement"/>
    <xsd:element name="physicalLocation" type="xsd:string"
        substitutionGroup="avs-drel:contextElement"/>
    <xsd:element name="url" type="xsd:anyURI" substitutionGroup="avs-drel:contextElement"/>
    <xsd:element name="reference" type="xsd:anyURI"
        substitutionGroup="avs-drel:contextElement"/>
    <xsd:element name="note" type="xsd:string" substitutionGroup="avs-drel:contextElement"/>
    <xsd:element name="language" type="xsd:string"
        substitutionGroup="avs-drel:contextElement"/>
</xsd:schema>
```

A.4 avs-drel中定义的元素

A.4.1 < characterGroup >属性组

< characterGroup >属性组

说明：characterGroup 属性将用来标识的 id 和 idref 的属性捆绑在一起

类型：属性组（attributeGroup）

子属性：id，idref

< id >属性

说明：每一个许可证中的元素都用一个 id 标识，便于定位和存储

类型：是由默认名称空间（http://www.w3.org/2001/XMLSchema）中 ID 元素定义的

< idref >属性

说明：idref 用来引用出现的其它的元素的 id

类型：是由默认名称空间（http://www.w3.org/2001/XMLSchema）中 IDREF 元素定义的

A.4.2 < license >元素

< license >元素

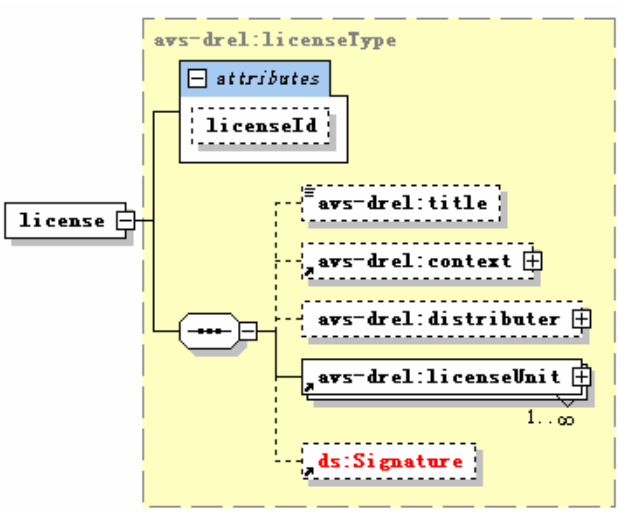
说明：license 元素代表的是 AVS DREL 中许可证类的最外层的元素

频度：没有限定在每个 XML 实例文档中的出现次数（默认为有且仅有 1 次）。

类型：license 元素的类型是由 licenseType 元素来定义

属性：见 licenseType

元素：见 licenseType



< licenseType >元素
说明：用 licenseType 元素来定义 AVS DREL 中许可证类的类型
类型：复杂类型（complexType）
属性：licenseId
元素：title，context，distributer，licenseUnit，Signature
< title >元素
说明：元素 title 进行内容定义，并显示于用户界面中以用户可读性语言对许可证进行简单描述
频度：在 licenseType 元素下，该元素可以出现 0 次或是 1 次
类型：由默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）中 string 定义
属性：没有定义
元素：没有定义
< context >元素
说明：该元素表示用来描述许可证的必要的一些属性
频度：在 licenseType 元素下，该元素出现 0 次或是 1 次
其它性质见最外层定义
< distributer >元素
说明：distributer 元素用来表示资源发布商
频度：在 licenseType 元素下，该元素出现 0 次或是 1 次
类型：distributer 元素的类型是由 subjectType 元素来定义
属性：定义见 subjectType
元素：定义见 subjectType
< licenseId >属性
说明：licenseId 属性用来对 license 进行唯一性标识
频度：在 licenseType 元素下，该元素有且仅有 1 次
< licenseUnit >元素
说明：licenseUnit 元素用来表示该许可证中的许可单元
频度：在 licenseType 元素下，该元素出现 1 次或多次
其它性质见最外层定义
< Signature >元素
说明：Signature 元素用来表示资源发布商提供的许可证签名
频度：在 licenseType 元素下，该元素出现 0 次或是 1 次
类型：在 avs-drel schema 中前缀名是 ds 的名称空间名是 http://www.w3.org/2000/09/xmldsig#，在

这个名称空间中的 Signature 元素，定义了该元素的类型

A.4.3 < licenseUnit >元素

< licenseUnit >元素

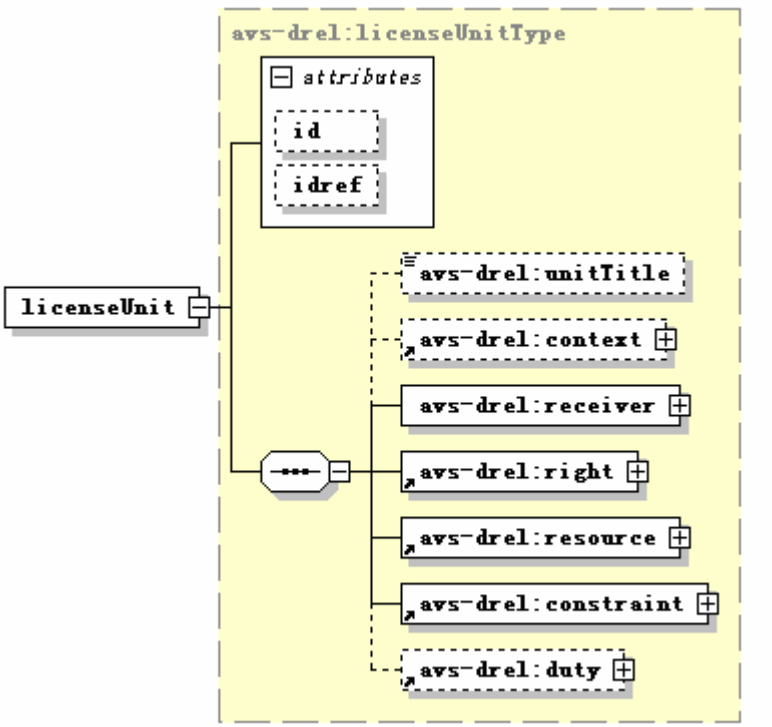
说明：licenseUnit 元素代表的是 AVS DREL 中许可单元类的最外层的元素

频度：在 license 元素下，该元素出现 1 次到多次。

类型：licenseUnit 元素的类型是由 licenseUnitType 元素来定义

属性：见 licenseUnitType

元素：见 licenseUnitType



< licenseUnitType >元素

说明：licenseUnitType 元素描述 AVS DREL 中许可单元类的类型

类型：复杂类型（complexType）

属性：characterGroup

元素：unitTitle，context，receiver，right，resource，constraint，duty

< unitTitle >元素

说明：元素 unitTitle 进行内容定义，并显示于用户界面中以用户可读性语言对许可单元进行简单描述

频度：在 licenseType 元素下，该元素出现 0 次或 1 次

类型：由默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）中 string 定义

属性：没有定义

元素：没有定义

< context >元素

说明：该元素表示用来描述许可单元的必要的一些属性

频度：在 licenseUnitType 元素下，该元素出现 0 次或是 1 次

其它性质见最外层定义

< receiver >元素

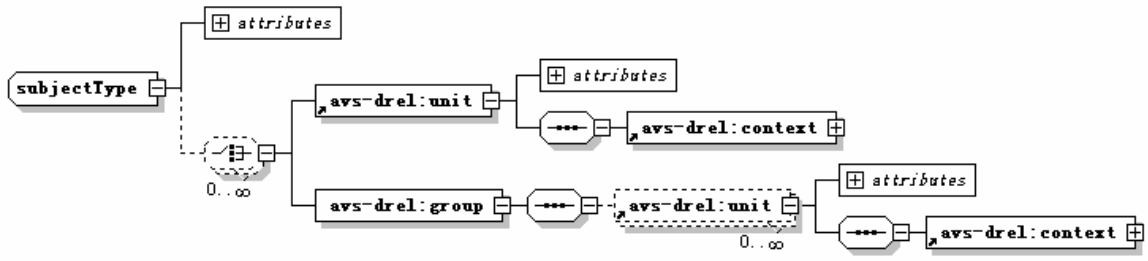
说明：receiver 元素定义了许可证中的主体元素，在这里是指权利接受者。

频度：默认出现次数（有且必须有 1 次）

类型：receiver 元素的类型是由 subjectType 元素来定义
属性：定义见 subjectType
元素：定义见 subjectType
< right >元素
说明：定义该许可单元中，权利接受者依据许可证可以行使的权利
频度：默认出现次数（有且仅有 1 次）
其它性质见最外层定义
< resource >元素
说明：定义该许可单元中，权利接受者依据许可证可以使用的资源
频度：默认出现次数（有且仅有 1 次）
其它的性质见最外层定义。
< constraint >元素
说明：定义该许可单元中，权利接受者依据许可证必须遵守的约束
频度：默认出现次数（0 次或 1 次。constraint 元素中可以包含有多个约束条件）
其它性质见最外层定义
< duty >元素
说明：duty 元素定义了权利接受者在行使一定权利的同时承担的义务。
频度：在 licenseUntiType 元素下，该元素可以出现 0 次或是 1 次（一个 duty 元素下可以包含有多个义务条件）
其它性质见最外层定义

A.4.4 < subjectType >元素

< subjectType >元素
说明：subjectType 元素描述 AVS DREL 中许可单元中主体类的最外层容器。
类型：复杂类型（complexType）
属性：characterGroup
元素：unit，group（这些元素被要求只能出现其中一种元素）



< unit >元素
说明：unit 元素定义了主体组中的每个个体
频度：在 group 元素下，该元素可以出现 0 次到无穷多次
类型：复杂类型（complexType）
属性：name（类型是字符串）
元素：没有定义
< group >元素
说明：group 元素定义了许可证中的出现的主体元素组，即多个用户共用一个主体。
频度：在 subjectType 元素下，该元素可以出现 0 次或是 1 次
类型：复杂类型（complexType）
属性：characterGroup
元素：unit

A.4.5 < right >元素

< right >元素

说明：right 元素定义了许可证授权权利接受者元素对资源进行的行为类型的最外层容器。

频度：默认出现次数（有且仅有 1 次）

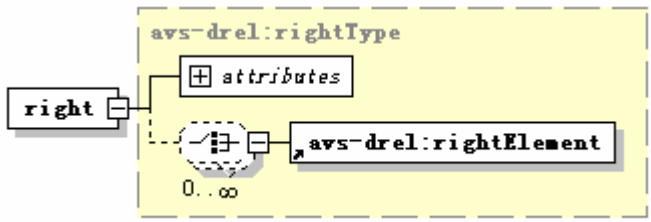
类型：rihgt 元素的类型是由 rightType 元素来定义

属性：定义见 rightType

元素：定义见 rightType

举例：

```
<avs-drel:right avs-drel:id="ID005">
        <avs-rdd:display/>
</avs-drel:right>
```



< rightType >元素

说明：rightType 元素定义许可证授权权利接受者元素对资源进行的行为类型。

类型：复杂类型（complexType）

属性：characterGroup

元素：rightElement

< rightElement>元素

频度：在 rightType 元素下，该元素可以出现 0 次到无穷多次。

其它性质见最外层定义。

< rightElement >元素

说明：rightElement 是个抽象的元素，这种元素是不能在 xml 的实例中出现的，但是在 avs-rdd 中一些元素被定义去实例化 rightElement，以此来表示具体用到的权利。

频度：默认出现次数（有且仅有 1 次）

类型：抽象的元素类型

属性：没有定义

元素：没有定义

A.4.6 < resource >元素

< resource >元素

说明：resource 元素定义了许可证授权权利接受者元素的资源类型的最外层容器。

频度：默认出现次数（有且必须有 1 次）

类型：复杂类型（complexType）

属性：characterGroup

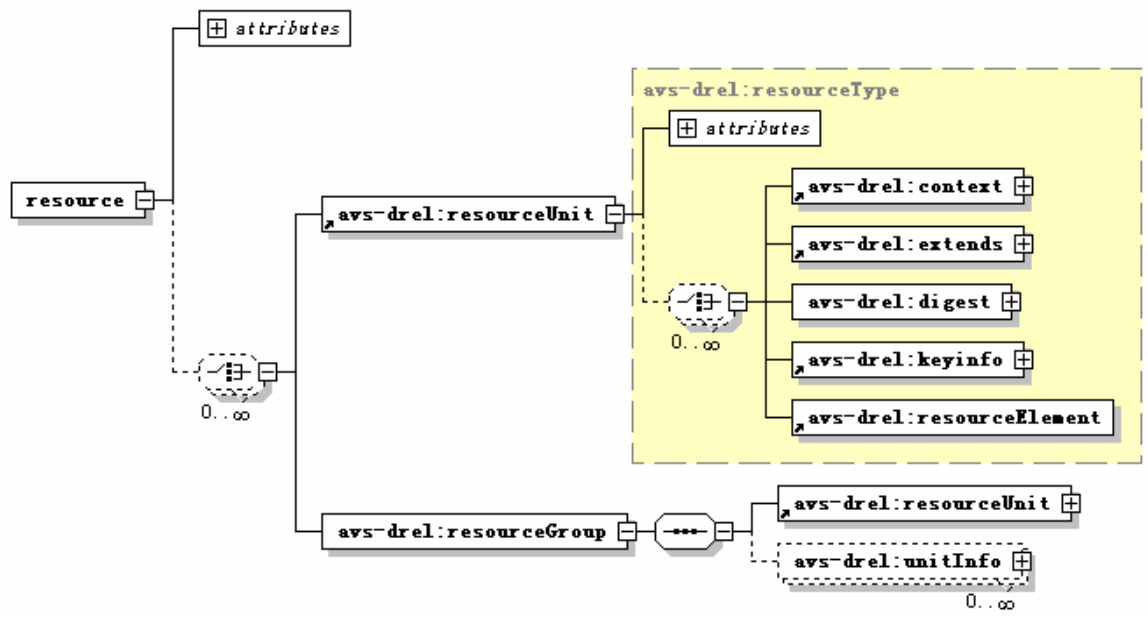
元素：resourceUnit，resourceGroup

< resourceUnit >元素：

说明：resourceUnit 表示持有的资源的个体

频度：在 resource 元素下，该元素可以出现 0 次到无穷多次

其它性质见最外层定义



< resourceGroup >元素

说明：resourceGroup 表示持有的资源组

频度：在 resource 元素下，该元素可以出现 0 次到无穷多次

类型：复杂类型（complexType）

属性：没有定义

元素：resourceUnit，unit

< resourceUnit >元素

说明：resourceUnit 表示持有的资源组作为一个整体所具有的信息

频度：默认出现次数（有且仅有 1 次）

其它性质见最外层定义

< unit >元素

说明：unit 表示持有的资源组中每一个资源个体的名字

频度：在 resourceGroup 元素下，该元素可以出现 0 次到无穷多次

类型：由默认名称空间（即名为 http://www.w3.org/2001/XMLSchema 的名称空间）中 string 定义

< resourceUnit >元素

说明：resourceUnit 元素定义了许可证授权权利接受者元素的资源类型的中的个体。

频度：默认出现次数（有且必须有 1 次）

类型：resourceUnit 元素的类型是由 resourceType 元素来定义

属性：定义见 resourceType

元素：定义见 resourceType

< resourceType >元素

说明：resourceType 元素定义了许可证授权权利接受者元素的资源类型的最外层容器。

类型：复杂类型（complexType）

属性：characterGroup

元素：context，extends，digest，KeyInfo，resourceElement（这些元素被限制只能出现其中的一个）

< context >元素

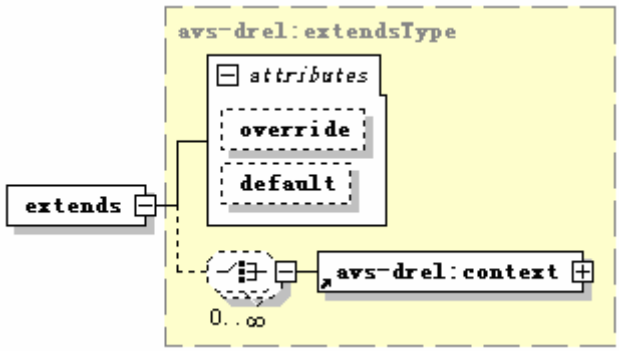
频度：在 resourceType 元素下，该元素可以出现 0 次到无穷多次

其它性质见最外层定义

< extends >元素
频度：在 resourceType 元素下，该元素可以出现 0 次到无穷多次
其它性质见最外层定义
< digest >元素
说明：digest 元素描述了资源作为摘要形式进行存储
频度：在 resourceType 元素下，该元素可以出现 0 次到无穷多次
类型：复杂类型（complexType）
属性：没有定义
元素：DigestMethod，DigestValue。（这些元素被限制只能出现其中的一个）
< DigestMethod >元素
说明：DigestMethod 从名为 http://www.w3.org/2000/09/xmldsig#（前缀名是：ds）名称空间中引用过来，表示摘要的方法
频度：在 digest 元素下，该元素可以出现 0 次到无穷多次。
类型：在 avs-drel schema 中前缀名是 ds 的名称空间名是 http://www.w3.org/2000/09/xmldsig#，在这个名称空间中的 DigestMethod 元素，定义了该元素的类型
< DigestValue >元素
说明：DigestValue 从 http://www.w3.org/2000/09/xmldsig#（前缀名为：ds）名称空间中引用过来，表示摘要的信息
频度：在 digest 元素下，该元素可以出现 0 次到无穷多次。
类型：在 avs-drel schema 中前缀名是 ds 的名称空间名是 http://www.w3.org/2000/09/xmldsig#，在这个名称空间中的 DigestValue 元素，定义了该元素的类型
< keyinfo >元素
说明：keyinfo 元素是自定义的用来存储密钥相关信息的容器。
频度：在 resourceType 元素下，该元素可以出现 0 次到无穷多次
类型：复杂类型（complexType）
其它性质见最外层定义。
< resourceElement >元素
说明：resourceElement 是个抽象的元素，这种元素是不能在 xml 的实例程序中出现的，但是在 avs-rdd 中一些元素被定义去实例化 resourceElement，以此来表示具体用到的资源类型。
频度：默认出现次数（有且仅有 1 次）
类型：抽象的元素类型
属性：没有定义
元素：没有定义

A.4.7 < extends >元素

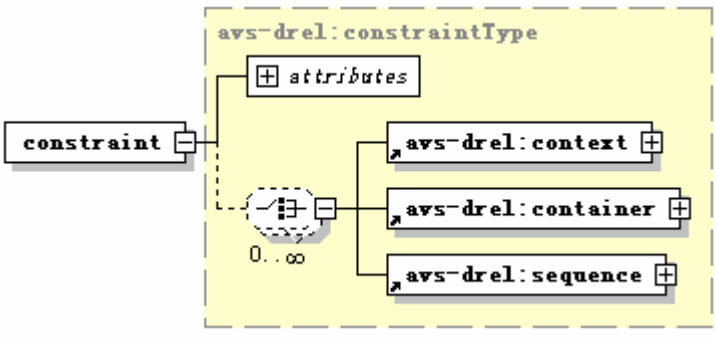
< extends >元素
说明：不同的许可单元（llicenseUnit）可能对同一个资源进行使用。当这个资源在第一个许可单元进行了定义，可以用 extends 元素引用该资源，可以避免重复的定义
频度：默认出现次数（有且必须有 1 次）
类型：extends 元素的类型是由 extendsType 元素决定的。
属性：见 extendsType
元素：见 extendsType



< extendsType >元素
说明：定义了 extends 的类型
频度：默认出现次数（有且仅有 1 次）
类型：复杂类型（complexType）
属性：override，default
元素：context
<context>元素
频度：在 extendsType 元素下，该元素可以出现 0 次到无穷多次
其它性质见最外层定义。
< override >属性
说明：该属性表示将先前定义的属性覆盖掉
类型：boolean 类型
< default >属性
说明：该属性表示将保留先前定义的属性
类型：boolean 类型

A.4.8 < constraint >元素

< constraint >元素
说明：constraint 元素定义了权利接受者对资源使用相应权利时的约束类型的最外层容器。
频度：默认出现次数（0 次或 1 次）
类型：constraint 元素的类型是由 constraintType 元素来定义
属性：定义见 constraintType
元素：定义见 constraintType



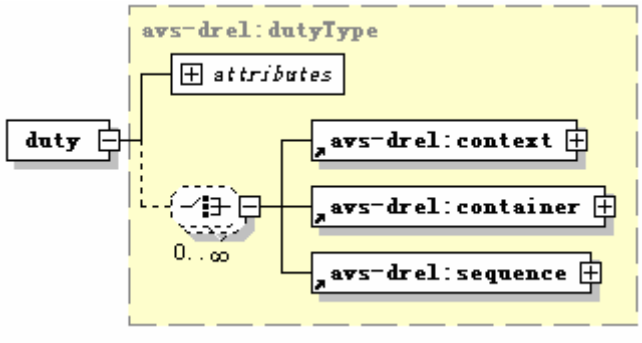
< constraintType >元素
说明：constraintType 元素定义了权利接受者对资源使用相应权利时的约束类型。
类型：复杂类型（complexType）
属性：characterGroup
元素：context, container，sequence

< context >元素
说明：该元素表示用来描述许可证的必要的一些属性
频度：在 constraintType 元素下，该元素可以出现 0 次或是 1 次
其它性质见最外层定义
< container >元素
频度：在 constraintType 元素下，该元素可以出现 0 次到无穷多次
其它性质见最外层定义
< sequence >元素
频度：在 constraintType 元素下，该元素可以出现 0 次到无穷多次
其它性质见最外层定义
< constraintElement >元素
说明：constraintElement 是个抽象的元素，这种元素是不能在 xml 的实例程序中出现的，但是在 avs-rdd 中一些元素被定义去实例化 constraintElement，以此来表示具体对许可证接受者的约束类型。
频度：默认出现次数（可以出现 0 次或多次）
类型：抽象的元素类型
属性：没有定义
元素：没有定义

A.4.9 < duty >元素

< duty >元素
说明：duty 元素定义了权利接受者在行使一定权利的同时承担的义务。
频度：默认出现次数（有且仅有 1 次）
类型：duty 元素的类型是由 dutyType 元素来定义
属性：定义见 dutyType
元素：定义见 dutyType
举例：

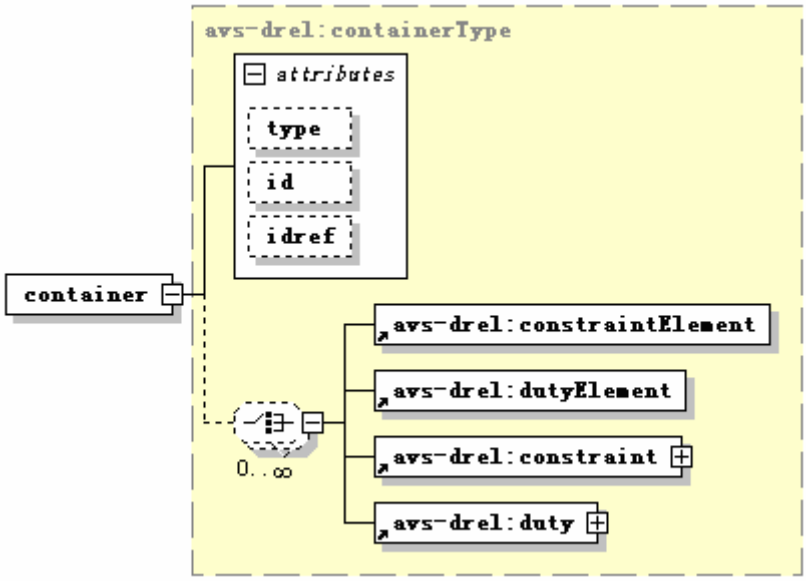
```
<avs-drel:duty avs-drel:id="ID009">
    <avs-drel:sequence avs-drel:order="total">
        <avs-drel:seqItem avs-drel:number="2">
            <avs-rdd:paymentMethod>
                <avs-rdd:prepay>
                    <avs-rdd:fee>
                        <avs-rdd:amount avs-rdd:currency="RMB">100</avs-rdd:amount>
                        <avs-rdd:taxPercent avs-rdd:code="rr">0.25</avs-rdd:taxpercent>
                    </avs-rdd:fee>
                </avs-rdd:prepay>
            </avs-rdd:paymentMethod>
        </avs-drel:seqItem>
        <avs-drel:seqItem avs-drel:number="1">
        </avs-drel:seqItem>
</avs-drel:sequence>
</avs-drel:duty>
```



< dutyType >元素
说明：dutyType 元素定义了权利接受者在行使一定权利的同时承担的义务。
类型：复杂类型（complexType）
属性：characterGroup
元素：container，sequence
< context >元素
说明：该元素表示用来描述许可证的必要的一些属性
频度：在 dutyType 元素下，该元素可以出现 0 次或是 1 次
其它性质见最外层定义
< container >元素
频度：在 dutyType 元素下，该元素可以出现 0 次到无穷多次
其它性质见最外层定义
< sequence >元素
频度：在 dutyType 元素下，该元素可以出现 0 次到无穷多次
其它性质见最外层定义
< dutyElement >元素
说明：dutyElement 是个抽象的元素，这种元素是不能在 xml 的实例程序中出现的，但是在 avs-rdd 中一些元素被定义去实例化 dutyElement，以此来表示具体的义务类型。
频度：默认出现次数（有且仅有 1 次）
类型：抽象的元素类型
属性：没有定义
元素：没有定义

A.4.10 < container >元素

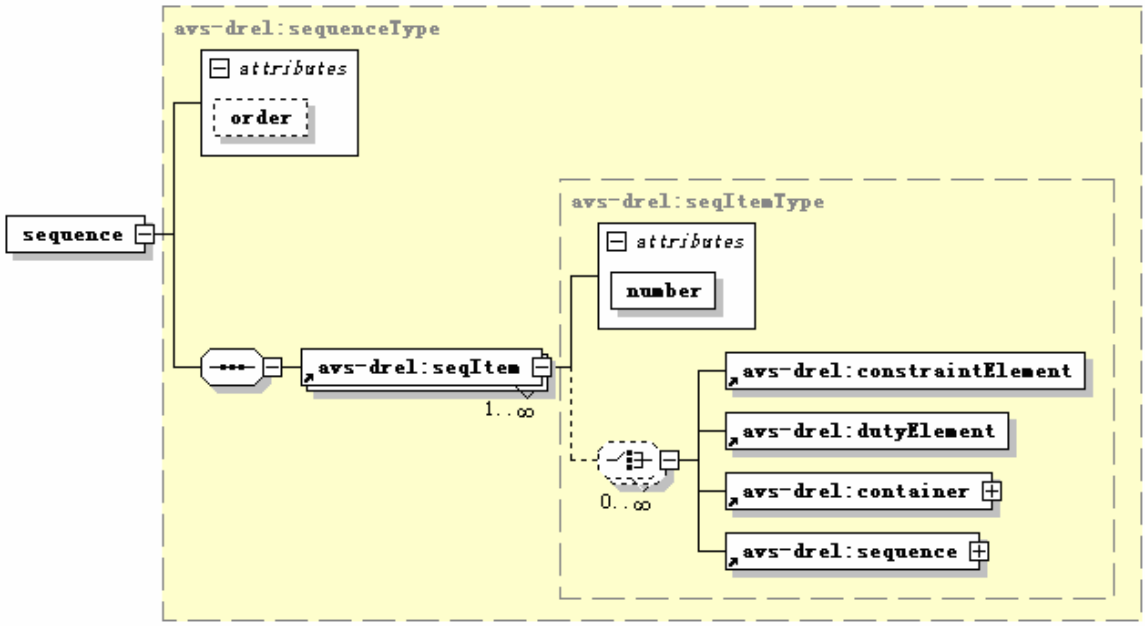
< container >元素
说明：用 container 定义在 duty 和 constraints 元素中嵌套的机制，另外可以定义嵌套进来的元素的关系（如 and、in-or、ex-or）
频度：默认出现次数（有且仅有 1 次）
类型：container 元素类型由 containerType 元素定义
属性：见 containerType 元素定义
元素：见 containerType 元素定义



< containerType >元素
说明：containerType 定义 container 的类型
频度：默认出现次数（有且仅有 1 次）
类型：复杂类型（complexType）
属性：type，characterGroup
元素：constraintElement，dutyElement，constraint，duty（这些元素被限制只能出现其中的一个）
< constraintElement >元素
频度：在 containerType 元素中，该元素可以出现 0 次到无数次
其它性质见最外层定义
< dutyElement >元素
频度：在 containerType 元素中，该元素可以出现 0 次到无数次
其它性质见最外层定义
< constraint >元素
频度：在 containerType 元素中，该元素可以出现 0 次到无数次
其它性质见最外层定义
< duty >元素
频度：在 containerType 元素中，该元素可以出现 0 次到无数次
其它性质见最外层定义
< type >属性
说明：由默认名称空间（即名为 http://www.w3.org/2001/XMLSchema 的名称空间）中 NMTOKEN 元素下继承下来的，将它的取值限定在“and”、“in-or”和“ex-or”。

A.4.11 < sequence >元素

< sequence >元素
说明：sequence 元素用于将不同的元素按照指定的顺序排列起来
频度：默认出现次数（有且仅有 1 次）
类型：sequence 元素类型由 sequenceType 元素定义
属性：见 sequenceType 元素定义
元素：见 sequenceType 元素定义



< sequenceType >元素
说明：该元素用来定义 sequence 的类型
频度：默认出现次数（有且仅有 1 次）
类型：复杂类型（complexType）
属性：order
元素：seqItem
< order >属性
说明：该属性定义了 sequence 下的元素是否应该遵守顺序，如果它的值是“total”，则表示元素必须遵守规定的顺序，如果值是“partial”，则遵守顺序是可选的
类型：类型从由默认名称空间（即名为 http://www.w3.org/2001/XMLSchema 的名称空间）中 NMTOKEN 元素下继承下来的
其它性质见最外层定义
< seqItem >元素
频度：在 sequenceType 元素中，该元素可以出现 1 次到无数次
其它性质见最外层定义
说明：该元素定义了元素在 sequence 下的顺序
类型：seqItem 元素类型由 seqItemType 元素定义
属性：见 seqItemType 元素定义
元素：见 seqItemType 元素定义
< seqItemType >元素
说明：定义了 seqItem 元素的类型
频度：默认出现次数（有且仅有 1 次）
类型：复杂类型（complexType）
属性：number
元素：constraintElement，dutyElement，container，sequence
< constraintElement >元素
频度：在 seqItemType 元素中，该元素可以出现 0 次到无数次
其它性质见最外层定义
< dutyElement >元素

频度：在 seqItemType 元素中，该元素可以出现 0 次到无数次
其它性质见最外层定义
< container >元素
频度：在 seqItemType 元素中，该元素可以出现 0 次到无数次
其它性质见最外层定义
< sequence >元素
频度：在 seqItemType 元素中，该元素可以出现 0 次到无数次
其它性质见最外层定义
< number >属性
说明：number 属性被定义为标记在各个元素里，用来指定它们的顺序，它的类型是默认名称空间（即名为 http://www.w3.org/2001/XMLSchema 的名称空间）中的 integer，并且这个属性在实例文档定义 seqItem 时必须出现。

A.4.12 < context >元素

< context >元素
说明：许可证中每个部分都有共同的特征，这些共同的特征组成了许可证的基本属性元素。context 元素是用来承载 AVS DREL 中各个部分基本属性的最外层容器。
频度：默认出现次数（有且仅有 1 次）
类型：context 元素的类型是由 contextType 元素来定义
属性：见 contextType
元素：见 contextType
举例：

```
<avs-drel:context>
    <avs-rdd:edition>v1.0</avs-rdd:edition>
    <avs-rdd:language>English</avs-rdd:language>
</avs-drel:context>
```

< contextType >元素
说明：contextType 元素是用来描述 AVS DREL 中各个部分元素基本属性
频度：默认出现次数（有且仅有 1 次）
类型：复杂类型（complexType）
属性：没有定义
元素：contextElement
< contextElement >元素
频度：在 contextType 元素下，该元素可以出现 0 次到无穷多次
其它性质见最外层定义
< contextElement >元素
说明：contextElement 是个抽象的元素，这种元素是不能在 xml 的实例程序中出现的，但是在 avs-rdd 中一些元素被定义去实例化 contextElement，以此来表示各个元素具体的属性值。
频度：默认出现次数（有且仅有 1 次）
类型：抽象的元素类型
属性：没有定义
元素：没有定义

A.4.13 < keyInfo >元素

< keyInfo >元素
说明：keyInfo 元素是自定义的用来存储密钥相关信息的容器。
频度：在 resourceType 元素下，该元素可以出现 0 次到无穷多次

类型：复杂类型（complexType）
元素：KeyArray，ControlKeyDeducedAlgorithm
< KeyArray >元素
说明：KeyArray 元素是存储主控密钥密文及其序号列表的容器。
频度：默认出现次数（有且仅有一次）
类型：复杂类型（complexType）
元素：EncryptedKey，KeyInfo
< ControlKeyDeducedAlgorithm >元素
说明：ControlKeyDeducedAlgorithm 元素存储的是主控密钥加密算法的信息。
频度：默认出现次数（有且仅有一次）
类型：复杂类型（complexType）

A.5 avs-rdd中定义的元素：

在 avs-rdd 以及 avs-drel 两个 Schema 中，有些元素，如 rightElement、resourceElement 等被定义成了抽象类型的元素，这些元素在 xml 实例中不能直接出现，因此必须单独声明元素，使得这些元素的类型为抽象元素。我们将这些声明的元素全部放在 avs-rdd schema 里面。

A.5.1 声明为rightElement的元素

< normalright >元素
说明：该元素定义的是定义允许直接使用和处理资源的行为或动作
类型：复杂类型（complextype）
元素：output，play，print，modify，split，package，copy，backup，save
< output >元素
说明：display 表示对资源进行输出的权利
类型：rightType
< play >元素
说明：play 表示对资源进行播放的权利
类型：rightType
< print >元素
说明：print 表示对资源进行打印的权利
频度：默认出现次数（有且仅有 1 次）
类型：rightType
< modify >元素
说明：modify 表示对资源进行修改的权利
类型：rightType
< split >元素
说明：split 表示对资源进行分割的权利
类型：rightType
< package >元素
说明:：package 表示对资源进行打包的权利
类型：rightType
< move >元素
说明：move 表示对资源进行移动的权利
类型：rightType
< copy >元素
说明：copy 表示对资源进行拷贝的权利

类型：rightType
< backup >元素
说明：backup 表示对资源进行备份的权利
类型：rightType
< save >元素
说明：save 表示对资源进行保存的权利
类型：rightType
< advancedright >元素
说明：该元素定义的是允许直接使用和处理权利的行为或动作
类型：复杂类型（complextype）
元素：moveRight，copyRight, revoke
< moveRight >元素
说明：moveRight 表示对权利进行转移的权利
类型：rightType
< copyRight >元素
说明：copyRight 表示对权利进行拷贝的权利
类型：rightType
< revoke >元素
说明：revoke 表示对权利进行撤销的权利
类型：rightType

A.5.2 声明为constraintElement的元素

< spatial >元素
说明：spatial 表示在空间上对使用该资源的约束
类型：字符串类型（complexType）
元素：没有定义
< bounds >元素
说明：bounds 表示在范围上对使用该资源的约束
类型：复杂类型（complexType）
元素：count，range
举例：

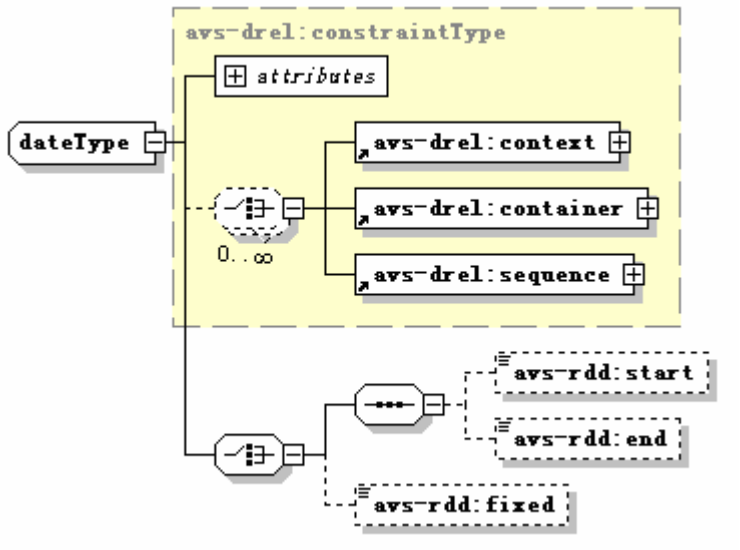
```
<avs-rdd:bounds>
    <avs-rdd:count>4</avs-rdd:count>
</avs-rdd:bounds>
```

< temporal >元素
说明：temporal 表示在时间上对使用该资源的约束
类型：复杂类型（complexType）
元素：dateTime，accumulated，interval，period
举例：

```
<avs-rdd:temporal>
     <avs-rdd:dateTime>
        <avs-rdd:start>2005-12-25</avs-rdd:start>
        <avs-rdd:end>2005-12-27</avs-rdd:end>
     </avs-rdd:dateTime>
</avs-rdd:temporal>
```

< dateAndOrTime >元素

说明：将默认名称空间（即名为 http://www.w3.org/2001/XMLSchema 的名称空间）date 元素和 dateTime 元素进行绑定
类型：联合类型（union）
属性：没有定义
元素：没有定义
< dateType >元素
说明：dateType 用来定义时间约束中的“时刻”，例如约束有些资源要在哪个时刻开始，哪个时刻结束，以及一段特定的时间
类型：复杂类型（complexType）
属性：只继承 constraintType 下的元素
元素：继承 constraintType 的类型，增加了 start，end 和 fixed 几个元素。（其中 start 和 end 元素绑定在一起，它们与 fixed 元素只能出现一个）



< start >元素
说明：表明一个开始时刻，说明权利必须在这个时刻之后才能行使
类型：dateAndOrTime
属性：见 dateAndOrTime
元素：见 dateAndOrTime
< end >元素
说明：表明一个结束时刻，说明权利必须在这个时刻之前才能行使
类型：dateAndOrTime
属性：见 dateAndOrTime
元素：见 dateAndOrTime
< fixed >元素
说明：表明一个结束时刻，说明权利必须在这个时刻才能行使
类型：dateAndOrTime
属性：见 dateAndOrTime
元素：见 dateAndOrTime
< count >元素
说明：count 表示在所表示的使用次数约束内使用该资源
类型：默认名称空间（即名为 http://www.w3.org/2001/XMLSchema 的名称空间）
positiveInteger 元素

< range >元素
说明：range 表示在所表示的资源范围约束内使用该资源
类型：复杂类型（complexType）
元素：从 constraintType 继承过来，添加了两个元素 min，max
< min >元素
说明：资源使用范围的下限值
频度：在 range 元素下，该元素可以出现 0 次到 1 次
类型：默认名称空间（即名为 http://www.w3.org/2001/XMLSchema 的名称空间）的 decimal 元素
< max >元素
说明：资源使用范围的上限值
频度：在 range 元素下，该元素可以出现 0 次到 1 次
类型：默认名称空间（即名为 http://www.w3.org/2001/XMLSchema 的名称空间）的 decimal 元素
< dateTime >元素
说明：dateTime 表示时间约束中的“时刻”或时间段
类型：dateType
< accumulated >元素
说明：accumulated 表示累加时间约束
类型：默认名称空间（即名为 http://www.w3.org/2001/XMLSchema 的名称空间）的 duration 元素
< interval >元素
说明：interval 表示从第一次使用开始计时的时间段约束
类型：默认名称空间（即名为 http://www.w3.org/2001/XMLSchema 的名称空间）的 duration 元素
< period >元素
说明：period 表示时间约束中会出现的周期时间的约束
类型：继承 constraintType 类型
属性：增加了一个属性，periodType
举例：

```
<avs-rdd:temporal>
    <avs-rdd:period avs-rdd:periodType="monthly">
      <avs-drel:container>
          <avs-rdd:temporal>
              <avs-rdd:dateTime>
                  <avs-rdd:start>9999-12-12</avs-rdd:start>
                  <avs-rdd:end>2005-12-22</avs-rdd:end>
              </avs-rdd:dateTime>
          </avs-rdd:temporal>
      </avs-drel:container>
</avs-rdd:period>
</avs-rdd:temporal>
```

< periodType >属性
说明：由默认名称空间（即名为 http://www.w3.org/2001/XMLSchema 的名称空间）中 NMTOKEN 元素下继承下来的，将它的取值限定在“daily”、“weekly”和“monthly”
< system >元素
说明：system 对处理资源的计算机系统等硬件和软件方面的要求;当同时出现多个< system >元素

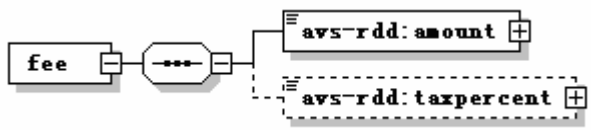
时，要求同时满足。若< system >元素下存在多个子元素，则要求满足其一。
类型：复杂类型（complexType）
元素：CPU，screen，storeDevice，memory，printer, DRMversion，DECversion
举例：

```
<avs-rdd: system >
        <avs-rdd: DRMversion> GB/T20090.6-200Y </avs-rdd: DRMversion>
        <avs-rdd: DRMversion> GB/T20090.6-YYYY </avs-rdd: DRMversion>
</avs-rdd: system >
<avs-rdd: system >
        <avs-rdd: DECversion> GB/T20090.2-2006 </avs-rdd: DECversion>
</avs-rdd: system >
<avs-rdd: system >
        <avs-rdd: DECversion> GB/T20090.3-200Y </avs-rdd: DECversion>
</avs-rdd: system >
```

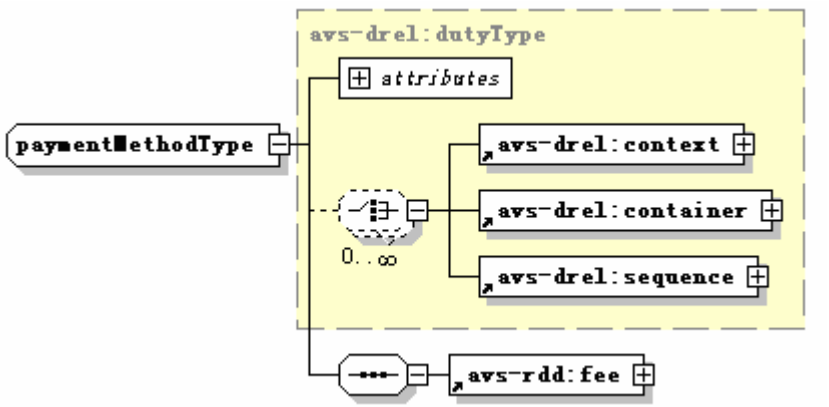
< CPU >元素
说明：CPU 表示对设备的 CPU 型号限定
类型：默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）的 string 元素
< screen >元素
说明：screen 表示对显示设备的型号限定
类型：默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）的 string 元素
< storeDevice >元素
说明：storeDevice 表示对存储设备的型号限定
类型：默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）的 string 元素
< memory >元素
说明：memory 表示对内存设备的型号限定
类型：默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）的 string 元素
< printer >元素
说明：printer 表示对打印设备型号的限定
类型：默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）的 string 元素
< DRMversion >元素
说明：DRMversion 表示对 DRM 系统版本的限定
类型：默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）的 string 元素
< DECversion >元素
说明：DECversion 表示对可信解码器版本的限定
类型：默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）的 string 元素

A.5.3 声明为dutyElement的元素

< fee >元素
说明：用 fee 来定义对资源所支付的费用的数据类型
类型：复杂类型（complexType）
属性：没有定义
元素：amount，taxpercent



< amout >元素
说明：amout 元素定义了为了获得资源所要付费的金额
频度：默认出现次数（有且仅有一次）
类型：从默认名称空间（即名为 http://www.w3.org/2001/XMLSchema 的名称空间）中的 decimal 元素类型继承下来
属性：添加了一个名为 currency 的属性。
元素：只保留了 deciaml 下的元素
< currency >属性
说明：currency 付费的金额的货币的类型
频度：这个属性被要求必须出现
类型：由默认名称空间（即名为 http://www.w3.org/2001/XMLSchema 的名称空间）中的 NMTOKEN 元素定义
< taxPercent >元素
说明：taxPercent 元素定义了为了获得资源所要付费的税率
频度：在 fee 元素下，该元素可以出现 0 次或是 1 次
类型：从默认名称空间（即名为 http://www.w3.org/2001/XMLSchema 的名称空间）中的 decimal 元素类型继承下来
属性：添加了一个名为 code 的属性。
元素：只保留了 deciaml 下的元素
< code >属性
说明：currency 付费的货币类型
频度：这个属性被要求必须出现
类型：由默认名称空间（即名为 http://www.w3.org/2001/XMLSchema 的名称空间）中的 NMTOKEN 元素定义
< paymentMethodType >元素
说明：用 paymentMethodType 来定义对资源所支付采取的方式
频度：默认出现次数（有且仅有 1 次）
类型：从 dutyType 类中继承而来
属性：只继承了 dutyType 的属性
元素：添加了一个 fee 类型的元素



< fee >元素
频度：默认出现次数（有且仅有 1 次）
其它性质见最外层定义
< paymentMethod >元素

说明：paymentMethod 表示对资源付费的相关信息
类型：复杂类型（complexType）
属性：没有定义
元素：prepay，postpay，peruse（这些元素被限制只能出现其中的一个）
< prepay >元素
说明：即要求授权或者使用权利之前支付金额
频度：在 paymentMethod 元素下，该元素可以出现 0 次或是 1 次
类型：paymentTypeMethod
< postpay >元素
说明：即允许权利使用后支付金额
频度：在 paymentMethod 元素下，该元素可以出现 0 次或是 1 次
类型：paymentTypeMethod
< peruse >元素
说明：即要求每次使用授权权利时支付金额
频度：在 paymentMethod 元素下，该元素可以出现 0 次或是 1 次
类型：paymentTypeMethod
< accept >元素
说明：accept 表示被授权资源权利前，用户必须浏览并且同意原文的信息
类型：dutyType
< register >元素
说明：register 表示授权并行使资源权利前，用户必须通过服务提供商注册自己的详细信息
类型：dutyType
< tracked >元素
说明：tracked 表示用户对某资源行使权利后，记录用户对该资源行使权利的行为
类型：dutyType

A.5.4 声明为context的元素

< name >元素：
说明：name 表示许可证中的每一元素的名字
类型：默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）的 string 元素
< edition >元素
说明：edition 表示许可证中的每一元素的版本号
类型：默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）的 string 元素
< dateTime >元素
说明：dateTime 表示许可证中的每一元素的创建的日期
类型：默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）的 dateTime 元素
< physicalLocation >元素
说明：physicalLocation 表示许可证中的每一元素的物理地址
类型：默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）的 string 元素
< url >元素
说明：url 表示许可证中的每一元素数字地址
类型：默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）的 anyURI 元素
< reference >元素
说明：reference 表示许可证中的每一元素对其他地址的引用
类型：默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）的 anyURI 元素
< note >元素

说明：note 表示许可证中的每一元素中注释
类型：默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）的 string 元素
< language >元素
说明：language 表示许可证中的每一元素采用的语言
类型：默认名称空间（http://www.w3.org/2001/XMLSchema 的名称空间）的 string 元素

附录 B AVS-DREL 实践指南（参考性资料）

B.1 规范一致性描述

B.1.1 严格一致性的实现

应该支持所有的必须性的数据元素和选择性的数据元素；

不应该使用扩展的数据元素；

不应该超出这一部分所描述的最小允许的最大值或者限制；

不应该解释或产生某些数据元素，这些元素依赖于任何不确定的、非定义的、定义执行的，或者局部确定的行为；

注：扩展特性或者扩展数据元素的使用是不确定的行为。

B.1.2 一致性的实现

应该支持所有的必须性数据元素；

可以支持一个或者多个选择性数据元素；

只要严格一致性应用的意义和行为没有被改变，可以在应用和数据交换的参与者的许可下使用扩展数据元素；

不应该支持或者使用改变了严格一致性应用意义或行为的扩展数据元素；

可以解释或产生依赖于定义执行的、局部确定的、或者不确定行为的数据元素；

注 1：扩展特性或者扩展数据元素的使用是不确定行为；

注 2：所有严格一致性的实现也是一致性实现；

注 3：如果一个实现通过扩展方法重新定义了特性，并且这些特性改变了严格一致性实现的意义和行为，那么这个实现不遵从这部分的内容。

B.2 应用样例

B.2.1 样例一

一段视频资源的权利接受者是 John。John 选择播放这段视频资源 4 次，并且只能在 2005 年的 2 月 9 日以前观看这段视频。

这个场景中的 xml 编码如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<avs-drel:license xmlns:avs-drel="urn:avs:drm:2005:03-AVS-DREL-NS" :
    dsig="http://www.w3.org/2000/09/xmldsig#"
    :xsi=http://www.w3.org/2001/XMLSchema-instance
    :enc=http://www.w3.org/2001/04/xmlenc#
    :ds=http://www.w3.org/2000/09/xmldsig#
    :avs-rdd="urn:avs:drm:2005:03-AVS-RDD-NS"
    drel:id="ID001"
    :schemaLocation=  "urn:avs:drm:2005:03-AVS-DREL-NS
                       http://localhost:8080/relSchema/avs-drel.xsd
                       urn:avs:drm:2005:03-AVS-RDD-NS
                       http://localhost:8080/relSchema/avs-rdd.xsd">
<avs-drel:context>
    <avs-rdd:name>The Voucher for XML: The Music</avs-rdd:name>
    <avs-rdd:url>http://example.com/avs/383838383.xml</avs-rdd:url>
    <avs-rdd:language>English</avs-rdd:language>
```

```
        <avs-rdd:datetime>2006-05-09</avs-rdd:datetime>
</avs-drel:context>
<avs-drel:distributer avs-drel:id="ID002">
        <avs-drel:unit avs-drel:loginName="hust">
                <avs-drel:context/>
        </avs-drel:unit>
</avs-drel:distributer>
<avs-drel:licenseUnit avs-drel:id="ID003">
        <avs-drel:receiver>
                <avs-drel:unit avs-drel:loginName="John">
                        <avs-drel:context/>
                </avs-drel:unit>
        </avs-drel:receiver>
        <avs-drel:right avs-drel:id="ID004">
                <avs-rdd:normalRight>
                        <avs-rdd:play/>
                </avs-rdd:normalRight>
        </avs-drel:right>
        <avs-drel:resource avs-drel:id="ID005">
                <avs-drel:resourceUnit>
                        <avs-drel:context>
                                <avs-rdd:name>XML: The Music</avs-rdd:name>
                        </avs-drel:context>
                        <avs-drel:keyinfo>
                                <avs-drel:KeyArray>
                                        <ds:KeyInfo>
                                                <ds:KeyName>key001</ds:KeyName>
                                                        <ds:KeyValue>
                                                                <ds:Y>UjBsR09EbGhjZ0dTQUxNQUFBUUN
                                                                        BRU1tQ1p0dU1GUXhEUzhi</ds:Y>
                                                        </ds:KeyValue>
                                        </ds:KeyInfo>
                                </avs-drel:KeyArray>
                        </avs-drel:keyinfo>
                        <avs-drel:digest>
                                <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                        </avs-drel:digest>
                </avs-drel:resourceUnit>
        </avs-drel:resource>
        <avs-drel:constraint avs-drel:id="ID006">
                <avs-drel:container>
                        <avs-rdd:bounds>
                                <avs-rdd:count>4</avs-rdd:count>
                        </avs-rdd:bounds>
                        <avs-rdd:temporal>
```

```
                    <avs-rdd:dateTime>
                        <avs-rdd:end>2005-2-9T23:59:59</avs-rdd:end>
                    </avs-rdd:dateTime>
                </avs-rdd:temporal>
                <avs-rdd:spatial>wuhan</avs-rdd:spatial>
            </avs-drel:container>
        </avs-drel:constraint>
</avs-drel:licenseUnit>
</avs-drel:license>
```

B.2.2 样例二

这个场景中，某公司获得授权，可以免费的将一段视频资源分发给一个叫 John 的用户，但对行使播放权利的时间范围进行了约束，只能在 2005 年的 2 月 9 日以前免费观看这段视频，并允许 John 传递许可证中的权利给另外的用户，传递的次数被限定为 2 次。

```
<?xml version="1.0" encoding="UTF-8"?>
<avs-drel:license xmlns:avs-drel ="urn:avs:drm:2005:03-AVS-DREL-NS"
    :dsig=http://www.w3.org/2000/09/xmldsig#
    :xsi="http://www.w3.org/2001/XMLSchema-instance"
    :enc=http://www.w3.org/2001/04/xmlenc#
    :ds=http://www.w3.org/2000/09/xmldsig#
    :avs-rdd="urn:avs:drm:2005:03-AVS-RDD-NS" drel:id="ID001"
    :schemaLocation="urn:avs:drm:2005:03-AVS-DREL-NS
                     http://localhost:8080/relSchema/avs-drel.xsd
                     urn:avs:drm:2005:03-AVS-RDD-NS
                     http://localhost:8080/relSchema/avs-rdd.xsd">
<avs-drel:context>
    <avs-rdd:name>The Voucher for XML: The Music</avs-rdd:name>
    <avs-rdd:url>http://example.com/avs/383838383.xml</avs-rdd:url>
    <avs-rdd:language>English</avs-rdd:language>
    <avs-rdd:datetime>2006-05-09</avs-rdd:datetime>
</avs-drel:context>
<avs-drel:distributer avs-drel:id="ID002">
    <avs-drel:unit avs-drel:loginName="hust">
        <avs-drel:context/>
    </avs-drel:unit>
</avs-drel:distributer>
<avs-drel:licenseUnit avs-drel:id="ID003">
    <avs-drel:receiver avs-drel:id="ID004">
        <avs-drel:unit avs-drel:loginName="John">
            <avs-drel:context/>
        </avs-drel:unit>
    </avs-drel:receiver>
    <avs-drel:right avs-drel:id="ID005">
        <avs-rdd:normalRight>
            <avs-rdd:play/>
        </avs-rdd:normalRight>
```

```
    </avs-drel:right>
    <avs-drel:resource avs-drel:id="ID006">
        <avs-drel:resourceUnit>
            <avs-drel:context>
                <avs-rdd:name>XML: The Music</avs-rdd:name>
            </avs-drel:context>
            <avs-drel:keyinfo>
                <avs-drel:KeyArray>
                    <ds:KeyInfo>
                        <ds:KeyName>key001</ds:KeyName>
                            <ds:KeyValue>
                                <ds:Y>UjBsR09EbGhjZ0dTQUxNQUFBUUN
                                        BRU1tQ1p0dU1GUXhEUzhi</ds:Y>
                            </ds:KeyValue>
                    </ds:KeyInfo>
                </avs-drel:KeyArray>
            </avs-drel:keyinfo>
            <avs-drel:digest>
                <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            </avs-drel:digest>
        </avs-drel:resourceUnit>
    </avs-drel:resource>

    <avs-drel:constraint avs-drel:id="ID007">
        <avs-drel:container>
            <avs-rdd:temporal>
                <avs-rdd:dateTime>
                    <avs-rdd:end>2005-2-9T23:59:59</avs-rdd:end>
                </avs-rdd:dateTime>
            </avs-rdd:temporal>
        </avs-drel:container>
    </avs-drel:constraint>
</avs-drel:licenseUnit>
<avs-drel:licenseUnit>
    <avs-drel:receiver>
        <avs-drel:unit avs-drel:loginName="John">
            <avs-drel:context/>
        </avs-drel:unit>
    </avs-drel:receiver>
    <avs-drel:right>
        <avs-rdd:advancedRight>
            <avs-rdd:moveRight avs-drel:idref="ID005"/>
        </avs-rdd:advancedRight>
    </avs-drel:right>
    <avs-drel:resource>
```

```
            <avs-drel:resourceUnit>
                <avs-drel:context>
                    <avs-rdd:name>XML: The Music</avs-rdd:name>
                </avs-drel:context>
                <avs-drel:digest>
                    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                </avs-drel:digest>
            </avs-drel:resourceUnit>
        </avs-drel:resource>
        <avs-drel:constraint>
            <avs-drel:container>
                <avs-rdd:bounds>
                    <avs-rdd:count>2</avs-rdd:count>
                </avs-rdd:bounds>
            </avs-drel:container>
        </avs-drel:constraint>
    </avs-drel:licenseUnit>
    </avs-drel:license>
```

附录 C　m 序列发生器（参考性资料）

本标准采用的 m 序列发生器的生成多项式为 x^25+x^3+1，初相位为长度为 25 的比特系列。此 m 序列发生器产生序列的步骤如下：

(3)　输出序列的第 25 位；

(4)　将序列的第 25 位和第 3 位取异或得到反馈比特；

(5)　把寄存器中第 1 至第 24 位右移一位，寄存器的第一位补为反馈比特的值。

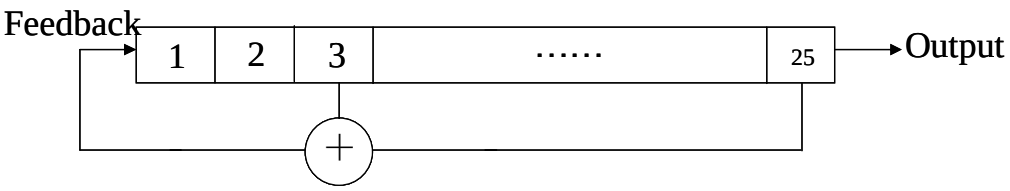


图 23　m 序列发生器

以下是初相位＝0101101001011010010110101 的一个用 matlab 实现 m 序列发生器的例子。

例：

```
//初始化初相位
register=[0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 1];
//定义输出序列
output=zeros(1,2^25-1);

for k=1:2^25-1
//计算输出序列第 K 位
    output(k)=register(25);
//按照生成多项式计算反馈比特
    feedback=mod(register(25)+register(3),2);
//把寄存器中第 1 至第 24 位右移一位，第一位补为反馈比特的值
    register=[feedback,register(1:24)];
end
```