国 际 电 信 联 盟

ITU-T J.1012

(04/2020)

国际电信联盟
电信标准化部门

系列J：有线网络和电视、声音节目及其他多媒体信号的传输

有条件接入和保护 – 可交换嵌入式有条件接入和数字版权管理解决方案

可交换CA/DRM解决方案的嵌入式通用接口（ECI）；CA/DRM容器、加载程序、接口、撤销

ITU-TJ.1012 建议书

国际电信联盟

ITU-T J.1012 建议书

可交换CA/DRM解决方案的嵌入式通用接口（ECI）；
CA/DRM容器、加载程序、接口、撤销

摘要

ITU-T J.1012建议书是多部分可交付成果的一部分，涵盖有关可交换CA/DRM解决方案的嵌入式通用接口规范的CA/DRM容器、加载程序、接口、撤销。

本ITU-T建议书是ETSI标准ETSI GS ECI 001-3的转换，是ITU-T SG9与ETSI ISG ECI之间合作的结果。对第2、7.7.2.5.2、9.4.4.6.2、9.4.6.1、9.5.2.2、9.8.1、9.8.2、10.2、I-2条和参考书目做了修改，并做了一些必要的编辑性更正。

沿革

版本	建议书名称	批准日期	研究组	唯一标识*
1.0	ITU-T J.1012	2020-04-23	9	11.1002/1000/13573

关键词

有条件访问（CA）、数字版权管理（DRM）、交换。

* 欲查阅此建议书，请在网络浏览器的地址字段内输入URL http://handle.itu.int/，然后再输入该建议书的唯一ID，例如：http://handle.itu.int/11.1002/1000/11830-en。

前言

国际电信联盟（ITU）是从事电信领域工作的联合国专门机构。ITU-T（国际电信联盟电信标准化部门）是国际电信联盟的常设机构，负责研究技术、操作和资费问题，并且为在世界范围内实现电信标准化，发表有关上述研究项目的建议书。

每四年一届的世界电信标准化全会（WTSA）确定ITU-T 各研究组的研究课题，再由各研究组制定有关这些课题的建议书。

WTSA第1号决议规定了批准建议书须遵循的程序。

属ITU-T研究范围的某些信息技术领域的必要标准，是与国际标准化组织（ISO）和国际电工技术委员会（IEC）合作制定的。

注

本建议书为简明扼要起见而使用的“主管部门”一词，既指电信主管部门，又指经认可的运营机构。

遵守本建议书的规定是以自愿为基础的，但建议书可能包含某些强制性条款（以确保例如互操作性或适用性等），只有满足所有强制性条款的规定，才能达到遵守建议书的目的。“应该”或“必须”等其他一些强制性用语及其否定形式被用于表达特定要求。使用此类用语不表示要求任何一方遵守本建议书。

知识产权

国际电联提请注意：本建议书的应用或实施可能涉及使用已申报的知识产权。国际电联对无论是其成员还是建议书制定程序之外的其他机构提出的有关已申报的知识产权的证据、有效性或适用性不表示意见。

至本建议书批准之日止，国际电联尚未收到实施本建议书可能需要的受专利保护的知识产权的通知。但需要提醒实施者注意的是，这可能并非最新信息，因此特大力提倡他们通过下列网址查询电信标准化局（TSB）的专利数据库：http://www.itu.int/ITU-T/ipr/。

© 国际电联 2020

版权所有。未经国际电联事先书面许可，不得以任何手段复制本出版物的任何部分。

目录

引言

本ITU-T建议书[1]是ETSI标准ETSI GS ECI 001-3的转换，是ITU-T SG9与ETSI ISG ECI之间合作的结果。对第2、7.7.2.5.2、9.4.4.6.2、9.4.6.1、9.5.2.2、9.8.1、9.8.2、10.2、I-2条和参考书目做了修改，并做了一些必要的编辑性更正。

本建议书的目的是促进电子通信服务、尤其是广播和视听设备市场的互操作性和竞争。然而，根据成员国的情况，还可以使用其他技术，这些技术也可能是适当的和有益的。

有条件访问（CA）和数字版权管理（DRM）实现的服务和内容保护对快速发展的数字广播和宽带服务领域而言至关重要。这包括将高清（HD）和超高清（UHD）内容分发到各种类型的客户端设备（**CPE**）[2]，以保护内容所有者和服务提供商（包括广播公司和付费电视运营商）的商业模式。尽管CA系统主要关注通过广播环境中通常使用的单向网络分发内容的保护，但DRM系统始于双向网络环境，并允许访问经认证用户的认证设备上的内容，典型地具有丰富的内容权利表达。在实践中，明确区分CA与DRM之间的功能在所有情况下都是不可行的，因此在本建议书中使用CA/DRM系统这一术语。

目前实施的CA/DRM解决方案无论是嵌入式的还是可拆卸的硬件，通常会导致服务/平台提供商和消费者的使用限制。对消费者而言，后果是依赖于适用的网络、服务和内容提供商以及适用于传统数字广播、使用网际协议的电视（IPTV）或过顶（OTT）业务的应用**CPE**。虽然具有嵌入式平台专有CA或DRM功能的**CPE**将客户绑定于某特定的平台运营商，但可拆卸硬件模块允许使用零售**CPE**，例如，机顶盒（STB）和集成电视机（iDTV）。由于其外形和成本的原因，可拆卸硬件模块无法满足未来的需求，特别是那些有关在平板电脑和移动设备上消费受保护内容以及成本敏感之部署的需求。

因此，现有技术约束了数字多媒质内容市场中众多参与者的自由。由于技术进步，创新的、基于软件的CA/DRM解决方案变得可行。这些解决方案能够最大限度地提高互操作性，同时保持高度的安全性，因此能够满足即将到来的市场需求，支持新业务，并通过广播和宽带连接拓宽消费者有关内容消费的选择范围。

在迁移或更换网络提供商后，消费者购买以供自身使用的**CPE**仍可供进一步使用，并且这些设备可用于不同商业视频门户的服务，这符合消费者的利益。这可以通过基于一个适当的安全架构，在**CPE**内部实现可互操作的CA和DRM机制来实现。只能防止**CPE**市场的进一步分散，通过确保CA和DRM系统（与先进的安全环境相关联）消费者友好的和灵活可交换的解决方案，来鼓励竞争。

平台运营商的兴趣在于，可以灵活部署安全技术，并可以轻松管理各种各样的网络和各种各样的设备。以一种无缝的方式、以最新的安全系统更新现有设备的优势在于，可提供无与伦比的商业机会。

---

[1] 附录II确定了一些有待进一步发展的领域。

[2] 在本建议书的文本中，使用粗体字来表示具有特定于嵌入式通用接口上下文的定义的术语，它们可能与通常用法有所不同。

根据**ECI**多部分可交付成果，本建议书中规定的**ECI生态系统**解决了有关重要属性的问题，例如，因基于软件的实施方案而带来的灵活性和可扩展性，推动实现面向未来之解决方案的可交换性，并实现了创新。其他属性为适用性，即适用于通过不同类型网络分发的内容，包括传统数字广播、IPTV和OTT服务。开放式生态系统的**ECI**系统规范促进了市场发展，为**CPE**中CA和DRM系统的可交换性奠定了基础，并为消费者提供了最低的成本，同时为CA或DRM供应商开发有关付费电视市场的目标产品提供了最低限度的限制。

除了本由多个部分构成的可交付成果的第4部分（解决虚拟机问题）以及第5部分（解决高级安全问题），构成第3部分的本建议书规定了所有必需的元素，它们对可信环境下下载和交换CA/DRM客户端（**ECI客户端**）及其执行环境（**ECI主机**）而言至关重要，包括通过详细规定的API与必要功能实体的通信。

ITU-T J.1012 建议书

可交换CA/DRM解决方案的嵌入式通用接口（ECI）；CA/DRM容器、加载程序、接口、撤销

1 范围

ECI系统的体系结构在[ITU-T J.1011]中进行定义；也请参阅[b-ETSI GS ECI 001-1]。**ECI**系统基于[ITU-T J.1010]中定义的要求；也请参阅[b-ETSI GS ECI 001-2]。本建议书规定了**ECI生态系统**的核心功能，包括CA/DRM容器、加载程序、接口和撤销细节；另见[b-Illgner]。与当前部署的系统相比，**ECI生态系统**的主要优势和创新是用于加载和交换CA/DRM系统的、完整的基于软件的体系结构，它避免了任何可拆卸的硬件模块。软件容器为CA或DRM内核（以下称为**ECI客户端**）及其各自的**虚拟机**实例提供了一个安全的（“沙箱”）环境。**ECI客户端**与**ECI主机**之间必需的和相关的应用程序编程接口（API）确保多个**ECI客户端**可以工作于安全的操作环境中，并与其余的**CPE**固件完全隔离，为此做了详细规定。**ECI主机**以及多个**ECI客户端**的安装和交换是**ECI**加载程序的任务，它最初由芯片加载器进行加载。在宽带接入的情况下，**ECI主机**和**ECI客户端**通过用于广播服务的数字视频广播（DVB）数据轮播与/或通过基于IP的机制从服务器进行下载。此过程嵌入在安全和可信的环境中，为**ECI主机**和**ECI客户端**的安装和交换提供一个信任层次结构，从而实现有效保护，防止完整性和替代性攻击。因此，**ECI生态系统**集成了高级安全机制，它依赖于控制字（CW）的高效和高级处理，规定为**密钥阶梯块**并集成在片上系统（SoC）硬件中，以便提供最高的安全性，以符合**ECI**要求。在存储受保护内容与/或将受保护内容输出给**ECI**兼容或不兼容的外部设备的情况下，**ECI**特定的高级安全功能在重加密过程中也扮演着重要角色。先进的微DRM系统提供所需的功能，并成为这一概念的有机组成部分。在撤销**CPE**或特定**ECI客户端**的情况下，高级安全功能也是相关的。本建议书规定了相关的API，而[ITU-T J.1014]和[ITU-T J.1015]详细介绍了高级安全性，也请参阅[b-ETSI GS ECI 001-5-1]和[b-ETSI GS ECI 001-5-2]。

众多API表征了**ECI生态系统**，以保证与相关实体的通信，例如，与**ECI**加载程序、受保护内容的入口和出口、高级安全性、解密和加密、本地存储设施和水印。其他API可用于**ECI客户端**人机接口（MMI）或者可选的**智能卡**读卡器。

ECI客户端的交换是由**用户**发起的，或者在必要的更新情况下可能会由**运营商**提出要求。就个人录像机（PVR）上的本地存储器可用或出于输出原因而言，至少支持两个**ECI客户端**，并有两个额外的**ECI客户端**。

本建议书涵盖以下条款中的规范细节：

ECI证书系统在第5节中进行规定，涵盖有关**ECI主机加载程序**、**ECI客户端加载程序**和**ECI运营商证书**等各种用途的**证书**，包括这些**证书**和相关**撤销列表**的定义、其链构成以及**根证书**的结构。

ECI主机加载程序在第6节中进行论述，其中**ECI主机**加载过程涉及图像的存储，**CPE**使用**ECI TA**提供的验证数据来验证图像的真实性，并随后激活图像。这包括文件格式规范、传输协议以及**运营商**特定之**ECI主机**图像的撤销。

第7节涵盖与**ECI客户端加载程序**有关的所有规范细节，它基于以下事实：**ECI主机**可以下载、存储和激活**ECI客户端图像**和相关的数据。**ECI客户端**加载过程可分为几个步骤，从发现过程到**ECI客户端**的下载和初始化，它允许使用来自广播流或来自互联网的数据来执行下载过程。

第8节涉及撤销规范详细信息，包括根据**CPE**硬件、**ECI主机**、加载之其他**平台操作**和**ECI客户端**的**ECI TA**状况，有选择地排除向**CPE**提供服务的功能。

ECI客户端接口的详细规范可以在第9节中找到，涵盖**ECI**生态系统所需的非常全面的规范细节、用于一般**ECI主机**资源的API、**ECI**特定的**ECI主机**资源、**ECI主机**解密资源、**ECI主机**重加密资源、内容保护相关的资源以及**ECI客户端**到**ECI客户端**相关的资源。

最后，第10节涉及强制的和可选的**ECI主机**功能。

本**ECI**核心规范仅适用于接收和进一步处理内容，它由有条件访问与/或数字版权管理系统进行控制，并已由服务提供商进行加密。

不受条件访问与/或DRM系统控制的内容不包含在本建议书中。

本建议书旨在与信任机构控制下的合同框架（许可协议）、合规性和稳健性规则以及适当的认证过程协议结合使用，它们不受**ECI**小组规范所代表之技术规范的约束。当中的一些基本问题可以在“信任环境”相关的、[b-ETSI GS ECI 001-6]的资料性附件中找到，它规定了与可信环境有关的技术机制和关系。

2 参考文献

下列ITU-T建议书及含有本建议书引用条款的其他参考文献构成本建议书的条款。所注明版本在出版时有效。所有建议书及其他参考文献均可能进行修订；因此鼓励建议书的使用方了解使用最新版本的下列建议书和其他参考文献的可能性。ITUT-T建议书的现行有效版本清单定期出版。本建议书在引用某一独立文件时，并未给予该文件建议书的地位。

[ITU-T J.1010] ITU-T J.1010建议书（2016年），用于可交换CA/DRM解决方案的嵌入式通用接口；用例和要求。

[ITU-T J.1011] ITU-T J.1011建议书（2016年），用于可交换CA/DRM解决方案的嵌入式通用接口；体系结构、定义和概述。

[ITU-T J.1013] ITU-T J.1013建议书（2020年），用于可交换CA/DRM解决方案的嵌入式通用接口；虚拟机。

[ITU-T J.1014] ITU-T J.1014建议书（2020年），用于可交换CA/DRM解决方案的嵌入式通用接口；高级安全性 – ECI特定的功能。

[ITU-T J.1015] ITU-T J.1015建议书（2020年），用于可交换CA/DRM解决方案的嵌入式通用接口；高级安全系统 – 密钥阶梯块。

[ITU-T T.871] ITU-T T.871建议书（2011年），信息技术 – 连续色调静止图像的数字压缩和编码：JPEG文件交换格式（JFIF）。

[ISO/IEC 23001-7] ISO/IEC 23001-7:2015，信息技术 – MPEG系统技术 – 第7部分：ISO基本媒体文件格式文件中的通用加密。

[ISO/IEC 23009-1] ISO/IEC 23009-1:2014，信息技术 – HTTP上的动态自适应流传输（DASH）– 第1部分：媒体表示描述和分段格式。

[ISO/IEC 13818-1-1] ISO/IEC 13818-1-1:2007，信息技术 – 运动图像和相关音频信息的通用编码 – 第1部分：系统。

[NIST Block 2001] National Institute of Standards and Technology, 2001,*Recommendation for Block Cipher Modes of Operation Methods and Techniques.* <https://www.nist.gov/publications/recommendation-block-cipher-modes-operation-methods-and-techniques>

[NIST FIPS 197] NIST U.S. FIPS PUB 197 (FIPS 197) (2001)，高级加密标准（AES）。

[ISO/IEC 21320] ISO/IEC 21320，信息技术 – 文件容器文件 – 第1部分：核心。

[IETF RFC 4122] IETF RFC 4122 (July 2015)，通用唯一标识符（UUID）URN命名空间。

[CEN EN 50221] CEN EN 50221 (1997)，有条件访问和其他数字视频广播解码器应用程序的通用接口规范。

[ETSI TS 102 006] ETSI TS 102 006，数字视频广播（DVB）；DVB系统中系统软件更新的规范。

[ETSI EN 301 192] ETSI EN 301 192，数字视频广播（DVB）；DVB数据广播规范。

[ETSI TR 101 202] ETSI TR 101 202，数字视频广播（DVB）；数据广播的实施指南。

[ISO/IEC 13818-6] ISO/IEC 13818-6，信息技术 – 运动图像和相关音频信息的通用编码 – 第6部分：DSM-CC的扩展。

[ETSI EN 300 468] ETSI EN 300 468，数字视频广播（DVB）；DVB系统中的服务信息（SI）规范。

[ETSI TS 101 162] ETSI TS 101 162，数字视频广播（DVB）；数字视频广播（DVB）系统的标识符和代码分配。

[ETSI TS 101 211] ETSI TS 101 211，数字视频广播（DVB）；服务信息（SI）的实施和使用指南。

[IETF RFC 768] IETF RFC 768，用户数据报协议（UDP）。

[IETF RFC 791] IETF RFC 791，网际协议（IP）。

[IETF RFC 793] IETF RFC 793，传输控制协议（TCP）。

[IETF RFC 1034] IETF RFC 1034，域名 – 概念和设施。

[IETF RFC 1035] IETF RFC 1035，域名 – 实现和规范。

[IETF RFC 8200] IETF RFC 8200，网际协议，版本6（IPv6）规范。

[IETF RFC 1123] IETF RFC 1123，互联网主机的要求 – 应用程序和支持。

[IETF RFC 952] IETF RFC 952，互联网主机表规范

[ISO/IEC 7816-1] ISO/IEC 7816-1，识别卡 – 集成电路卡 – 第1部分：带触点卡 – 物理特性。

[ISO/IEC 7816-2] ISO/IEC 7816-2，识别卡 – 集成电路卡 – 第2部分：带触点的卡 – 触点的尺寸和位置。

[ISO/IEC 7816-3] ISO/IEC 7816-3，识别卡 – 集成电路卡 – 第3部分：带触点的卡 – 电气接口和传输协议。

[ETSI TS 103 205] ETSI TS 103 205 (V1.2.1) (2015)，数字视频广播（DVB）；CI Plus™规范的扩展。

[ISO/IEC 7816-5] ISO/IEC 7816-5，识别卡 – 集成电路卡 – 第3部分：带触点的卡 – 注册应用提供商。

[ISO/IEC 7810] ISO/IEC 7810，识别卡 – 物理特性。

[ISO/IEC 23001-9] ISO/IEC 23001-9:2014，信息系统 – MPEG系统技术 – 第9部分：MPEG2传输流的通用加密。

[ETSI TS 103 285] ETSI TS 103 285 (2015)，数字视频广播（DVB）；用于在基于IP的网络上传输基于ISO BMFF的DVB业务的MPEG-DASH配置文件。

[ISO/IEC 14496-12] ISO/IEC 14496-12:2015，信息技术 – 视听对象编码 – 第12部分：ISO基本媒体格式。

[ETSI ETR 289] ETSI ETR 289 (1996)，数字视频广播（DVB）；支持在数字广播系统中使用加扰和有条件访问（CA）。

[ETSI TS 103 127] ETSI TS 103 127，数字视频广播（DVB）；使用MPEG2传输流的DVB IPTV业务的内容加扰算法。

[ETSI TS 100 289] ETSI TS 100 289，数字视频广播（DVB）；支持在数字广播系统中使用DVB加扰算法版本3。

[IETF RFC 7230] ETF RFC 7230 (2014)，超文本传输协议（HTTP/1.1）：消息语法和路由。

[IETF RFC 7231] IETF RFC 7231 (2014)，超文本传输协议（HTTP/1.1）：语义和内容。

[IETF RFC 5246] IETF RFC 5246 (2008)，传输层安全性（TLS）协议版本1.2。

[IETF RFC 5288] IETF RFC 5288 (2008)，TLS的AES伽罗瓦计数器模式（GCM）密码套件。

[IETF RFC 6066] IETF RFC 6066 (2011)，传输层安全性（TLS）扩展：扩展定义。

[IETF RFC 5280] IETF RFC 5280 (2008)，互联网X.509公钥基础设施证书和证书撤销列表（CRL）配置文件。

[IETF RFC 6818] IETF RFC 6818 (2013)，互联网X.509公钥基础设施证书和证书撤销列表（CRL）配置文件更新。

[IETF RFC 8446] IETF RFC 8446 (2018)，传输层安全性（TLS）协议版本1.3。

[W3C PNG] W3C Recommendation (2003)，便携式网络图形（PNG）规范（第2版）。

[IETF RFC 6151] IETF RFC 6151 (2011)，MD5消息摘要和HMAC-MD5算法的安全注意事项更新。

[IETF RFC 6125] IETF RFC 6125 (2011)，在传输层安全性（TLS）情形下使用X.509（PKIX）证书在互联网公钥基础设施中表示和验证基于域的应用程序服务标识。

[ISO/IEC 8859-1] ISO/IEC 8859-1:1998，信息技术 – 8比特单字节编码图形字符集，第1部分：拉丁字母1。

[ISO 3166-1] ISO 3166-1:2006，用于表示国家或地区名称的代码 – 第1部分：国家/地区代码。

[ISO 639-2] ISO 639-2:1998，用能与表示语言名称的代码 – 第2部分：Alpha-3代码。

[ISO/IEC 62766-5-2] ISO/IEC 62766-5-2:2017，用于访问IPTV和开放式多媒体服务的消费者终端功能 – 第5-2部分：万维网标准电视配置文件。

[W3C GIF V89a] W3C，图形交换格式版本89a。

[ISO/IEC 7816-4] ISO/IEC 7816-4，识别卡 – 集成电路卡 – 第4部分：交换的组织、安全和命令。

3 定义

3.1 其他地方定义的术语

无。

3.2 本建议书定义的术语

本建议书定义了下列术语：

在本建议书中使用粗体字和大写字母开头的术语表明，这些术语是用ECI特定的含义来定义的，这可能会偏离这些术语的通常用法。

3.2.1 高级安全系统（AS系统）（Advanced Security System (AS system)）：ECI兼容**CPE**的功能，为**ECI客户端**提供增强的安全功能（硬件和软件）。

3.2.2 AS时隙（AS slot）：由**ECI主机**专门提供给**ECI客户端**的高级安全块的资源。

3.2.3 AS时隙会话（AS slot session）：在与内容元素解密或重加密有关的AS时隙中的资源和计算。

3.2.4 兄弟（Brother）：同一**父**的其他**子**。

注 – **父**、**子**、**兄弟**指的都是管理**证书**的实体。

3.2.5 证书（Certificate）：本建议书第5节中定义的数据结构，带有标识某个**实体**的互补安全数字签名。

注 – 签名密钥的持有者证明数据的正确性 – 利用其密钥进行签名来验证数据的正确性。其公钥可用于验证数据。

3.2.6 证书链（Certificate Chain）：相互认证的**证书**列表，包括**根撤销列表**。

3.2.7 证书处理子系统（certificate processing subsystem (CPS)）：ECI主机的子系统，提供**证书**验证处理并提供额外的防篡改的稳健性。

3.2.8 子（child, children）：由（共同的）**父**签署的**证书**所指的**实体**。

注 – **父**、**子**、**兄弟**指的都是管理**证书**的实体：初始化用于启动**CPE** SoC的数据和软件。

3.2.9 内容保护系统（content protection system）：ECI生态系统中的系统，采用加密技术来管理对内容和服务的访问。

注 – 该术语可能会与替代服务保护系统频繁互换。这种典型的系统是有条件访问系统（CAS）或数字版权管理系统（DRM）。

3.2.10 客户端设备（customer premises equipment (CPE)）：已实施**ECI**的媒质接收器，允许**用户**访问数字媒质服务。

3.2.11 CPE制造商（CPE manufacturer）：生产符合**ECI**标准的**CPE**的公司。

3.2.12 ECI（嵌入式通用接口）（ECI (embedded CI)）：ETSI ISG“嵌入式CI”中规定的体系结构和系统，允许在客户端设备（**CPE**）中开发和实现基于软件的可交换**ECI客户端**，从而提供用于**ECI**的**CPE**设备互操作性。

3.2.13 ECI应用程序（ECI application）：ECI客户端上托管的、基于HTML的应用程序，并在专用浏览器会话中运行，以便与**用户**交互并向**ECI客户端**提供**用户**输入。

3.2.14 ECI芯片制造商（ECI chip manufacturer）：提供实现**ECI**指定芯片组功能的片上系统的公司。

3.2.15 嵌入式通用接口客户端（ECI client (embedded CI client)）：实现符合嵌入式通用接规范的CA/DRM客户端。

注 – 它是**CPE**中的软件模块，提供所有手段以受保护的方式来接收并控制消费者与内容有关的权益，内容通过内容分销商或**运营商**来分发。它还接收消费者可享有权益的条件，以及解密各种各样消息和内容的密钥。

3.2.16 ECI客户端图像（ECI client image）：包含软件作为VM代码的文件，以及**ECI客户端加载程序**所需的初始化数据。

3.2.17 ECI客户端加载程序（ECI client loader）：ECI主机的软件模块部分，允许在**ECI主机**的**ECI容器**中下载、验证和安装新的**ECI客户端**软件。

3.2.18 ECI容器（ECI container）：具有补充支持库和**ECI** API的单个VM实例，允许一个**ECI客户端**在**CPE**上运行单个实例。

3.2.19 ECI生态系统（ECI ecosystem）：由一个**TA**和若干个平台以及实地**ECI**兼容的**CPE**组成的商业运营环境。

3.2.20 ECI主机（ECI host）：CPE的硬件和软件系统，涵盖**ECI**相关功能，并有至**ECI客户端**的接口。

注 – **ECI主机**是CPE固件的一部分。

3.2.21 ECI主机图像（ECI host image）：包含**ECI**环境的软件和初始化数据的文件。

注1 – 一个**ECI主机**图像可包含多个**ECI主机**图像文件。

注2 – 它也可包含其他软件，不会对**ECI主机**造成干扰或不允许观察**ECI主机**。

3.2.22 ECI主机加载程序（ECI host loader）：软件模块，允许将**ECI主机**软件下载、验证和安装到**CPE**中。

注 – 在多阶段加载配置中，本术语用于指加载**ECI主机**时涉及的所有安全关键加载功能。

3.2.23 ECI根证书（ECI root certificate）：颁发用于验证**ECI TA**批准之项目的**证书**。

3.2.24 实体（entity）：由**ECI生态系统**中一个唯一标识确定的组织（例如制造商、**运营商**或**安全供应商**）或者现实世界科目（例如**ECI主机**、**平台操作**或**ECI客户端**）。

3.2.25 出口链（export chain）：用于将出口授权给一个或一组**微DRM系统**的**证书**链。

3.2.26 出口连接（export connection）：可以解密内容的**ECI客户端**与可以重加密内容的**微服务器**之间已认证的关系。

3.2.27 出口组（export group）：允许出口的**微DRM系统**组。

3.2.28 父（**father**）：**子实体证书**的签署者。

注 – **父**、**子**、**兄**弟指的都是管理证书的实体。

3.2.29 图像系列（image series）：ECI主机或**ECI客户端**的图像系列，因**CPE**的**CPE**_id不同而不同，尽管呈现（几乎）相同的功能。

3.2.30 入口链（import chain）：从**ECI客户端**的POPK到代表出口系统或**出口组**的**实体**的链。

注 – 出口链和匹配的入口链可用于验证将内容导入到出口E**CI客户端**的微服务器会话。

3.2.31 入口连接（import connection）：从**ECI客户端**到**微服务器**的批准连接，允许它为后续的重加密导入解密的内容。

3.2.32 制造商（manufacturer）：开发和销售**CPE**的实体，这些实体适应**ECI**系统的实施，并允许按照软件下载来安装**ECI主机**和**ECI客户端**。

3.2.33 媒质句柄（media handle）：参照**ECI客户端**与**ECI主机**之间的单个程序解密或重加密处理设置。

3.2.34 微客户端（micro client）：ECI客户端或非**ECI客户端**，可以解密由**微服务器**重加密的内容。

3.2.35 微服务器（micro server）：ECI客户端，可以导入解密的内容、对该内容进行重加密，并对特定的**ECI客户端**或**ECI客户端**组进行验证，作为后续解密的**目标**。

3.2.36 微DRM系统（micro DRM system）：内容保护系统，它使用**微服务器**对**CPE**上的内容进行重加密，并允许通过经认证的**微客户端**来对重加密的内容进行解码。

注 – 由微DRM系统运营商配置的微服务器和微客户端。

3.2.37 运营商（operator）：运营平台的组织，**ECI TA**支持之，以便签署**ECI生态系统**。

注 – 一个**运营商**可以**运营多个平台**。

3.2.38 平台操作（platform operation (PO)）：技术服务交付操作的具体实例，在安全性方面具有单一**ECI**标识。

3.2.39 重加密会话（re-encryption session）：由**微服务器**控制的、从**入口连接**导入内容的过程，重加密之，并生成经认证之**目标**所需的解密信息，以便后续解密之。

3.2.40 请求（request）：从发送方到接收方的消息，要求提供某些信息或者在一个**ECI生态系统**内执行某个操作，该操作在该请求的数据字段中指定。

注 – 更多细节在第9.2.3节中给出。

3.2.41 响应（response）：应答一个**请求**的、**ECI生态系统**内的消息。

注 – 更多细节在第9.2.3节中给出。

3.2.42 撤销列表（revocation list (RL)）：已撤销的**证书**列表，因此不应再使用。

3.2.43 根（root）：公钥或包含一个公钥的**证书**，作为认证一个**证书**链的基础。

3.2.44 安全认证信道（secure authenticated channel (SAC)）：已经在两个**实体**之间建立的通信路径（信道），当中各**实体**已经彼此安全地识别它们自己（已经认证）并同意对在它们之间传输的数据进行加密（安全）。

3.2.45 服务（service）：通过**平台操作**提供的内容。

注 – 在**ECI**情形下，只考虑受保护的内容。

3.2.46 发送方公钥（sender public key (SPK)）：加密内容发送方的公钥，在**ECI生态系统**中验证用于解密内容的、密钥链第一个密钥的签名来源，发送方是**平台操作**的一部分。

3.2.47 智能卡（smart card）：由若干个CA或DRM提供商使用的、可拆卸的硬件安全设备，以增强其**ECI生态系统**中产品的安全级别。

3.2.48 目标（target）：**微客户端**或一组**微客户端**，其内容由**微服务器**进行重加密。

3.2.49 信任机构（trust authority (TA)）：管理所有规则和法规的组织，它们适用于**ECI**的某个实施方案，针对的是某个市场。

注 – **信任机构**必须是一个能够实现法律主张的合法实体。**信任机构**需要对其管理的**ECI生态系统**中的所有参与者保持公正。

3.2.50 可信的第三方（Trusted Third Party (TTP)）：安全服务提供商，它向**ECI系统**相关组件的合规**制造商**颁发**证书**和**密钥**。

注 – 它受**信任机构（TA）**的控制。

3.2.51 用户（user）：操作符合**ECI**标准的设备的人员。

3.2.52 VM实例（VM instance）：ECI主机建立之虚拟机的实例化，在**ECI客户端**上显示为运行的执行环境。

4 缩写词和首字母缩略语

本建议书采用下列缩写词和首字母缩略语：

4CC 四字符编码（也称4CC）

3DES 3DES

AEAD 通过关联数据进行认证加密

AES 高级加密标准

AES-GCM AES伽罗瓦计数器模式

AID 应用程序标识符

AK 认证密钥

APDU	应用协议数据单元
API	应用程序编程接口
AS	高级安全
ASCII	美国信息交换标准代码
ATR	答复重置
BAT	业务群关联表
BMFF	基础媒质文件格式
BSD	伯克利软件分发
CA	有条件访问
CA/DRM	有条件访问/数字版权管理
CAT	有条件访问表
CBC	密码块链接
CENC	通用加密
CI	通用接口
CP	内容属性
CPE	客户端设备
CPS	证书处理子系统
CPU	中央处理器
CRC	循环冗余校验
CRL	证书撤销列表
CSA	通用加扰算法
CSA1	常用加扰算法，第1版
CSA3	通用加扰算法，第3版
CSS	W3C级联样式表
CSS3	CSS版本3
CTR	计数器模式
CW	控制字
DASH	HTTP上的动态自适应流
DDB	下载数据块
DDOS	分布式拒绝服务
DES	数据加密标准
DHE	瞬时Diffie-Hellman
DII	下载信息指示
DLNA	数字生活网络联盟
DNS	域名系统
DRM	数字版权管理
DSI	下载服务器启动

DSMCC	数字存储媒质命令和控制
DVB	数字视频广播
EAC	出口授权证书
EAOC	出口授权运营商证书
ECM	权利控制消息
EGC	出口组证书
EIT	事件信息表
EMM	权利管理消息
ES	基本流
ESC	出口系统证书
GCM	伽罗瓦/计数器模式
GMT	格林威治标准时间
HD	高清
HDCP	高带宽数字内容保护
HTML	超文本标记语言
HTTP	超文本传输协议
HTTP(S)	安全超文本传输协议
iDTV	集成数字电视接收器
IFSC	卡的信息字段大小
IFSD	设备的信息字段大小
IP	网际协议
IPTV	使用网际协议（IP）的电视
IPv4	网际协议版本4
IPv6	网际协议版本6
ISO	国际标准化组织
ISOBMFF	ISO基本媒质文件格式
LAN	局域网
LSB	最低有效位
MIME	多用途互联网邮件扩展
MMI	人机接口
MP4	数字多媒体容器格式（也称MPEG-4第14部分）
MPD	媒质演示说明
MPEG	运动图像专家组
MSB	最高有效位
n.a.	不适用
NV memory	非易失性存储器
NV	非易失性

OS	操作系统
OTT	过顶（在开放的互联网之上）
OUI	组织上的唯一标识符
PAT	节目关联表
PayTV	付费电视
PES	包基本流
PID	MPEG包标识符
PIN	个人识别码
PKIX	公钥基础设施X.509
PMT	节目映射表
PO	平台操作
POC	平台操作证书
POPK	平台操作公钥
PPS	协议和参数选择
PSI	节目特定信息
PSSH	保护系统特定标题
PVR	个人录像机
RAM	随机存取存储器
RFU	保留以供未来使用
RL	撤销列表
SAC	安全认证通道
SDT	业务描述表
SHA	安全散列算法
SI	业务信息
SIM	用户身份模块
SoC	片上系统
SPK	签名公钥（也称签名验证密钥）
SSK	签名密钥（也称签名私钥）
SSL	安全套接字层
SSU	系统软件更新
STB	机顶盒
TA	信任机构
TCK	校验字节
TCP	传输控制协议
TLS	传输层安全
TPDU	传输协议数据单元

TPEGC	第三方出口组证书
TS	传输流
TTP	可信的第三方
TV	电视
UDP	用户数据报协议
UHD	超高清
UI	用户接口
uimsbf	无符号整数，最高位在前
UNT	更新通知表
URI	使用权信息
URL	统一资源定位符
USB	通用串行总线
UTF	UCS（通用字符集）转换格式
UUID	通用唯一标识符
VM	虚拟机
WAN	广域网
WEB	万维网

5 ECI证书系统

5.1 引言

5.1.1 范围

ECI将**证书**用于各种各样的目的，例如**ECI主机加载程序**、**ECI客户端加载程序**和**ECI运营商证书**。本条款定义了有关这些**证书**和相关**撤销列表**的定义，以及其链组成和**根证书**结构的定义。该定义使用本建议书中规定的紧凑二进制格式，它适用于硬件实施方案和合适的密码学，以及针对未来版本和扩展的简单信令系统。

5.1.2 字段的记法和约定

下面的数据结构定义直接映射到一个字节序列上。任何加密函数都被定义为是对字节序列表示的操作。

数据定义遵循16字节和32字节字段的自然对齐方式，以简化32位CPU内核上的数据处理。填充用作通用字段，以表示此目的所需的填充字段。它使用函数填充（n_bytes），其中n_bytes是从所定义数据结构开始的字节数中的对齐边界。在解释数据结构时，应该跳过填充字段。填充字段的值应设置为0。

由另一个数据结构通过类型定义进行定义的任何字段都没有助记符。一般来说，不为这种字段提供字段长度定义。

5.1.3 扩展字段

定义的许多更实质的数据结构都有一个扩展字段，它允许添加未来的（后向兼容的）扩展。定义参见表5.1.3-1。

表5.1.3-1 – 扩展字段定义

语法	位数	助记符
Extension_Field {		
padding(4)		
length	32	uimsbf
for (i=0; i<length; i++) {		
extension_byte	8	uimsbf
}		
}		

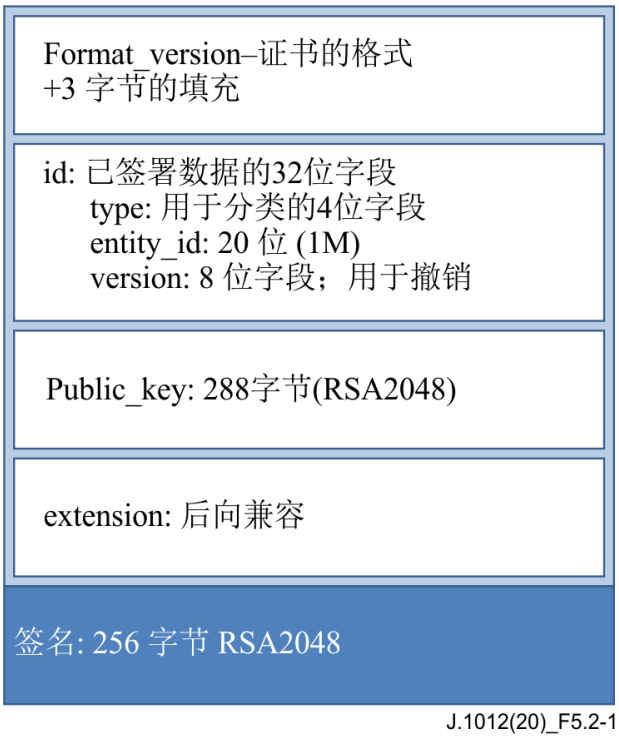
语义：

length:整数	循环之后的字节数。该值应该是4的倍数，并可以是0。
extension_byte:字节	基于未定义此字段内容的本文件版本，包含信息的数据字段可能会被实施方案忽略。

5.2 ECI证书

ECI证书有一个简单明了的结构。**证书**的ID只是一个二进制数字，仅用于机器解释，而不像互联网上使用的X.509**证书**。

一个**证书**的通用布局如22-1所示。



J.1012(20)_F5.2-1

图5.2-1 – ECI证书格式版本1

ECI证书格式在表5.2-1中定义。

任何签名的项目都应使用不同的8字节开始字段，第一个字节是已签名项目的版本格式，然后（对于版本1项目）填充3个字节，接着是第二个4字节，表示签名实体密钥情况下的唯一ID。

表5.2-1 – ECI证书定义

语法	位数	助记符
ECI_Certificate_Id {		
padding(4)		
Type	4	Uimsbf
entity_id	20	Uimsbf
Version	8	Uimsbf
}		
ECI_Public_Key_v1 {		
byte**modulus**[256]	2 048	
}		
ECI_Certificate_Data_v1 {		
ECI_Certificate_Id **id**	32	Uimsbf
Public_Key_v1**public_key**	2 304	
Extension_Field**extension**		
}		
ECI_Signature_v1 {		
byte**signature**[256]	2 048	Uimsbf
}		
ECI_Certificate {		
format_version	8	Uimsbf
if (version == 0x01) {		
ECI_Certificate_Data_v1 **data**		
ECI_Signature_v1 **signature**		
}		
}		

语义：

format_version:整数	值0x00，0x02..0xFF：保留。 值0x01：**ECI证书**格式版本1。 无法识别**证书**类型的实施方案不得处理它，并在验证请求上返回失败。
id:整数	在**证书父**（**证书**签署者）情形下唯一的、32位数字形式的**证书**标识。保留值0x00000和0xF0000-0xFFFFF。
type:整数	在签署者（**父**）情形下，类型定义实体的类型，如**制造商**、**ECI主机**、**运营商**等。类型值为0x0 ... 0x7的**证书**需要一个**撤销列表**来验证子。0x8及以上的类型值不需要**撤销列表**来验证子（参见表5.2-2）。
entity_id:整数	定义实体的编号。**entity_id**承载依据**证书**类型定义的各种子格式。除非另有定义，否则在**父**\（**证书**或**撤销列表**的签署者）情形下，entity_id是唯一的。
version	实体**证书**的版本号，按升序分配（通常以1递增）。
extension: Extension_Field	未被定义用于解释此情况的处理函数将忽略该字段中的数据。该字段可用于通用**证书**定义的特定应用中的特定数据。其解释依赖于上下文。除非明确规定允许，否则该字段不得用于非**ECI**应用。
public_key: ECI_Public_Key_v1	该**证书**的实体的公钥（由**父**指派）。
data:ECI_Certificate_Data	这是**证书**的数据部分。
signature:字节[256]	签名字段包含**证书父**签名的字节序列表示，使用附件A中定义的密码函数。

对**ECI证书**的任何验证都应包括根据字段定义的累积来验证**证书**的总长度。

通用类型值用于大多数**证书**和**撤销列表**，以确保所有分配的值都是唯一的。表5.2-2给出了所有**ECI TA**签名数据的概述。

表5.2-2 – 已签署项目的ID分配和父

父	类型	ID字段	描述
根	0x0	0xFFFFF	根
根	0x1	制造商id，<> 0xFxxxx	制造商**证书**
根	0x1	制造商RL的id， == 0xFxxxx	制造商**撤销列表**
制造商	0x0	主机id，<> 0xFxxxx	**ECI主机证书**
制造商	0x0	主机RL，== 0xFxxxx	**ECI主机撤销列表**
主机	0x8	主机图像id	**ECI主机图像**
主机	0x9	主机图像系列id	**ECI主机图像系列证书**
主机图像系列	0x9	图像目标id	**ECI主机**系列图像
根	0x2	供应商id，<> 0xFxxxx	**安全供应商证书**
根	0x2	供应商RL的id，== 0xFxxxx	**安全供应商撤销列表**
供应商	0x0	客户端id，<> 0xFxxxx	**ECI客户端证书**
供应商	0x0	客户端RL，== 0xFxxxx	**ECI客户端**和**ECI客户端**系列**撤销列表**
客户端	0x0	客户端id	**ECI客户端图像**
客户端	0x1	客户端系列id	客户系列**证书**
客户端系列	0x8	图像目标id	客户端系列图片
根	0x3	运营商id，<> 0xFxxxx	运营商**证书**
根	0x3	运营商RL的id，== 0xFxxxx	运营商**撤销列表**
运营商	0x0	平台操作ID，<> 0xFxxxx	**平台操作证书**
运营商	0x0	平台操作RL的id，== 0xFxxxx	**平台操作**RL
平台操作	0x0	平台操作客户端图像联署id，<> 0xFxxxx	**平台操作**客户端图像联署
平台操作	0x0	平台操作客户端图像联署RL的id，== 0xFxxxx	**平台操作**客户端**图像撤销列表**
平台操作或目标组	0x0	目标组的id，<> 0xFxxxx	目标组，在[ITU-T J.1014]中定义
平台操作或目标组	0x0	目标RL的id，== 0xFxxxx	目标撤销列表，在[ITU-T J.1014]中定义
平台操作或目标组	0x8	微客户端id，<> 0xFxxxx	微客户端，在[ITU-T J.1014]中定义
平台操作、出口组、第三方出口组	0x4	出口组id，<> 0xFxxxx	**出口组**
平台操作、出口组、第三方出口组	0x4	出口组RL的id，==0xFxxxx	**出口组撤销列表**
出口组	0x5	第三方出口组id <> 0xFxxxx	第三方**出口组**
出口组	0x8	出口组RL的id， == 0xFxxxx	**出口组**撤销列表
出口组、第三方出口组	0xE	出口系统id，<> 0xFxxxx	出口系统
根	0x4	出口授权运营商id，<> 0xFxxxx	出口授权运营商
根	0x4	出口授权运营商id，== 0xFxxxx	出口授权运营商**撤销列表**
出口授权运营商，出口授权	0x0	出口授权id，<> 0xFxxxx	出口授权（带有**子**）
出口授权运营商，出口授权	0x0	出口授权id，== 0xFxxxx	出口授权撤销列表
其他	其他		保留
注：ECI功能可以分别传输和处理**证书**或另一已签名**数据**项的数据字段和**签名**区段。			

5.3 ECI撤销列表

撤销列表应由最初签署撤销**证书**的同一**实体**来签署。**撤销列表**是定义其**证书**最低可接受之版本的实体标识符的一个列表。如果**撤销列表**中的条目是具有相关撤销列表的**证书**，则**撤销列表**的最低版本号将与该**证书**一起应用。**ECI撤销列表**的布局在图5.3-1中定义。

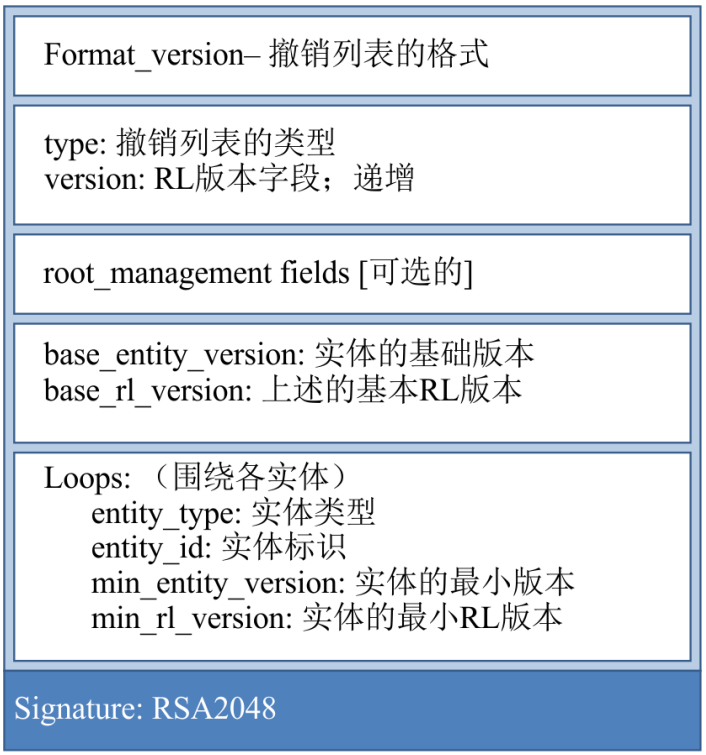


图5.3-1 – 撤销列表结构

无论数据来源如何，**ECI主机**实施方案都应为其所管理的**实体**存储最新的（在**rl_version**中定义）**撤销列表**。

撤销列表（ECI_RL）在表5.3-1中定义。

表5.3-1 – 撤销列表定义

语法	位数	助记符
ECI_RL_Id {		
padding(4)		
Type	4	Uimsbf
indicator = 0xF	4	Uimsbf
version	24	Uimsbf
}		
ECI_Revocation_List_v1 {		
base_entity_version	8	Uimsbf
base_rl_version	24	Uimsbf
number_of_entities	24	Uimsbf
for (i=0; i<number_of_entities; i++){		
entity_type	4	Uimsbf
entity_id	20	Uimsbf
min_entity_version	8	Uimsbf
min_rl_version	24	Uimsbf
}		
}		
ECI_RL {		
format_version	8	Uimsbf
if (format_version == 0x01){		
ECI_RL_Id**rl_id**	32+24	Uimsbf
root_version_indicator	1	Uimsbf
padding(1)	7	Uimsbf
root_version	8	Uimsbf
min_root_version	8	Uimsbf
padding(4)		
ECI_Revocation_List_v1**rev_list**		
Extension_Field**extension**		
ECI_Signature_v1 **rl_signature**	2 048 （见注释）	Uimsbf
}		
}		
注 – =在版本**1证书**关联的CRL中。		

语义：

format_version:整数	值0x00，0x02..0xFF：保留。 值0x01：ECI**撤销列表**格式版本1。 无法识别**证书**类型的实施方案不得处理它，并应验证请求返回失败。
type:整数	类型字段在ECI_Certificate_Id中定义，参见表5.3-1。
indicator:整数	指示**撤销列表**；值应等于0xF。
version:整数	RL的版本。从1开始（对新**证书**通常为空），并在每次更新时递增。
base_entity_version:整数	所有**id.version**小于**base_id_version**的实体都将被撤销。
base_rl|_version	版本等于**base_entity_version**且小于**base_rl_version**之实体的所有撤销列表不再有效。
number_of_entities:整数	撤销列表中的实体数量。参见表5.3-1.低于最大值。
entity_type:整数	旧版本被撤销的实体的类型。
entity_id:整数	旧版本被撤销的实体的Entity_id。
min_entity_version:整数	与**entity_type**和**entity_id**匹配的实体（**证书**ID）的最小版本号。较低版本被撤销。
min_rl_version	与实体匹配**entity_type**、**entity_id**和**entity_min_version**结合使用的撤销列表的最小版本。较低的撤销列表版本不再有效。
root_version_indicator:位	如果值等于0，则root_version和min_root_version字段应没有任何意义。如果值等于1且父是根**证书**，则**root_version**和**min_root_version**字段的解释如下。
root_version	根**证书**的版本，是该**撤销列表**的签署者。
min_root_version:整数	如果**父**（即根）版本大于或等于此字段，则将撤销小于**min_root_version**的所有根**证书**版本，以验证在**revocation_id_lead**中定义的类型的**证书**。
extension: Extension_Field	附加数据：不是设计用于解释该字段的实施方案将忽略之（不包括签名计算），除了用于计算签名。
rl_signature: ECI_Signature_v1	与**撤销列表**关联的**ECI实体**的签名。签名是在所有先前数据的基础上计算得到的。

注 – 硬件实施方案可以处理区块中的**撤销列表**，查找随后**证书**的ID，应在累积签名散列时进行验证，并在到达列表的末尾时对签名进行验证。

作为一般规则，**ECI主机**应存储验证实体（由**ECI主机**加载）所需的所有**证书**的**TA撤销列表**。**ECI主机**应使用新收到的、具有更高版本号的**撤销列表**替换存储的、**证书**或项目的**撤销列表**。

撤销列表的最大长度应符合第B.2节的规定。

5.4 证书链和撤销列表树

5.4.1 数据结构定义

一个**证书链**是一系列具有相关**撤销列表**的**证书**，其中**证书**由管理先前**证书**的实体签署。它始于父**证书**（通常是一个根）的**撤销列表**。**证书**的最小（有效）版本号和**子**的最小（有效）**撤销列表**版本由其**父**的**撤消列表**定义。链被用作**证书**来验证要加载的项目，因此**证书**通常不会出现在其前任的**撤销列表**中。尽管如此，**撤销列表**处理是强制性的，以便验证链的完整性。表5.4.1-1给出了典型**证书链**的结构。

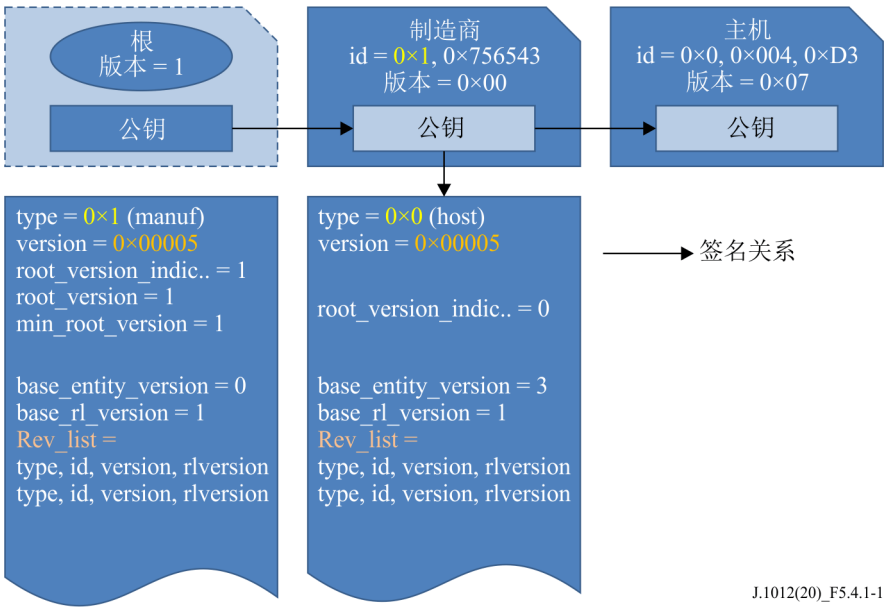


图5.4.1-1 – 主机证书链示例

链可以传输或存储，并可由不同的部分组成。

撤销列表树是链接之撤销列表的序列，它们使用前一个链中的**证书**作为**父**，因此跨越了大量认证项目。平台操作可以使用这些来否决（指示撤销）其他（**TA**撤销的）实体。**证书链**和**撤销列表**树的定义应符合表5.4.1-1的规定。

表5.4.1-1 – 证书链和撤销列表树定义

语法	位数	助记符
ECI_Certificate_Chain {		
chain_length	8	uimsbf
padding(4)		
for (i=0; i<**chain_length**; i++){		
ECI_RL **rl**		
ECI_Certificate**certificate**		
}		
}		
ECI_RL_Tree {		
ECI_RL **father_revocation_list**		
three_breadth	32	uimsbf
for (i=0; i<**three_breadth**; i++){		
father_node_depth	8	uimsbf
chain_length	8	uimsbf
padding(4)	16	uimsbf
for (i=0; i<**chain_length-1**; i++){		
ECI_Certificate**certificate**		
ECI_RL **rl**		
}		
}		
}		

语义：

chain_length:整数	链的长度。
rl: ECI_RL	在链第一次迭代的情况下，链的先前**证书**或**父**的**撤销列表**。链中**撤销列表**的标识符字段的版本号应相等。
certificate:ECI_Certificate	当前序列中的下一个**证书**的**父**。
father_revocation_list: ECI_RL	链的**父**的**撤销列表**。
three_breadth:整数	树中的子链数。
father_node_depth:整数	先前**证书链**（包括树的父）的**父证书**级别。继承的父列表是该链的父，在其**父**之前，直到树本身的**父**。

撤销列表树中**证书**的排序规则是：

- 树不得包含重复的证书。
- 树应以如下方式进行排序：即最后一张叶证书的所有兄弟应在最后一个证书后立即列为chain_length = 0子树，然后是父的兄弟的子树等。
- 兄弟证书应以树状结构中的ID顺序出现（最低优先）。

5.4.2 证书链的处理规则

ECI主机对**证书链**进行验证，并使用**高级安全系统**为撤销的项目提供适当的响应。**证书**和**撤销列表**验证的关键安全步骤由安全的**高级安全系统**执行。**高级安全系统**还为**ECI客户端**提供能力，以便随后验证所用链撤销版本号的有效性。

ECI主机可以在迭代过程中处理**证书链**。这从**ECI TA**根**撤销列表**开始，并以链中的最终项目结束。**证书链**处理在任何中间检查失败时都会失败。如果**ECI主机**在某种条件下出现故障，在对撤销的实体或无效**证书**上触发**ECI主机**策略措施之前，它应确保现有的**证书**和**撤销列表**以及所有先前的**撤销列表**和**证书**均通过其签名经过了验证。[ITU-T J.1014]和[ITU-T J.1015]中定义的**高级安全系统**应确保**证书链**处理保持适当的稳健性。

任何处理顺序都是允许的，只要它在接受链时产生相同的结果。

1) **ECI主机**应对**撤销列表**执行以下验证步骤：

   a) ECI主机应验证撤销列表format_version字段与它可以解释的版本匹配，且确保rl_id.type和rl_id.rl_indicator字段匹配期望值。

   b) **ECI主机**应验证**撤销列表**的长度是否与其字段值相符。

   c) 如果**root_version_indicator** = 1，则**ECI主机**应检查链处理中此时是否期望根为**父**，检查**root_version**是否存在，用于验证和检查**min_root_version**是否未超过目前在链处理中使用的任何根版本。

   d) **ECI主机**应验证该撤销列表是否未被该**撤销列表**的最低版本号（自链中先前**撤销列表**）弄失效，或者在根**撤销列表**的情况下，被链处理中目前使用的min_root_revocation_list号弄失效。

   e) **ECI主机**应使用**父证书**的公钥验证**撤销列表**的签名。

   f) 如果能够这样做，则**ECI主机**应处理**撤销列表**中的任何扩展字节。

   g) **ECI主机**应验证链中的下一个<实体类型、实体id、版本>是否未根据**撤销列表**被撤销，并建立适用于该**证书**的最小撤销列表版本。

2) **ECI主机**应对下一个**证书**执行以下预验证步骤：

a) **ECI主机**应验证**证书**的版本。如果版本与其处理能力不匹配，则无法加载链。

b) **ECI主机**应验证证书ID的类型字段，如果与期望值不匹配，则失败。

c) **ECI主机**应验证**证书**的长度是否与其格式定义相匹配。

d) **ECI主机**应使用**父证书**的公钥验证**证书**的签名。

e) 如果能够这样做，则**ECI主机**应处理**证书**中任何额外的字段与/或扩展字节。

从**撤销列表**树中提取的**撤销列表**链可用于验证需要通过**高级安全系统**加载的特定项目的撤销情况。这样的项目可以通过**证书**的id序列（用于在将其加载到**高级安全系统**中时对其进行验证）来识别。**撤销列表**链的默认处理规则应与**证书链**的默认处理规则相同。

3) **CPS**应在CPS中加载当前的**撤销列表**和下一个**证书**的<实体类型、实体id、版本>。CPS应执行以下验证：

a) CPS应检查撤销列表的format_version字段，以匹配它可解释的版本，以及rl_id.type和rl_id.rl_indicator字段，以匹配预期值。

b) 如果**父**是**根证书**（**root_version_indicator** = 1），则CPS应选择具有**root_version**的根证书作为**父**，否则使用预加载的**证书**或先前的证书。

c) CPS应使用**父证书**的公钥来验证**撤销列表**的签名。

d) CPS应验证**撤销列表**的长度是否与其字段值相符。

e) CPS应验证**撤销列表**的版本号是否未失效。

f) CPS应验证链中的下一个<实体类型、实体id、版本>是否未根据**撤销列表**被撤销，并应建立随该**证书**一起提供的最小撤销列表版本。

4) 然后，**ECI主机**应将**证书**加载到适当的CPS处理位置中，它将执行以下验证：

a) CPS应检查**撤销列表**的**format_version**字段，以匹配它可以解释的版本，以及**id.type**和**id.entity_id**字段，以匹配预期值。

b) CPS应验证**证书**的长度是否与其字段值相符。

c) CPS应对**父证书**的公钥进行签名验证。

5.5 撤销树组和撤销数据文件

用于验证特定**实体**的撤销数据应选择包含目标**实体**之**父**的RL的撤销数据。

分发撤销数据时，撤销多个目标实体的链可以组合成一棵树，从而避免重复的根和**子证书**及其关联的**撤销列表**，并允许在**CPE**中进行更有序的搜索。

为了便于组装和拆卸撤销数据，撤销树也可以简单地组合成一个树集。不过，除了对共同**父**撤销列表，树集不得重叠。树集可以包含多个根RL（在正在进行的**TA**根变更展开期间）。

证书链和**撤销列表**树的定义应符合表5.5-1的规定。

表5.5-1 – 撤销列表树集定义

语法	位数	助记符
ECI_RL_Tree_Set {		
tree_number	32	uimsbf
for (i=0; i<tree_number; i++) {		
ECI_RL_Tree**tree**		
}	8	uimsbf
}		

语义：

tree_number:整数	集合中的树数
tree:ECI_RL_Tree	**证书**树（包括**根证书**）及其**撤销列表**。

注 – 在线服务器可以分发单个**实体**目标树（有效链），以最大限度地减少数据流量。在广播网络上，树可以很容易地拆分和合并，以匹配传输轮播中使用的桶数（见第7.7.2节）。

就包含一个类的所有实体而言，撤销树或树集不需要是完整的。由**平台操作**视情组成撤销树集，确保**平台操作**网络中部署的**CPE**风险最小。在广播网络上，**撤销列表**也可以及时进行更换，以扩大撤销范围。

ECI要求**CPE**永久存储所有可能被加载项目的**ECI TA**链，以便确保一旦撤销，实体仍保持被撤销状态。这在相关的条款中有规定。

为了便于传输，**ECI**撤销树集以表5.5-2中给出的格式进行分组。

表5.5-2 – 撤销数据文件

语法	位数	助记符
ECI_revocation_data_file {		
magic = 'ERD'	24	uimsbf
version	8	uimsbf
father_type	4	uimsbf
sub_type	4	uimsbf
ECI_RL_Tree_Set**revocation_data**		
}		

语义：

magic:字节[3]	幻数用于验证以下数据的格式。它有字符“ERD”三个8位ASCII表示的值。**ECI主机**应检查该字段的值，以验证**ECI**文件是否具有用于其他数据完整性的预期格式。
version: 字节	图像标题的格式版本。值0x01是当前定义的版本；所有其他值都保留。**ECI主机**应忽略任何版本号未被认可的图像。
father_type:整数	**撤销列表**数据的共同**父**的类型。0x0表示**ECI根证书**。值0x1-0x7保留。值0x8-0xF可用于私有应用程序。
sub_type: 整数	对于father_type字段等于0x0的情况，它根据表5.2-2（包含在撤销数据中的数据的**ECI根证书**）来定义公共**撤销列表**的类型。对于father_type的其他值未定义该值。
revocation_data: ECI_RL_Tree_Set	已撤销项目的撤销列表的撤销列表树集。

5.6 大的数据项签名

对大的数据散列，**ECI**使用高效的散列函数并结合常规签名操作来对大的数据项（例如软件图像）上的签名进行计算。在表5.6-1中定义了大的数据元素的签名。

表5.6-1 – 定义大的数据元素的签名

语法	位数	助记符
ECI_Data_Signature {		
sign_version	8	uimsbf
padding(4)	24	uimsbf
if (sign_version == 0x01){		
for (i=0; i<256; i++){		
signature_byte	8	uimsbf
}		
}		
}		

语义：

sign_version:整数	签名版本。值0x01为当前版本；所有其他版本值均予保留。尚未实施版本的**CPE**应忽略此字段（以及任何后续数据）。
signature_byte:字节	表示大项目签名的字节序列。

签名算法在附件A中定义。

5.7 根证书

5.7.1 根证书的定义

ECI使用一系列**根证书版本**。**ECI TA**可以从使用一个新的**根证书**版本开始，例如，当任何子的任何先前**撤销列表**过大时，或者当与**证书**公钥相关的密钥不再被认为足够保密时。

根证书使用**ECI证书**的标识符字段，其字段定义在表5.7-1中给出。类型和标识符字段从不使用；只用了版本字段。

表5.7-1 – ECI Root_ID字段的定义

语法	位数	助记符
ECI_Root_Id {		
type /* see Table 5.2-1*/	4	uimsbf
id /* see Table 5.2-2*/	20	uimsbf
version	8	uimsbf
}		

语义：

version:整数	**证书**的版本号；编号从1开始，并且每颁布一个新的**根证书**，编号将递增1。值0x00被保留。

5.7.2 ECI主机根证书管理

ECI TA可以从使用一个具有更高版本号的新的**根证书**开始。然后，它可以在其后的某个时间为新的**根证书**发布**撤销列表**，撤销之前的**根证书**。这会使由这样一个根签名的所有**证书**无效。

或者，**ECI TA**可以决定用于特定类型实体（例如**制造商**）的**撤销列表**太大，并通过在该类型**实体**的**撤销列表**中使用更高的**min_id_version**字段，来决定重新发布所有先前颁发之**证书**的新版本。这实际上使该类型**实体**先前颁发的所有**证书**无效，最高为**min_entity_version**-1。典型地，这需要为仍使用较低**证书**版本的实体颁发具有较高版本号的大量新**证书**来替换已被撤销的**证书**。

ECI主机应提供用于存储**根证书**的资源在[b-ITU-T J Suppl. 7]中提议。

6 ECI主机加载程序

6.1 引言

ECI主机加载过程区分以下几个方面：

1) 存储图像，由**CPE**使用**ECI TA**提供的认证数据和随后激活的图像来验证图像的真实性。
2) 包含图像的文件的文件格式以及将图像加载到**CPE**所需的所有其他信息。
3) 传送**ECI主机图像**到**CPE**的传输协议。这包括**CPE**发现所需图像的位置。它包括存储所传输的图像、互补的**ECI**验证链以及签名数据。
4) 任何**运营商**特定的**ECI主机**图像撤销；在第6节中定义了此类信息的数据格式；在第8节中定义了应用程序。

在每次重启**CPE**时，以及在**CPE**正常运行期间需要这么做时，认证和图像验证的逻辑应适用于新下载的**ECI主机图像**和验证数据。

6.2 存储、验证和激活

6.2.1 操作原则

ECI主机确保**ECI客户端**可以在私有和无篡改的环境中运行，以符合实施这些客户端的**ECI**稳健性要求。**ECI主机**还可以防止一个**ECI客户端**与另一个**ECI客户端**的干扰。为此，**ECI TA**可以为**CPE**认证软件，**CPE**加载程序应验证其加载的软件图像的真实性。

许多**CPE**使用多级加载程序。**ECI**假定核心**CPE**芯片在开始加载任何常规软件图像之前加载许多芯片特定的初始化图像。这些图像可能会根据**ECI TA**芯片供应商的许可协议进行隐式认证。或者，它们可以成为本条款中定义之**制造商**认证过程的一部分。

如果其中一个**ECI**管理图像的软件稍后显示出存在安全漏洞，则**ECI TA**和**CPE制造商**可撤消之，并将之替换为带漏洞修复的版本。

在图6.2.1-1中，假设**Img1**是芯片特定的图像，需要将芯片置入可以开始加载更多常规应用程序图像的状态。它受芯片特定签名**CS1**的保护，**芯片加载程序**使用芯片供应商专有密钥对其进行验证。

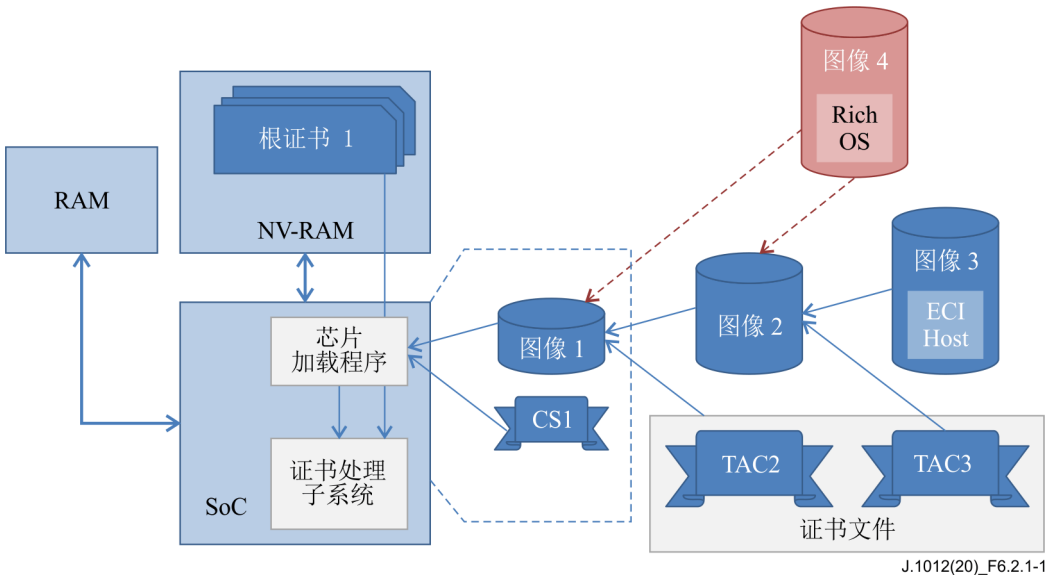


图6.2.1-1 – ECI主机加载过程示例

一旦**Img1**运行，芯片继续加载其他图像。它加载**Img2**，可以通过**证书链**和图像签名**TAC2**进行验证。使用**TA根证书**、**证书处理子系统**和**TAC2**来执行对图像的验证。**Img2**继续加载包含**ECI主机**软件的**Img3**。**Img3**通过**证书处理子系统**、**根证书**、信任机构**证书链**和图像签名**TAC3**进行验证。如果加载环境可以保证这不会对**ECI主机**造成安全隐患，则可以加载其他图像，例如，包含未通过**ECI TA**认证的富OS实例的**Img4**。

图像的信任机构证书由特殊的证书文件承载。

ECI TA认证**ECI主机**的安全完整性：它提供客户端私密性、**ECI主机**外部威胁的防篡改能力以及确保客户端相互产生不良干扰的能力。**CPE制造商**可能希望使用补充的安全措施来加载图像，使用其专有的图像加密和认证。

平台操作可以验证**ECI主机图像**的新鲜度并决定不解密服务。为此，CPS提取用于验证任何加载项目的最小**撤销列表**版本号，从而允许**平台操作**验证最近**撤销列表**的应用。第8节中定义了针对**ECI主机**的、这些**平台操作**特定的验收程序。

ECI主机加载程序应将最新的**ECI主机图像**及其最新的证书存储在NV-RAM中。**ECI主机加载程序**将在重启**ECI主机**时重新验证其加载的每个图像。此过程在每次重启时重建**ECI主机**的真实性。

6.2.2 证书定义

6.2.2.1 ECI主机图像相关证书

ECI针对**ECI主机**图像分集提供两种类型的**ECI CPE**：

1) 通用**CPE**将在相同**CPE**类型和版本的每个实例上加载相同的**ECI主机图像**集。
2) 个性化的**CPE**将在相同CPE类型和版本的每个**CPE**上加载（部分）不同的图像集。这样一系列相同“类型”但依据**CPE**个性化的图像被表示为**图像系列**。

ECI主机证书链由以下**证书**组成（每个证书均通过其前身进行认证）：

1) 根证书：
   – 这是中心**ECI** TA根**实体**的表示。该**证书**的公钥应用于验证。

2) 制造商证书：

– 这是针对特定**制造商**的**ECI TA实体**的表示。该**证书**的公钥应用于验证。

3) 主机证书：

– 这是**ECI TA**认证的**CPE**硬件和**ECI主机**软件版本的表示。对于通用**ECI主机**，该**证书**的公钥应用于认证所有**ECI主机图像**。对于“个性化”的**ECI主机图像**，该**证书**的公钥应用于验证。

4) 主机图像系列证书：

– 该**实体**为特定于某个特定**CPE**配置的一系列图像提供通用批准，但从**ECI TA**角度来看，这些图像在其他方面是相同的。对于个性化的**ECI主机**，该**证书**的公钥应用于验证针对特定**CPE**的**ECI主机图像**，其**CPE** ID与**证书**中的标识符相匹配。

注 – 每个实体标识符必须在授权实体情形下予以解释；即ID是相对的。

ECI主机图像和相应的认证结构如图6.2.2.1-1所示，表6.2.2.1-1给出了相关参数的概述。

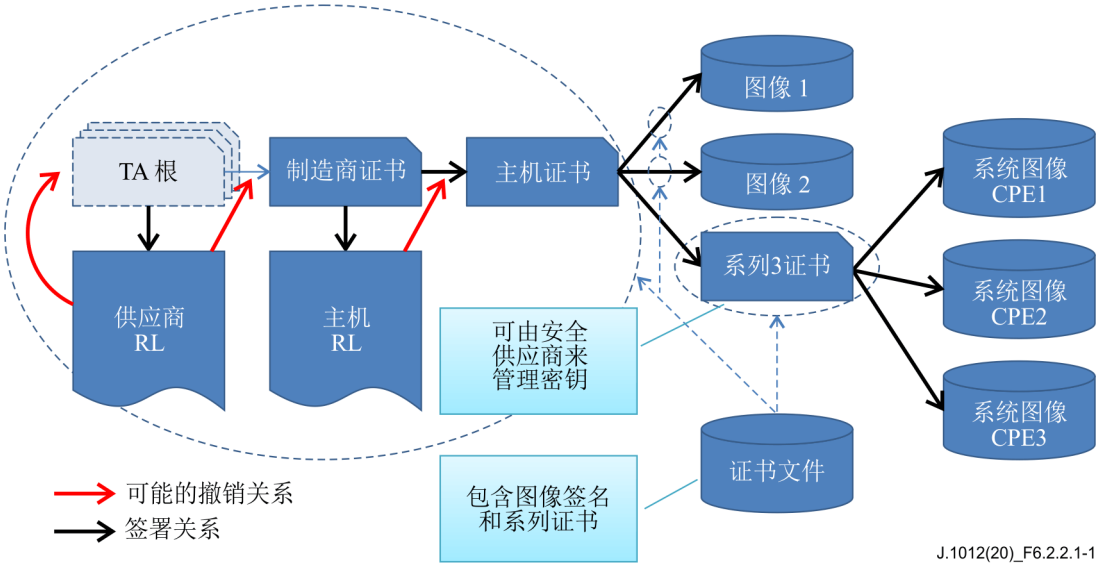


图6.2.2.1-1 – ECI主机图像认证结构

表6.2.2.1-1 – ECI主机相关证书参数概述

类型	实体	证书ID字段值	由ECI主机进行的特定处理
0x0	**制造商**	manufacturer_id, version	应根据AS块中**CPE**的制造商ID，来对Manutacturer_id进行检查。
0x0	主机	cpe_type, cpe_model, host_version	应通过AS块来检查**CPE**类型和**CPE**模型的cpe_type和cope_model。
0x8	CPE**图像系列**	target_id	应根据**CPE**的身份来检查target_id。
0x8	CPE图像	n.a.	
0x8	**ECI主机图像**	ECI_Host_Image_Id	这是实际图像签名的类型。

ECI主机相关**证书**的**证书**定义应符合第5.2节中定义的通用**ECI_Certificate**定义。表6.2.2.1-2给出了**ECI主机**管理**证书**的标识符字段的定义。

表6.2.2.1-2 – 主机相关证书的ID字段定义

语法	位数	助记符
ECI_Manufacturer_Id {		
padding(4)		
type /* see table 5.2-2 */	4	uimsbf
manufuacturer_id	20	uimsbf
Version	8	uimsbf
}		
ECI_CPE_Type_ID {		
cpe_type	12	uimsbf
cpe_model	8	uimsbf
}		
ECI_Host_Id {		
padding(4)		
type /* see table 5.2-2 */	4	uimsbf
ECI_CPE_Type_Id**cpe_type_id**	20	uimsbf
host_version	8	uimsbf
}		
ECI_Host_Image_Series_Id {		
padding(4)		
type /* see table 5.2-2 */	4	uimsbf
image_series_model	8	uimsbf
image_series_model_extension	4	uimsbf
image_series_version	16	uimsbf
}		

语义：

type	依据表5.2-2的值。
manufacturer_id:整数	ID由**ECI TA**分配给**制造商**。
cpe_type:整数	ID由**ECI TA**分配给**CPE**模型。值0x000和0x3F0..0x3FF保留。相同模型的**CPE**应具有较大的共同点，并使用相同的**ECI**安全技术。
cpe_model:整数	ID分配给某个特定模型的一个版本，该版本在很多方面都是相同的，但有一些不小的差异。值由**ECI TA**分配。值0x00和0xF0..0xFF保留。
cpe_type_id:ECI_CPE_Type_id	**CPE**硬件类型的ID（版本+模型）；在manufacturer_id情形下是唯一的。
cpe_host_version	将ID分配给某个图像集，构成**CPE**的**CPEECI主机**配置。
image_series_model:整数	支持**序列图像**之**CPE**的相同类型图像的ID，区别在于**cpe_id**。值0x000和0xF00..0xFFF保留。
image_series_version:整数	Id由**ECI TA**递增地分配给**图像系列**模型的版本。值0x0000和0xF000..0xFFFF保留。

6.2.2.2 ECI主机图像签名

ECI主机图像ID应等于主机图像系列ID，它在表6.2.2.2-1中定义。

表6.2.2.2-1 – 主机图像ID和主机系列图像ID定义

语法	位数	助记符
ECI_Host_Image_Id {		
padding(4)		
type /* see table 5.2-2 */	4	uimsbf
image_model	8	uimsbf
image model_extension	4	uimsbf
image_version	16	uimsbf
}		
ECI_CPE_Id {		
cpe_serial_number	28	uimsbf
cpe_type	12	uimsbf
manufacturer_id	20	uimsbf
}		
ECI_Image_Target_Id {		
padding(4)		
targettype	4	uimsbf
if (target type == 0x1){		
ECI_CPE_Id **cpe_id**	60	uimsbf
}		
}		

语义：

Type	依据表5.2-2的值。
image_model:整数	将Id分配给某**ECI主机图像**或相互替换的系列图像。值0x00和0xF0.0xFF保留。
image_model_extension:整数	以上字段的扩展。在常规应用中，该字段应设置为0x0。
image_version:整数	递增分配的相同类型图像的版本。值0x00和0xF0..0xFF保留。
cpe_serial_number:整数	图像所针对的**CPE**的序列号。cpe_serial_number在<manufacturer_id，cpe_type_id>情形下应是唯一的。
cpe_type:整数	在ECI_CPE_Type_Id结构中定义的cpe_type字段。
manufacturer_id:整数	manufacturer_id字段在ECI_Manufacturer_Id结构中定义。
target type: 整数	系列图像的目标识别类型。值0x1定义了该结构的定义，并指示cpe_id用作目标，保留其他值。
cpe-id: ECI_CPE_Id	作为系列（**ECI主机**或**ECI客户端**）图像之目标的**CPE**的ID。

用于签署实际**ECI主机图像**的**ECI主机图像签名**和**ECI主机图像**系列签名应使用第5.5节中定义的大的数据签名结构。

6.2.2.3 ECI主机证书

表6.2.2.3-1定义了用于验证**ECI主机**图像集的**ECI主机证书**结构。

表6.2.2.3-1 – ECI主机证书结构定义

语法	位数	助记符
ECI_Host_Credentials{		
image_credential_version	8	uimsbf
if (image_credential_version == 0x01) {		
padding(4)	24	uimsbf
ECI_Certificate_Chain **image_chain**		
nr_images	8	uimsbf
padding(4)	24	uimsbf
for (i=0; i<images; i++){		
ECI_Host_Image_Id**image_id**	32	uimsbf
if (image_id.type == 0x8) {		
ECI_Certificate**series_cert**		
} else if (image_id.type == 0x9){		
ECI_Data_signature **image_signature**		
}		
}		
Extension_Field**extension**		
}		
}		

语义：

image_credential_version:字节	证书的格式版本。值0x01是当前定义的版本；所有其他值都保留。**ECI主机加载程序**应忽略值非其所认可的任何证书，。
image_chain: ECI_Certificate_Chain	开始于**制造商根RL**的2级深度**证书链**，直至**ECI主机证书**。最后一张**证书**应用于验证任何图像系列**证书的图像签名**。
nr_images:整数	包含签名的图像的数量。
image_id	循环中签名随后的图像的ID。 循环中列出的image_id应具有不同的**image_id.image_model**字段值。
series_cert:ECI_Certificate	用于验证**图像系列**的证书。
image_signature: ECI_Data_Signature	图像的签名（包括主机图像Id）。
extension: Extension-Field	后向兼容的扩展字段。

在验证**image_chain**时，**CPE**应遵守在第5.4节中定义的链的通用处理规则。

6.2.3 ECI主机图像文件的加载过程

CPE应存储、验证和激活执行启动**ECI主机**所需的**ECI主机图像**文件集。**ECI主机图像**的实际激活通常发生在**CPE**启动时。

CPE应使用称为**ECI主机加载**程序的强大处理功能来下载、验证和激活选定的**ECI主机图像**。例如，如果包含**ECI主机加载**程序的**CPE**启动图像开始执行第二个图像，并且第二个图像加载并开始执行第三个图像，则实际上适当加载第三个图像（执行图像签名验证）的第二个图像的功能应被视为该**CPE**的**ECI主机加载**程序功能。只有**ECI主机加载**程序功能可以验证并启动**ECI主机图像**。**ECI主机加载**程序应使用**证书处理子系统**（CPS）来验证图像证书。

CPE应将最新的**ECI主机图像**文件集及其下载的证书存储在**NV存储器**中。在**CPE**启动时，**ECI主机加载程序**应能定位这些图像，并以适合特定**CPE**类型的方式开始加载图像。

使用**CPS**的**ECI主机加载**程序应使用第5.4节中的常规链处理规则来验证每个加载的图像。通用图像和**图像系列证书**应使用主机证书公钥进行验证。**图像系列证书**公钥应用于验证图像系列图像，并且**CPE**应根据**CPE**的cpe_id来验证图像中的cpe_id。

如果图像受损（因CPS引起的签名检查失败），则**ECI主机加载程序**应拒绝图像，这意味着**CPE**将无法实例化**CPE**上的**ECI主机**。**CPE**应能从这种情况中恢复：它应具有恢复过程来重新初始化最新的**ECI主机图像**及其证书，例如，通过从其在线**ECI主机图像**服务器或通过其他方式重新加载来自广播信道的最新**ECI主机图像**文件集。

ECI主机应存储它获取的最新版本的**ECI主机**链**证书**，而不管其获取的信道。实际上，这会“锁定”最新的可用主机**证书**，作为未来图像验证的基础。

加载**ECI主机图像**的序列并不直接通过签名验证过程进行验证：这应由引导加载程序为第一个**ECI主机图像**来执行，并由先前的**ECI主机图像**自身进行后续的激活。

6.3 ECI主机相关的文件格式

本建议书没有为**ECI主机图像**文件定义任何文件命名或其他元属性。它以数据容器集（**ECI**明智的无名文件）的形式管理**ECI主机图像**数据，这些数据容器由其主机图像ID标识，**ECI**证书（**证书链**和签名）用于验证这些数据。

一个**ECI主机图像**文件应该是ECI_Host_Image_Header和图像内容的一个序列。它应遵循在表6.3-1中定义的定义。

表6.3-1 – ECI主机图像文件定义

语法	位数	助记符
ECI_Host_Image_File {		
magic = 'EHI'	24	
image_header_version	8	uimsbf
if (image_header_version == 0x01) {		
ECI_Host_Image_Id**host_image_id**	32	uimsbf
ECI_Manufacturer_Id**manufacturer_id**	32	uimsbf
Extension_Field**extensions**		
for (i=0; i<n; i++) {		
host_image_byte	8	uimsbf
}		
}		
}		

语义：

host_image_byte:字节	实际的**ECI主机图像**；**CPE**专有的格式。
magic:字节[3]	幻数用于验证以下数据的格式。它具有字符“EHI”三个8位ASCII表示的值。**CPE**固件应检查该字段的值，以验证**ECI**文件是否具有用于其他数据完整性的预期格式。
image_header_version:字节	图像标题的格式版本。值0x01是当前定义的版本；所有其他值都保留。
host_image_id:ECI_Host_Image_Id	**ECI主机图像**的图像ID。在加载（新的）**ECI主机图像**之前，**CPE**应检查该字段。
manufacturer_id:ECI_Manufacturer_Id	**ECI主机图像**的**CPE**制造商的ECI_Manufacturer_ID。在加载（新的）**ECI主机图像**之前，**CPE**应检查此字段。参见注释。
extensions: Extension_Field	参见本建议书的第5.1节：后向兼容的扩展。
host_image_byte:字节	实际的**ECI主机图像**。
注： 这也应该与在广播轮播中使用的**制造商**OUI相对应，以承载关联的文件。	

图像系列文件有一个唯一的签名，它承载于图像文件本身中。因此，特定的文件格式应遵循表6.3-2中给出的定义。

表6.3-2 – ECI主机图像系列文件定义

语法	位数	助记符
ECI_Host_Image_Series_File {		
magic = 'EHS'		
image_header_version	8	uimsbf
if (image_header_version == 0x01) {		
ECI_Data_Signature **image_signature**		
ECI_Image_Target_Id**target_id**	64	
Extension_Field**extensions**		
for (i=0; i<n; i++) {		
host_image_byte	8	uimsbf
}		
}		
}		

语义：

host_image_byte: 字节	实际的**ECI主机图像**；**CPE**专有的格式。
Magic: 字节[10]	幻数用于验证以下数据的格式。它具有字符“EHS”三个8位ASCII表示的值。**CPE**固件应检查该字段的值，以验证ECI文件是否具有用于其他数据完整性的预期格式。
image_header_version:字节	图像标题的格式版本。值0x01是当前定义的版本；所有其他值都保留。
image_signature:ECI_Data_Signature	对图像文件中后跟的所有数据签名。
target_id:ECI_Series_Image_Target_Id	图像的目标ID。target_id.target_type的值为0x01，所有其他值都保留。
extensions: Extension_Field	参见第5.1节：后向兼容的扩展。
host_image_byte:字节	形成主机**图像**的字节序列。

ECI主机图像证书遵循表6.3-3中的定义，它实质上是带有图像签名集或**图像系列证书的证书链**。

表6.3-3 – ECI主机图像证书文件定义

语法	位数	助记符
ECI_Host_Image_Credential_File{		
magic = 'EHC'	24	uimsbf
version	8	uimsbf
if (version == 0x01) {		
ECI_Host_Credentials **credentials**		
}		
}		

语义：

magic	幻数用于验证以下数据的格式。它具有字符“EHC”三个8位ASCII表示的值。**CPE**固件应检查该字段的值，以验证**ECI**文件是否具有用于其他数据完整性的预期格式。
version	文件的格式版本。值0x01是当前定义的版本；所有其他值都保留。
credentials:ECI_Host_Credentials	一个或一组**ECI主机图像**的证书。

host_image_id用于识别**ECI TA**签名，用于**ECI主机**图像文件集，它由**ECI**证书结构中的完全下载构成。

允许符合**ECI**的**CPE**使用与用于**ECI主机图像**文件相同的传输协议来下载其他专有的**CPE**软件模块。没有任何这种图像所需的特定格式。

在广播媒质上，将多个**ECI主机**的撤销数据作为一个大的文件进行分发很方便。接收这些数据的**ECI主机**可以使用它来检查其自身的**ECI主机证书**。

ECI主机撤销数据文件使用表5.5-2中定义的ECI_Revocation_Data_File格式。**ECI主机**撤销数据文件使用等于0x0（**根证书**）的ather_type以及等于**制造商**撤销列表类型的sub_type。revocation_data符合树叶撤销列表为**ECI主机**撤销列表的约束要求。

6.4 ECI主机图像传输协议

6.4.1 引言

本建议书区分了三种类型的主机**图像**传送：

1) **广播：ECI**定义协议以允许**平台运营商**使用DVB-SSU从**CPE制造商**向现场的**CPE**指示并传送新的**ECI主机图像**文件。
2) **在线：ECI**允许互联网连接的**CPE**使用任何专有协议下载**ECI主机图像**文件，建议使用HTTP 1.1以及使用**ECI**定义的接口来连接运营商提供的万维网服务器。
3) **其他：CPE制造商**与/或**运营商**也可以使用其他方式来传送**ECI主机图像**文件，包括离线方式，如通过USB棒传送。这种图像传输方式超出了本建议书的讨论范围。尽管如此，载有这种协议的图像应符合第6.2节和第6.3节中的文件格式和图像验证要求。

旨在从数字广播网络获取**服务**的**CPE**应执行第6.4.2节中定义的**ECI主机图像**广播传输协议。

带有IP连接的**CPE**应执行第6.4.3节中定义的在线**ECI主机图像**互联网传输协议以及第7.7.3.3节中定义的协议。

CPE可以实现任何互补的**ECI主机图像**传输协议，包括**ECI主机**广播和离线传输协议（例如USB棒）。在所有情况下，**CPE制造商**都应确保实际的方法，通过结合上述传输协议，可以在现场更新**ECI主机**，同时考虑到某些网络连接未实现连接的实际使用情况。

6.4.2 ECI主机广播传输协议

6.4.2.1 概述和配置

ECI主机广播传输协议允许新的**ECI主机图像**文件和相关数据通过**运营商**广播前端基础设施从**CPE制造商**传输到**CPE**。该协议还允许传输非**ECI主机图像**文件（用于非安全关键功能）。**运营商**可以在管理**CPE**上的软件版本方面发挥积极的作用。该协议通过为**CPE制造商**与**运营商**之间的技术互操作点设定标准来促进合作：

- 从**CPE制造商**到**运营商**的下载数据自愿标准移交；

注 – 此类移交的技术细节超出了**ECI**规范的讨论范围。

- 标准广播传输协议（在**运营商**广播前端实现单一播出规定）；以及
- 接收器中的标准发现、传输协议实现和操作传输协议参数选择。

ECI主机广播传输流（TS）和**CPE**实施方案应符合DVB SSU [ETSI TS 102 006]的要求，并因此而符合DVB数据轮播定义[ETSI EN 301 192]有关章节、实施指南[ETSI TR 101 202]和MPEG数据轮播定义[ISO/IEC 13818-6]的要求。

运营商和**CPE**都应支持DVB-SSU简单配置文件；并可选择地支持DVB-SSU UNT配置文件。

运营商可支持多个同时轮播。

CPE将为相关的下载项目扫描所有的轮播，在SI、UNT（合适的话）和PMT中予以适当地指令。

图6.4.2.1-1描绘了下载图像的整体广播方案。

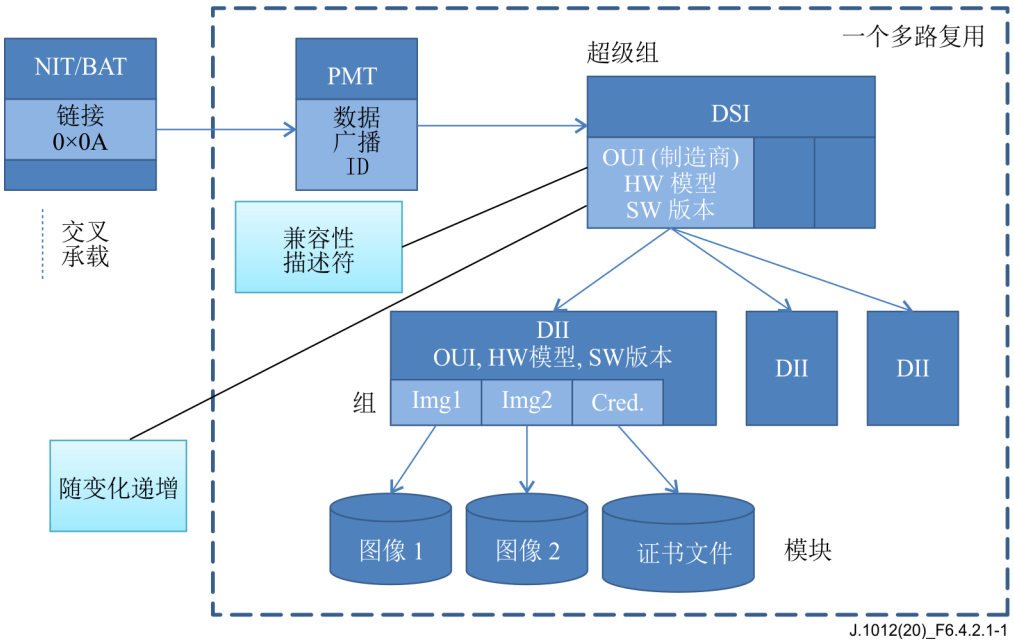


图6.4.2.1-1 – 主机图像信令和轮播结构概述（非UNT变体）

6.4.2.2 CPE制造商到运营商的移交

任何未来基于**ECI**的生态系统都需为**运营商**和**CPE制造商**定义一个指导方针，以提供统一的图像文件信息交换方式（包括**ECI主机**和非**ECI主机图像**）、有关从（许多）**CPE制造商**到（许多）**运营商**下载的**ECI**图像证书和元信息。

6.4.2.3 DVB SI信令

6.4.2.3.1 下载位置信令

在所有NIT（地面或电缆）或BAT表（卫星）中，**运营商**都应以最低限度的通用DVB OUI（即，所有轮播非**制造商**特定的链接）支持DVB-SSU链接描述符（链接类型0x09）。

简单配置文件**CPE**应支持DVB-SSU链接描述符（链接类型0x09）。

在所有NIT（地面或电缆）或BAT表（卫星）中，支持DVB-SSU UNT配置文件的**运营商**都应支持SSU扫描链接描述符（链接类型0xA）。

UNT配置文件**CPE**应支持DVB-SSU扫描链接描述符（链接类型0x09）。

6.4.2.3.2 紧急更新

为了表明需要紧急替换**ECI主机图像**，可以在NIT、BAT或其中一个SDT条目中放置一个或多个ECI_host_emergency_download描述符，供标记**ECI主机**可提供访问的服务使用。**ECI主机**应能够从任何当前调谐的多路复用中其出现的任何表中检索该描述符，执行关联的处理，并使用任何备用调谐器访问相关多路复用，以便在通电状态下，在最坏情况下可有30分钟的间隔时间来获取该描述符。建议更频繁地检查非调谐多路复用（间隔3分钟）。

ECI_host_emergency_download_descriptor允许针对特定的操作平台和特定的**平台操作**以及客户端图像，以便最大限度地减少紧急更新可能对之造成任何干扰的**用户**数量。

当**ECI主机**发现一个新的ECI_host_emergency_download描述符时，它应将其**ECI主机**和**ECI客户端**配置与描述符中的目标信息进行匹配。如果找到目标匹配且当前安装的主机图像的版本需要更新，则**ECI主机**应根据emergency_indicator执行此更新。这将导致**CPE**中正在进行的**用户**活动的中断。

ECI操作描述符是一个DVB专用描述符，并应始终位于出现它的表格之前，通过**ECI**private_data_specifier_field的DVB private_data_specifier_descriptor（参见[ETSI EN 300 468]和[ETSI TS 101 211]）。描述符的语法在表6.4.2.3.2-1中定义。

表6.4.2.3.2-1 – ECI_host_emergency_download_descriptor

语法	位数	助记符
ECI_host_emergency_download_descriptor{		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
/* main loop */		
main_loop_nr	8	uimsbf
for (i=0; i<main_loop_nr; i++){		
/* client loop */		
client_nr		
for (j=0; j<client_nr; j++){		
platform_operation_tag	8	uimsbf
Reserved	3	
client_flag	1	
client_tag	4	uimsbf
}		
/* host image loop */		
host_nr	8	uimsbf
for (j=0; j<host_nr; j++){		
Reserved	4	
emergency_indicator	4	uimsbf
manufacturer_id	20	uimsbf
cpe_type_id	20	uimsbf
min_host_version	8	uimsbf
}		
}		
/* private data till end of descriptor*/		
for (i=0; i<n; i++){		
private_data_byte	8	
}		
}		

语义：

descriptor_tag	descriptor_tag的**ECI**专用标签值：参见[b-ITU-T J Suppl. 7]。
descriptor_length	参见[ETSI EN 300 468]。
main_loop_nr	主循环中的条目数。单独的主循环条目应由**ECI主机**单独评估，即具有OR语义。一个循环条目中的各种元素都应具有AND语义。
client_nr	客户端目标循环中的条目数；值0x00意味着任何客户端都将匹配。单独的循环条目应具有OR语义，并且所有匹配的客户端都应被考虑用于紧急更新。一个循环条目的各字段都应具有AND语义。
platform_operation_tag	**ECI平台操作**的标签值，如NIT/BAT中ECI_platform_operation_descriptor中列出的那样。如果platform_operation与其中一个已安装客户端的platform_operation匹配，则**ECI主机**应考虑进行紧急更新。
client_flag	指示client_tag字段是否与匹配相关。值=0b0表示不相关（即，任何client_id将匹配），值=0b1表示client_tag相关。
host_tag	NIT/BAT中ECI_platform_operation_descriptor中列出的、标识**ECI主机**的标签值，它与同一客户端循环条目中的platform_operation_tag字段相匹配。如果引用的vendor_id和client_id与**ECI主机**中安装的客户端之一（用于**平台操作**）相匹配，则**ECI主机**应考虑进行紧急更新。
host_nr	主机循环中的条目数。最小值应为1。循环条目应具有OR语义；即，如果任何主机规范与目标条件匹配，则主循环具有匹配状态。
emergency_indicator	**ECI主机**应使用该字段的值来选择适当的行为，以启动下载和主机的后续更新，如表6.4.2.3.2-2中所定义。
manufacturer_id	由紧急更新定位的主机Manufacturer_id。如果该字段的值与**ECI主机**的manufacturer_id匹配，则**ECI主机**应考虑进行紧急更新。
cpe_type_id	值由表6.2.2.1-2中的ECI_CPE_Type_ID定义。如果主机的cpe_type_id与该字段的值匹配，则**ECI主机**应考虑紧急更新。cpe_type_id.cpe_type等于0x000意味着任何**ECI主机**cpe_type都是匹配的（并且cpe_model和host-version应被忽略）。cpe_type_id.cpe_model等于0x00意味着任何任何**ECI主机**cpe_model都是匹配的（并且主机版本应被忽略）。
min_host_version	当且仅当其主机版本小于或等于该字段中的值时，**ECI主机**才会考虑紧急更新。 注 – 字段值等于0xFF意味着所有主机版本都匹配。
private_data_byte	私有数据：内容可由负责管理该描述符广播的**运营商**来定义。

表6.4.2.3.2-1定义了需满足的主循环中的诸多条件（具有AND语义），以便**ECI主机**考虑执行紧急更新。如果符合所有这些条件，则**ECI主机**应根据该**ECI主机**的emergency_indicator字段执行紧急下载并安装新的主机图像。指示符字段值在表6.4.2.3.2-2中定义。

表6.4.2.3.2-2 – ECI_host_emergency_download_descriptoremergency_indicator字段值

名称	值	描述
系统紧急	0×01	**ECI主机**应尽快下载并安装新的主机图像，以便在需要时中断正在进行的**用户**激发的活动。参见注释。
常规紧急	0x03	**ECI主机**应第一时间下载并安装新的主机图像，这不会对任何**用户**激发的活动造成任何干扰。**ECI主机**应在下一次启动事件期间下载最新的主机图像。 注 – **平台运营商**可以使用此功能，例如，在当前的**ECI主机**在解密服务方面存在严重缺陷但对常规使用情况可以合理执行时。
RFU	其他	保留以供未来使用。
注：如果当前的**ECI主机**与目标平台/客户端组合结合使用时存在严重的性能问题，则**平台运营商**可以对实例使用此能。		

6.4.2.4 PSI信令

对传输的每个轮播，**运营商**应支持PMT [ETSI EN 300 468]中data_broadcast_id_descriptor，但不要求在该描述符的选择器字节中支持任何OUI信令。

SSU简单配置文件**CPE**应使用data_broadcast_id_descriptor来定位承载DVB-SSU轮播的流的PID。

6.4.2.5 UNT选项

本条款仅适用于支持UNT配置文件的**CPE**和**运营商**。

在PMT中，应使用data_broadcast_id_descriptor，包含system_software_update_info结构，其中update_type为0x2，OUI字段设置为DVB OUI 0x00015A。

运营商应为其支持的每个**CPE**类型，在其中一个SSU表中承载一个SSU表格条目。

ECI主机应能够解释以下UNT描述符（参见[ETSI TS 102 006]）：

- SSU_location_descriptor（如果正在广播一个**CPE**类型的轮播）。
- Scheduling_descriptor（如果在可预见的未来计划一个**CPE**类型的轮播）。
- Message_descriptor .

CPE应能够始终如一地成功下载几乎没有错误的接收轮播，该轮播在名义上公布的时间进行安装和拆卸，并执行两个完整的周期（在轮播中重复所有的消息），条件是没有任何**用户**启动的、会干扰下载的活动。

6.4.2.6 轮播结构

ECI DVB SSU轮播（详情参见[ETSI TS 102 006]）应使用两层数据轮播。

ECI DVB SSU轮播应使用具有以下限制的DSI消息：

- 应有完整的、有关所有可用组的列表，以便下载。
- 每个组都应对应一个**制造商**的一个**cpe_type + cpe_model**，并包含用于**CPE**类型的**ECI主机**的所有资源。这意味着最多可以有255个模块（图像文件）可用（加上一个证书文件）。

注1 – 由于ECI_host_id.model_id值的限制，限值为239。

- GroupInfoIndication结构GroupCompatibility字段中的CompatibilityDescriptor（有关详细信息，请参见[ETSI TS 102 006]）应使用以下约定：
  - 循环应包含一个系统硬件描述符：
    - OUI应对应**CPE**的**制造商**。
    - 与系统硬件描述符相关的模型和版本字段应对应CPE的cpe_type和cpe_model，并等于组的证书文件中的ECI主机证书的id.cpe_type和id.cpe_model字段。
  - 循环应包含一个系统软件描述符；模型字段应设置为0，版本字段应反映组中总的**ECI主机**软件的版本（即**ECI主机**和非**ECI主机图像**）。

CPE应使用compatibilityDescriptor中的模型和版本字段来匹配其自身的**CPE**模型和**CPE**版本，并应使用软件版本字段来检查组是否包含一个更新，以及是否有新的版本继续下载新图像。

ECI DVB SSU轮播应使用具有以下限制的DII消息字段：

- blockSize应至少设置为值2KB（2048字节）。
- “tDownloadScenario”字段应赋予一个有意义的值，以反映所有模块的下载速度至少为最慢消息重复时间的4倍（轮播周转时间）。
- moduleId位7..0应等于图像文件的**id.image_model**。
- moduleVersion等于图像文件的**ECI id.image_version**。

CPE可以使用“tDownloadScenario”字段来终止无法成功的下载（例如由于高分组错误率），并将该问题报告给**用户**。

CPE类型的组应包含以下模块：

- **CPE**类型的图像文件（可能是部分图像集）。
- **ECI主机图像**证书文件，包含**ECI主机**所有图像的（最新）证书：
  - 该模块应将DII的moduleId位7..0设置为0xFF；以及
  - moduleVersion应在每次更改时递增。

注2 – 允许**运营商**通过在DII之间共享DownloadDataBlocks而在各种各样**CPE**类型的下载之间共享通用文件。不过，这意味着需要在CPE类型之间连贯地管理**ECI主机图像**ID。

6.4.2.7 ECI主机下载操作

如果网络访问资源可用，并且至少每隔6小时处于待机状态而不干扰**用户**，**ECI主机图像**加载程序应尝试每隔30分钟检查所有可能的轮播，例如，在将**CPE**切换到待机状态之后以及在非高峰观看时间期间。

如果网络提供商使UNT可用于承载CPE类型的潜在下载，则相应的**CPE**应定期检查UNT是否有可能进行新的更新。**CPE**应尝试使用与**ECI主机**图像轮播相同的频率条件进行检查。

如果仅有广播模式的**CPE**被阻止执行上述检查超过2周，则建议**用户**收到一个警告。

一旦检测到新下载的可用性，表示**CPE**和**用户**已提供批准，则**CPE**将尝试执行下载并安装新的图像（可能会覆盖之前的版本）。应向**用户**适当报告成功执行下载中出现的任何持续失败。**ECI主机**应始终能够从故障主机图像下载中恢复并恢复到一个功能状态，例如，通过恢复先前的主机图像或尝试重加载新的主机图像。

应该注意的是，持续无法下载新的**ECI主机图像**或证书可导致**运营商**拒绝服务。

6.4.2.8 运营商轮播时间表

运营商应为**CPE**图像数据轮播提供足够的带宽，以便在合理的时间内执行下载。

6.4.2.9 用户接口问题

一个能够通过广播网络执行**ECI主机图像**下载的**CPE**应：

- 具有下载扫描操作模式，它将定期自动检查新图像或证书的可用性；例如，作为待机状态的一部分，建议将此作为下载检查的默认**制造商**设置；以及
- 在**CPE**菜单中有一项设置，它将自动执行任何**用户**批准，以接受新的**ECI主机图像**文件或证书，并建议将其作为默认**制造商**设置，以便批准下载。

CPE应至少提供一种下载新**ECI主机图像**文件的替代方法，以防止**CPE**在广播网络中运行，它们不为其遭遇拒绝服务的**CPE**类型提供新的**ECI主机图像**文件。

6.4.3 ECI主机互联网传输协议

6.4.3.1 IP协议

ECI没有为**CPE**定义一个特定的协议，来对来自制造商提供之服务的新**ECI主机图像**文件进行检查。不过，建议使用HTTP1.1 [IETF RFC 7231]作为文件传输协议，并且可以使用第7.7.3.3节中定义的协议，它为来自**平台操作**服务器的**ECI主机图像**文件定义了标准化的下载服务。

通常，**ECI主机图像**下载服务器由**CPE制造商**提供。通过**CPE制造商**与**运营商**（或代表它们的第三方）之间的特定安排，这些也可能由**运营商**或第三方来提供。

6.4.3.2 在线加载程序操作

ECI在线**ECI主机图像**加载程序应每30分钟尝试检查一次其在线服务器，而不打扰**用户**。如果仅有在线模式的**CPE**被阻止长时间执行上述检查，则建议**用户**收到一个警告。

一旦检测到新下载的可用性，**CPE**将尝试执行下载并安装新的图像（可能会覆盖之前的图像版本）。此类下载的任何持续失败应向**用户**进行适当报告。

应该注意的是，未能下载新的**ECI主机图像**或证书可导致**运营商**拒绝服务。

CPE在线加载程序应提供第6.3节中定义的（新）图像和图像证书集，用于验证、存储和激活。

ECI在线主机图像加载程序应提供紧急下载功能，具有第6.4.2.3.2节为广播定义的相同效果。

6.4.4 替代传输协议

ECI主机可以使用任何其他（专有的）交付协议。

CPE加载程序应对第6.3节中定义的一组（新的）图像和图像证书进行处理，以便用于验证、存储和激活。

7 ECI客户端加载程序

7.1 引言

ECI主机可以下载、存储和激活**ECI客户端**图像和相关数据。**ECI客户端**加载过程可以按以下步骤进行分解：

1) 发现基于ECI的服务/服务包保护与/或其他识别**ECI客户端**需求的方式。这是**CPE**常规导航应用程序的一部分。
2) 确定在**ECI主机**上安装**ECI客户端**所需资源的网络位置（广播或在线）。
3) 下载并存储（在NV存储器中）安装**ECI客户端**所需的**平台操作**信息，并验证证书。
4) 用**平台操作**的安全系统注册**ECI主机**，并接收（如果需要的话）用于解密**ECI客户端**的**CPE**特定的初始化数据。

5) 从网络下载并存储（在NV存储器中）**ECI客户端**图像和关联的**ECI客户端**证书，并验证证书和图像，存储在NV存储器中以供未来使用。

6) 使用**ECI客户端**图像、**平台操作证书**初始化**ECI客户端**，分配ECI容器和所需的AS资源，并开始执行**ECI客户端**。

除了**运营商**注册**CPE**之外，所有处理都可以使用来自广播流或互联网的数据来执行，只有在广播连接可用的情况下才需要人工协助。

运营商可以随时通过在广播或在线网络上发布信息来更新**ECI客户端**资源。**ECI主机**会定期检查这些更新。

ECI需要用于**CPE**各种各样功能的支持数据，例如，撤销数据或者**ECI客户端**与/或**ECI主机**所需的、更新的**证书链**，以便能够支持**ECI客户端**。在广播网络上，传输协议允许基于数据标识的“索引”（散列）选择性地下载**CPE**所需的数据。通过索引散列对数据分组被称为“桶化”。在在线网络上选择性下载基于将所需数据的标识作为参数传递给万维网服务API。

以下数据项目可通过**ECI主机**下载：

- **ECI客户端**图像（在广播网络上以桶化格式）。
- **ECI客户端**撤销数据（在广播网络上以桶化格式）。
- 平台操作客户端链。
- **平台操作**撤销数据（在广播网络上以桶化格式）。
- **ECI主机图像**撤销数据（在广播网络上以桶化格式）。**ECI AS**设置用于解密加密客户端图像的客户端初始化数据（在广播网络上以桶化格式）。

7.2 ECI客户端的发现

7.2.1 引言

典型地，符合**ECI**的**CPE**（例如一个iDTV）在出厂时将未安装任何**ECI客户端**，因为该设备可能在全球范围内的任何市场上销售。以下条款定义了允许符合**ECI**的**CPE**找到**ECI客户端**的可用机制，这些客户端可能需要对其连接之网络提供的服务进行解扰。

对于发现过程，可以区分两种类型的网络：

1) 基于传输流的网络（广播和典型的IPTV网络）。

2) 基于IP协议的网络。

对基于传输流的网络，**ECI**支持两种提供商和客户端发现模式：

1) 手动安装 – 包括基本的（广播）网络设置参数。

2) 自我发现（带用户选项） - 这假设**CPE**可以自动地为网络实现自动安装。

基于传输流的网络上的手动安装和自我发现协议都使用通用信令。

对于基于IP协议的网络，**ECI**支持基于手动的URL输入。

7.2.2 基于传输流的网络

7.2.2.1 常见信号

为了减少**用户**手动输入参数，**ECI**提供了用于安装客户端的、关键ECI参数的在线信令：

- NIT中的一个或多个ECI_platform_operation_descriptor依据**Platform_Operation**承载可用的客户端（按ID）。描述符包括平台提供商的名称和short-id（以便允许手动安装字符串中的紧凑表示形式）。
- 平台提供上可以在ECI_base_URL_descriptor中为万维网API指定基本URL。

7.2.2.2 ECI_platform_operation_descriptor

ECI_platform_operation_descriptor提供关于**平台操作**的关键信息，该操作为基于传输流的网络提供访问服务。

对于每个**Platform_Operation**，$NIT_{实际}$（与/或卫星网络上的BAT）应至少将ECI_platform_operation_descriptor承载于安装字符串中标识的中心多路复用和表上，用于仅提供手动安装的网络，以及除卫星网络之外的所有多路复用上，用于提供自我发现的网络。卫星网络只允许在提供商承载服务的多路复用设备上承载ECI_platform_operation_descriptor：作为NIT或BAT的一部分。

ECI_platform_operation_descriptor 是 DVB 私有描述符，在 DVB private_data_specifier_descriptor [ETSI TS 101 162]中使用**ECI**的私有数据说明符。它在表7.2.2.2-1中定义。

表7.2.2.2-1 – ECI_platform_operation_descriptor

语法	位数	助记符
ECI_platform_operation_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
platform_tag	8	uimsbf
operator_id	20	uimsbf
platform_operation_id	20	uimsbf
platform_name_length	8	uimsbf
/* platform name loop */		
for (i=0; i<N; i++){		
platform_name_char	8	uimsbf
}		
for (i=0; i<N; i++){		
extension_byte	8	uimsbf
}		
}		

语义：

descriptor_tag	descriptor_tag的**ECI**专用标签值。参见[b-ITU-T J Suppl. 7]。
platform_tag	该8位字段为手动安装指定了**Platform_Operation**的标签。在支持每个**Platform_Operation**的各网络NIT和BAT上都应具有唯一的platform_tag值。每个platform_tag在各NIT或BAT中都只应出现一次。platform_tag不得用于排序提供商，并且不得出现在用于**Platform_Operation**选择的**CPE用户**接口中。
operator_id	运营商ID，如本建议书第7.5.2节所定义。这是**Platform_Operation运营商**的标识符。
platform_operation_id	本建议书第7.5.3节中定义的**Platform_Operation** ID。
platform_name_length	平台名称循环的八位字节序列的长度。如果长度为0，则提供商不支持自我发现，并且不应在**CPE**客户端安装菜单中的任何提供商选择菜单中列出。该字段的最大值应为40。
platform_name_char	表示平台操作名称的UTF8字符序列。
extension_byte	其他字节；保留以供本建议书未来使用。

7.2.2.3 ECI_base_url_descriptor

ECI_base_url_descriptor允许**Platform_Operation**发信号通知其万维网API的基本URL（见第7.7.3节），在联机访问的情况下可用于提供与客户端安装相关的服务。

对于每个**Platform_Operation**，$NIT_{实际}$（与/或卫星网络上的BAT）可以在承载ECI_platform_operation_descriptor的同一个表中承载ECI_base_url_descriptor。

ECI_base_url_descriptor是一个DVB私有描述符，在DVB private_data_specifier_descriptor [ETSI EN 300 468]中使用**ECI**的私有数据说明符。它在表7.2.2.3-1中定义。

表7.2.2.3-1 – ECI_base_url_descriptor

语法	位数	助记符
ECI_base_url_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
platform_tag	4	uimsbf
reserved	4	
base_url_length	8	uimsbf
/* base url loop */		
for (i=0; i<N; i++){		
base_url_char	8	uimsbf
}		
}		

语义：

descriptor_tag	descriptor_tag的**ECI**专用标签值。参见[b-ITU-T J Suppl. 7]。
platform_tag	该4位字段指定用于手动安装的供应商的标签。在支持每个**Platform_Operation**的每个网络NIT和BAT上都应具有唯一的platform_tag值。每个platform_tag在每个NIT或BAT中都只应出现一次。platform_tag不得用于排序**Platform_Operations**，并且不得出现在用于**Platform_Operation**选择的**CPE用户**接口中。
base_url_length	该字段应指示基本URL循环中的八位字节数。
base_url_char	构成平台操作基本URL的UTF8字符序列。

7.2.2.4 手动安装

Platform_Operation可以为用户提供一个安装字符串，**用户**可以将其输入到**CPE用户**接口的一个适当安装菜单项中，以便安装一个**ECI客户端**。安装字符串应按照本条款进行定义。安装字符串是一个可变长度的二进制数字的数字表示。最高有效位第一表示中的二进制数可以通过连接最高有效位第一表示中数字的3位二进制值来构造。

该号码以4位数字块的形式呈现给**用户**，并且**CPE** UI上的条目应同样代表4位数字块。

标识参数的安装字符串在表7.2.2.4-1中定义。

表7.2.2.4-1 – 安装字符串参数（以位数为单位）

参数	DVB-T/DVB-T2	DVB-C/DVB-C2	DVB-S/DVB-S2	IPTV	助记符
网络类型	3	3	3	3	uimsbf
网络ID	16	17	17	16	uimsbf
平台标签	8	8	8	8	uimsbf
客户端标签	4	4	4	4	uimsbf
填充	0	0	0	0	uimsbf
校验和	5	5	5	5	uimsbf
位数	36	36	36	36	uimsbf
数字数	12	12	12	12	uimsbf
块数	3	3	3	3	uimsbf

语义：

网络类型 3位字段。网络类型的值参见表7.2.2.4-2。

网络ID 包含ECI_service_provider_descriptor的DVB SI Table-id（见第7.2.2.2节）提供了访问服务所需的详细信息，如表7.2.2.4-3所定义。

平台标签 4位字段，代表NIT或BAT中ECI_service_provider_descriptor中所需服务提供商的提供商标签。

客户端标签 4位字段，代表NIT或BAT中提供商标签选择的ECI_service_provider_descriptor中所需客户端的提供商标签。

填充 0..2位字段，值为0，将前一个字符串填充为3位的倍数。

校验和 通过添加前一个字符串的连续5位块而形成的5位字段。字符串的最后部分用额外的前导零填充，长度为5位。例如。字符串0b01011010的校验和为0b01011 + 0x00010 = 0b01101。校验和应由CPE的用户接口使用，以拒绝**用户**的任何错误条目。

表7.2.2.4-2 – 网络类型值表示

网络类型	值
DVB-T/T2	0
DVB-C/C2	1
DVB-S/S2	2
IPTV	3
保留	4..7

表7.2.2.4-3 – 网络ID表示

网络类型	网络ID值	位数
DVB-C	0b0，后跟NIT表的网络ID，或者 0b1，后跟BAT表的BAT ID。	17 17
DVB-S/S2	0b0，后跟NIT表的网络ID，或者 0b1，后跟BAT表的BAT ID。	17 17

7.2.2.5 自我发现安装

对于这种安装方法，**CPE**应该能够自我发现基于传输流的网络的网络参数，从而能够访问网络上的所有传输流。

每个多路复用中的每个服务都将使用**ECI Platform_Operations**标签进行标记，以便提供对该服务的访问。这可以逐个业务地在SDT中进行（见第7.2.2.6节），或者逐个复用地在NIT或BAT中进行（仅适用于卫星网络）（见第7.2.2.6节）。

CPE应向**用户**提供安装**Platform_Operation**的任何**ECI客户端**的选项，作为自我发现安装过程的一部分。如果**用户**因为希望通过相关接入网络接收解密服务而决定安装**Platform_Operation**的**ECI客户端**，则**CPE**默认行为应该是在**CPE**的中央服务列表中安装标记为该**平台操作**的所有**服务**。

7.2.2.6 ECI服务标签描述符

ECI_service_tag_descriptor承载于SDT。它为每个服务标记提供解扰服务的**ECI**服务提供商。定义参见表7.2.2.6-1。

表7.2.2.6-1 – ECI业务标签描述符

语法	位数	助记符
ECI_service_tag_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
platform_tag	8	uimsbf
}		

语义：

descriptor_tag descriptor_tag的**ECI**专有标签值。参见[b-ITU-T J Suppl. 7]。

platform_tag 这是**ECI Platform_Operation**的platform_tag值，如ECI_platform_operation_descriptor中列出的那样，可以携带在网络的NIT或BAT中。

7.2.2.7 ECI平台列表描述符

ECI平台列表描述符提供了**ECI Platform_Operations**的列表，它提供对网络中不同多路复用的服务的访问。ECI_platform_list_descriptor承载于NIT与/或BAT中。定义参见表7.2.2.7-1。

表7.2.2.7-1 – ECI_platform_list_desciptor

语法	位数	助记符
ECI_platform_list_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for (i=0;i<N;i++){		
platform_count	8	uimsbf
/* plaform loop */		
for (j=1; j<M; j++){		
platform_tag	8	uimsbf
}		
service_count	16	uimsbf
/* service loop */		
for (j=0; j<M; j++){		
service_id	16	uimsbf
}		
}		
}		

语义：

descriptor_tag descriptor_tag的ECI专用标签值。参见[[b-ITU-T J Suppl. 7]]。

platform_count 8位字段是以下循环中提供商标签的数量。

platform_tag 这是ECI Platform_Operations的platform_tag值，如ECI_platform_operation_descriptor中所列，它承载于网络的NIT或BAT中。在服务循环中标记之Platform_Operation关联的服务跟随。Platform_tag值允许多次出现在该描述符的外部循环中。

service_count 16位字段，表示以下循环中service_id的数量。

service_id 可以使用之前平台循环中提到的平台的访问服务，来访问NIT或BAT多路复用中的服务的DVB服务ID。

7.2.3 基于IP网络的客户端发现

7.2.3.1 手动安装

可以访问IP网络的**CPE**应提供手动URL输入选项，以允许安装服务提供商。URL将用作万维网API的基本URL。

注 – 作为**CPE**应用功能的一部分，其中一些可以下载，**CPE**可以提供对各种各样在线服务的访问。**CPE**可以提供服务提供商和客户端安装API接口，以便**用户**自动执行客户端安装过程。

7.2.3.2 基于网页的安装

这种用于安装**ECI-Client**的解决方案超出了本建议书的讨论范围，并可能受制于补充规范。

7.3 存储、验证和激活

7.3.1 一般更新策略

ECI支持项目的频繁更新，以便实现高水平的完整性。因此，对所有下载的项目都会进行定期检查，以便更新。以下下载更新策略将适用于所有**ECI客户端**和**平台操作**数据以及随附的撤销数据。

ECI主机应尝试检查，以便定期更新，并在需要采取任何行动时通知**用户**。在[b-ITU-T J Suppl. 7]中提出了有关更新策略的详细要求。

ECI主机应将**平台操作客户端链**与相关的**ECI客户端**一起存储。存储和删除应作为安装和删除**ECI客户端**的一部分进行管理。

ECI主机应自动更新平台提供商**证书**并覆盖旧版本。

7.3.2 ECI客户端图像下载和存储

作为管理**ECI客户端**相关资源的一部分，**ECI主机**应只有在（隐性的）用户批准后，存储访问NV内存中服务或内容所需的**ECI客户端图像**。安装**ECI客户端**的任何自动策略都应提供一种用户透明的方法，来处置任何资源限制问题，以便以一种对**用户**透明的且不会导致意外丢失对内容或服务访问的方式来管理**ECI客户端**。相应地，任何**ECI客户端图像**删除都应（隐性地）由**用户**批准。

ECI主机应基于**平台操作**，以其原始证书，将下载的**ECI客户端**存储在NV存储器中。新的**ECI客户端**版本（仅包括新的证书）将覆盖旧的版本（基于**平台操作**）。示例：如果两个**平台操作**使用相同的**ECI客户端**类型但使用不同的版本，则两个版本都应通过**ECI主机**进行存储。

依据**ECI客户端**时隙，在[b-ITU-T J Suppl. 7]中提出了有关**CPE**可存储的最小图像大小。

7.3.3 ECI客户端验证和激活

ECI主机应在**高级安全系统**中载入**平台操作证书**的最新版本（通过版本号）**平台操作客户端链**，并尝试按照第5.4.2节中定义的处理链的通用规则，来安装**平台操作**公钥。

ECI主机应在**高级安全系统**中加载最新的**ECI客户端**。它应在**高级安全系统**中加载**平台操作**客户端共同签名。随后它应根据第5.5节处理链的通用规则来验证**ECI客户端**，并验证**ECI客户端图像**的签名和共同签名。如果发生撤销，则**ECI主机**应通知用户。

如果验证过程已成功完成，则只能安装并激活新的**ECI客户端**。

7.4 ECI客户端链结构格式

7.4.1 ECI客户端链结构格式简介

图7.4.1-1描绘了**ECI客户端证书链**的结构。该链从供应商**撤销列表**开始，随后是**安全供应商证书**、**ECI客户端撤销列表**，最后是**ECI客户端图像**文件。在**图像系列**的情况下，则会引入一个额外的**ECI客户端图像**证书。**ECI平台操作**客户端签名为客户端图像提供了第二个签名，以确保**ECI客户端**适用于平台操作。它在第7.5节中定义。

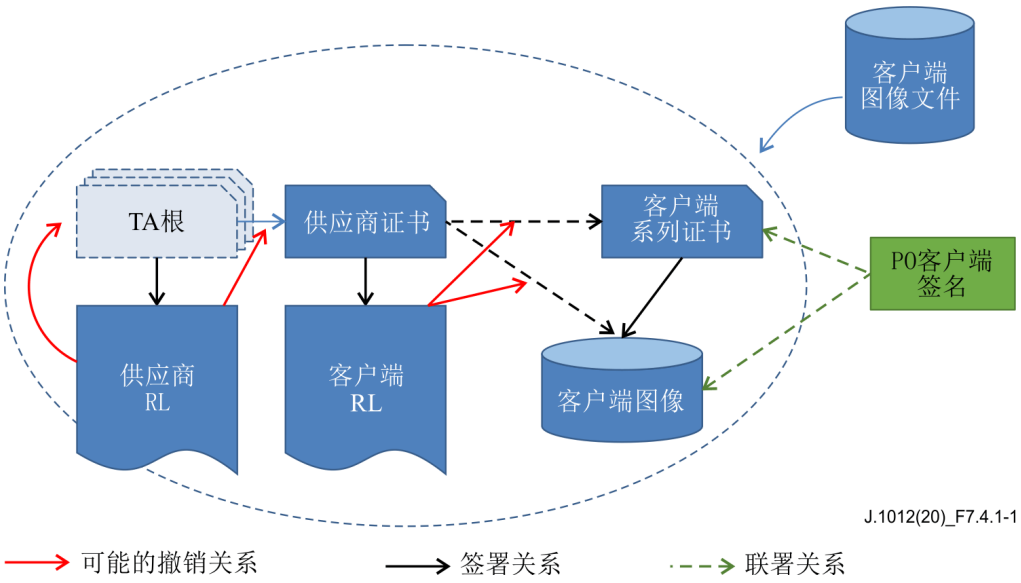


图7.4.1-1 – 客户端认证链

7.4.2 安全供应商证书

安全供应商证书由ECI_Certificate结构定义。表7.4.2-1中定义了**安全供应商证书**的证书ID。

表7.4.2-1 – 安全供应商ID定义

语法	位数	助记符
ECI_Vendor_Id {		
padding(4)		
type /* see Table 5.2-2 */	4	uimsbf
vendor_id	20	uimsbf
vendor_version	8	uimsbf
}		

语义：

type：整数	按照表5.2-2的值。
vendor_id：整数	分配给安全供应商的供应商编号，在**ECI**情形下是唯一的。
vendor_version：整数	递增分配给安全供应商**证书**版本的ID。值0x00和0xF0..0xFF保留。

7.4.3 ECI客户端系列证书和系列目标ID

ECI客户端系列**证书**由**ECI_Certificate**结构来定义。表7.4.3-1中定义了**安全供应商证书**的证书ID。

表7.4.3-1 – 客户端系列ID定义

语法	位数	助记符
ECI_Client_Series_Id {		
padding(4)		
type /* see Table 5.2-2 */	4	uimsbf
client_type	12	uimsbf
client_version_major	8	uimsbf
client_version_minor	8	uimsbf
}		

语义：

type：整数	按照表5.2-2的值。
client_type：整数	**ECI客户端**类型在**ECI客户端**安全供应商ID的情形下是唯一的。
client_version_major：整数	**ECI客户端**类型的**ECI客户端**的主版本号。新的主版本的版本递增（见注释）。
client_version_minor：整数	**ECI客户端**的次版本号。**ECI客户端**可以通过**ECI客户端撤销列表**中的次版本号比较来撤销，并自动予以替换。
注： 由于只有小的版本更新会自动触发，因此关于大的版本更改的**ECI客户端**替换在**ECI**兼容的**CPE**中不是自动完成的。	

注 – **ECI客户端**类型系列**证书**被分配给**ECI客户端**，这些客户端需要为每个**CPE**定制实施，从安全和功能角度来看是相同的。

使用ECI_Host_Series_Image_Target_Id结构，以与**ECI主机**相同的方式定义客户端目标ID。这将客户端图像绑定于一个特定的**ECI主机**。

7.4.4 ECI客户端图像签名

ECI客户端签名应使用第5.6节中定义的ECI_Data_Signature结构。

ECI客户端ID在表7.4.4-1中定义，结构上与表7.4.3-1中定义的ECI_Client_Series_Id相同。

表7.4.4-1 – 客户端ID定义

语法	位数	助记符
ECI_Client_Id {		
padding(4)		
type /* see Table 5.2-2 */	4	uimsbf
client_type	12	uimsbf
client_version_major	8	uimsbf
client_version_minor	8	uimsbf
}		

语义：

type：整数	按照表5.2-2的值。
client_type：整数	客户类型，由**ECI TA**分配。
client_version_major：整数	**ECI客户端**类型的**ECI客户端**大的版本号。为新的大的发布递增版本。
client_version_minor：整数	**ECI客户端**小的版本号。**ECI客户端**可以通过**ECI客户端**撤销列表中小的版本号的比较来撤销。

7.5 ECI平台操作链格式

7.5.1 概述

在图7.5.1-1中，介绍了**平台操作证书**和**平台操作**客户端签名的认证链。它从**运营商撤销列表**开始，然后是**运营商证书**、**平台操作撤销列表**，最后是包含**平台操作**公钥的**平台操作证书**。这与**平台操作**客户端**撤销列表**结合使用，以验证允许用于**平台操作**的**ECI客户端图像**。

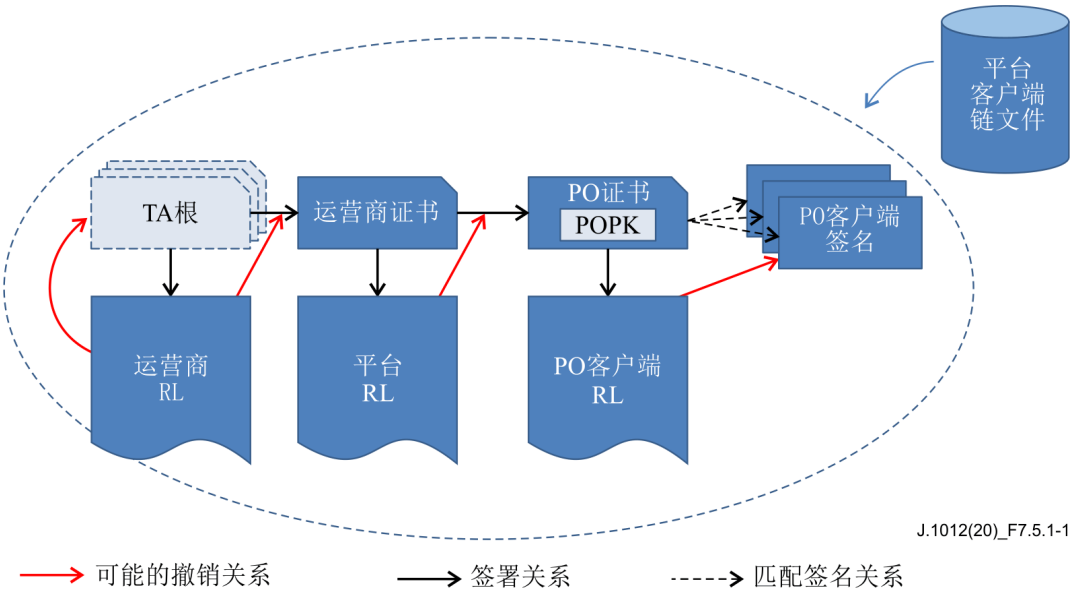


图7.5.1-1：平台客户端链的认证链

7.5.2 运营商证书

运营商证书通过**ECI_Certificate**结构定义。**运营商**的id在表7.5.2-1中定义。

表7.5.2-1 – 运营商ID定义

语法	位数	助记符
ECI_Operator_Id {		
padding(4)		
type /* see Table 5.2-2	4	uimsbf
operator_id	20	uimsbf
operator_version	8	uimsbf
}		

语义：

type：字节	按照表5.2-2的值。
operator_id：整数	分配给**运营商**的**运营商ID**，在**ECI**根的情形下是唯一的。
operator_version：整数	递增地分配给**运营商证书**版本的版本号。值0x00和0xF0..0xFF保留。

7.5.3 平台操作证书

平台操作证书通过**ECI_Certificate**结构定义。**平台操作**的秘钥由**平台操作**来管理。**平台操作证书**的证书ID在表7.5.3-1中定义。

表7.5.3-1 – 平台操作ID定义

语法	位数	助记符
ECI_Platform_Operation_Id {		
padding(4)		
type /* see Table 5.2-2	4	uimsbf
platform_operation_id	20	uimsbf
platform_operation_version	8	uimsbf
}		

语义：

type：字节	按照表5.2-2的值。
platform_operation_id：整数	分配给安全供应商的**平台操作**号，在运营商**证书**的情形下是唯一的。
platform_operation_version：整数	如果**平台操作**更改其**证书**，则递增。

7.5.4 平台操作客户端撤销列表

平台操作客户端撤销列表是使用标识符分配定义的第5.3节，如表5.2-2所定义。撤销列表中的entity_id字段指的是**平台操作**客户端签名数据结构的cosignature_id字段。

最小撤销列表版本号被定义为**ECI客户端**初始化的一部分，并使用高级安全系统来验证。

7.5.5 平台操作客户端共同签名

平台操作客户端共同签名提供**平台操作**签名，以验证客户端图像是否允许为平台提供访问服务。此外还提供图像的供应商和客户端ID，以便轻松匹配相关的客户端图像。**平台操作**客户端签名具有自己的标识符枚举；这允许使用**平台操作**客户端撤销列表来独立撤销先前允许的**ECI客户端图像**。详情参见表7.5.5-1。

表7.5.5-1 – 平台操作客户端联合签名定义

语法	位数	助记符
ECI_PO_Cosignature_Id {		
padding(4)		
type	4	uimsbf
entity_id	20	uimsbf
version	8	uimsbf
}		
ECI_PO_Client_Cosignature_Data {		
ECI_PO_Cosignature_Id **cosignature_id**	32	
client_tag	4	uimsbf
reserved	28	
ECI_Vendor_Id **vendor_id**	32	
if (/* image series cosignature */) {		
ECI_Client_Series_Id**client_series_id**	32	
format_version	8	uimsbf
if (format_version == 0x01){		
ECI_Signature_v1 **series_cosignature**		
}		
}		
if (/* image cosignature */){		
ECI_Client_id**client_id**	32	
ECI_Data_Signature **image_cosignature**		
}		
}		

语义：

type：字节	按照表5.2-2的值。
entity_id：整数	**平台操作证书**情形下分配给签名的唯一**标识符**。与仅分配给一个**允许的客户端图像**的cosignature_version字段一起使用。
version：整数	在**平台操作**改变其公钥的情况下增加（例如递增最高有效位）。该字段的较低有效位可用于表示客户端**图像系列**或客户端图像的（部分）版本，以方便**平台操作**使用**平台操作**客户端撤销列表中的版本字段来管理客户端版本的撤销。
cosignature_id：ECI_PO_Cosignature_Id	识别客户端图像上的共同签名标识符。该字段包含在共同签名计算中。
client_tag：整数	用于安装目的的短格式标识符，用于在**平台操作**情形下指定一个client_type。只有从**用户**角度可以互相替换的客户端，应具有相同的client_tag值。通常，客户的次要版本是相同的。
vendor_id：ECI_Vendor_Id	**ECI客户端**图像的**供应商证书**Id。该字段可用于查找客户端**图像系列**或客户端图像，为此在该数据结构中提供共同签名。
client_series_id：ECI_Client_series_id	客户端系列**证书**的ID用于验证图像。client_series_id字段的类型字段应与client_image_series的**平台操作证书**子类型相匹配：参见表5.2-2，从而定义数据结构的替代解释的正确选择。
format_version	适用于共同签名的**证书**定义格式的版本（参见表5.2-1）。这应匹配客户端**证书**版本定义。该字段唯一有效的值是0x01。
series_cosignature：ECI_Signature_v1	这是client_image_series证书的**平台操作**密钥的共同签名。输入到签名计算中的数据应定义为与client_image_series证书相同，用该数据结构的cosignature_id来替换client_image_series_id，并用承载**证书**初始client_image_series_id字段的4字节扩展来替换扩展字段。
client_id：ECI_Client_Id	客户端图像的ID。client_id字段的类型字段应与client_image的**平台操作证书**子类型相匹配：参见表5.2-2，从而定义数据结构的替代解释的正确选择。
image_cosignature：ECI_Data_Signature	这是客户端图像的**平台操作**密钥的共同签名。输入到签名计算中的数据应定义为：如第7.6.1节所定义，cosignature_id字段后跟输入到客户端图像签名计算中的客户端图像文件的数据。

7.6 文件格式

7.6.1 ECI客户端图像文件格式

ECI客户端证书包含验证**ECI客户端ECI TA**真实性所需的数据。它应使用在表7.6.1-1中定义的格式。

表7.6.1-1 – 客户端证书定义

语法	位数	助记符
ECI_Client_Credentials {		
ECI_Certificate_Chain **client_chain**		
if (client_chain.chain_length == 0x1) { /* no client series; regular image */		
ECI_RL**client_rl**		
}		
ECI_Data_Signature **client_signature**		
}		

语义：

header：ECI_Client_Chain_Header	**ECI客户端**链文件的标题。
client_chain：ECI_Client_Chain	用于验证**ECI客户端图像**的**证书链**，从安全供应商根撤消列表开始，以非基于图像系列的**ECI客户端**的安全供应商证书结束，或者以基于**ECI客户端**图像系列的**ECI客户端**系列**证书**结束。
client_rl：ECI_RL	**ECI客户端**图像Id的撤销列表。
client_signature: ECI_Data_Signature	签名验证**ECI客户端图像**，由**ECI客户端链**提供的公钥。

ECI客户端图像文件在表7.6.1-2中定义。

表7.6.1-2 – ECI客户端图像文件定义

语法	位数	助记符
ECI_Client_Image_File {		
magic = 'ECI'	24	uimsbf
image_header_version	8	uimsbf
ECI_Client_Credentials**credentials**		
if (image_header_version == 0x01) {		
if (credentials.client_chain.chain_length == 0x1) { /* regular image */		
ECI_Client_Id**client_id**	32	uimsbf
}		
if (credentials.client_chain..chain_length == 0x2) { /* Image Series image*/		
ECI_Image_Target_Id_Id**target_id**	64	uimsbf
ECI_Client_Series_Id**client_series_id**	32	
}		
vendor_id	20	uimsbf
image_encrypted_flag	14	uimsbf
online_flag	1	uimsbf
Reserved	10	
for (i=0; i<n; i++) {		
client_image_byte	8	uimsbf
}		
}		
}		

语义：

magic：字节[3]	幻数用于验证以下数据的格式。它具有字符“ECI”三个8位ASCII表示的值。**ECI主机**应检查该字段的值，以验证**ECI**文件是否具有用于其他数据完整性的预期格式。
image_header_version: 字节	图像标题的格式版本。值0x01是当前定义的版本；所有其他值都予保留。**ECI主机**应忽略任何版本号未被认可的图像。
credentials: ECI_Client_Credentials	**ECI客户端证书**，用于验证**ECI客户端图像**的真实性。
series_image: 布尔	系列图像不是一个字段，而是从证书计算得到的一个函数，指示存在一个**ECI客户端**类型系列**证书**。
series_id: ECI_Client_Series_Id	以下图像的图像系列的**ECI客户端**系列ID。**ECI主机**应在加载**ECI客户端图像**之前对该值进行检查。
series_image_id: ECI_Client-series_Image_Id	以下图像的图像系列中的图像ID。**ECI主机**应在加载**ECI客户端图像**之前对该值进行检查。
client_id: ECI_Client_Id	**ECI客户端**图像的**ECI客户端**ID。**ECI主机**应在加载**ECI客户端图像**之前对该值进行检查。
vendor_id: ECI_Vendor_Id	**ECI客户端图像**的安全供应商的供应商ID，如第7.4.2节ECI_Vendor_Id结构中的定义。**ECI主机**应在加载（新的）**ECI客户端图像**之前对该字段进行检查。
image_encrypted_flag: 整数	该标志指明图像是否加密。如果该字段的值为0b0，则图像未经加密。如果该字段的值为0b1，则图像经加密。
online_flag: 整数	该标志指明检索密钥以解密图像的协议是否需要使用随机数与供应服务器进行在线交互。参见第7.8.3节。
client_image_byte: 字节	包含客户端图像的字节序列。

在表7.6.1-2中，“**ECI主机**应检查”意味着**ECI主机**应验证它所期望的值将与现场的值相匹配。

ECI客户端图像签名应根据证书字段后文件中的所有数据来计算。

7.6.2 平台操作链数据

ECI客户端图像文件在表7.6.2-1中定义。

表7.6.2-1 – 平台操作链文件定义

语法	位数	助记符
ECI_Operation_Certificate_File {		
magic = 'EPC'	24	uimsbf
version	8	uimsbf
if (version == 0x01) {		
ECI_Certificate_Chain**operation_chain**		
ECI_RL po_client_rl		
client_image_count	16	uimsbf
for (i=0; i<client_image_count; i++) {		
ECI_PO_Client_Cosignature_Data **po_client_data**		
}		
ECI_RL po_client_rl		
}		
}		

语义：

magic：字节[3]	幻数用于验证以下数据的格式。它具有字符“EPC”三个8位ASCII表示的值。**ECI主机**应检查该字段的值，以验证ECI文件是否具有用于其他数据完整性的预期格式。
Image_header_version：字节	图像标题的格式版本。值0x01是当前定义的版本；所有其他值都予保留。**ECI主机**应忽略任何版本号未被认可的图像。
operation_chain：ECI_Client_Chain	用于验证**ECI客户端图像**的**证书链**，从运营商根撤销列表开始，以**平台操作证书**结束。
po_client_rl：ECI_RL	这是用于验证客户端图像共同签名的**平台操作**客户端撤销列表。**ECI主机**应检查po_client_data中的cosignature_ids，作为验证共同签名的一部分。
client_image_count：整数	以下循环中客户端图像的签名数据结构的数量。

表7.6.2-1“**ECI主机**应检查”意味着**ECI主机**应验证它所期望的值将与现场值相匹配。

7.6.3 撤销数据文件

代表**ECI客户端**加载程序，有两种类型的撤销数据文件。这两种文件都使用表5.5-2中定义的ECI_Revocation_Data_File格式。

ECI客户端撤销数据文件使用的father_type等于0x0（**根证书**），sub_type等于供应商撤销列表类型。revocation_data符合约束条件，即树中的叶子撤销列表是**ECI客户端**撤销列表。

平台操作撤销数据文件使用的father_type等于0x0（**根证书**），sub_type等于**运营商**撤销列表类型。revocation_data符合约束条件，即树中的叶子撤销列表是**平台操作**撤销列表。

7.7 ECI客户端资源传输协议

7.7.1 概述和配置

本节定义了**CPE**和**平台操作**中协议的应用。

广播协议不提供**图像系列**选项。基于系列的图像仅适用于IP连接的设备。

支持对**ECI客户端**资源进行广播和在线访问的**CPE**应使用优先级较高的广播访问（除非本建议书中另有说明），以便卸载在线流量，但可在紧急情况下（**用户**等待）使用在线访问，以及如果在广播网络上无法满足最低访问频率，则应使用在线访问。

7.7.2 广播传输协议

7.7.2.1 引言

ECI要求代表**ECI客户端**与/或**ECI主机**各种各样功能的支持数据能够初始化和支持**ECI客户端**。相同的传输协议用于所有类型的数据，并在本节中进行定义。它与用于下载**ECI主机图像**文件的协议密切相关。

对于广播传送，数据在桶中使用**CPE**所用访问索引上的散列函数进行分解，以确定它是否需要数据。通过使用桶，**CPE**需要下载的数据量显著减少，并且改善了与**CPE**实际相关的、监控数据中变化的选择性。

定义了以下单独的轮播组（按内容类型）：

- ECI客户端图像（按安全供应商）。

- **ECI客户端**撤销数据，基于<client_id，client-version_major>和vendor_id索引构建于桶中。
- 平台操作**证书链**。
- **平台操作**撤销数据，基于provider_id和operator_id索引构建于桶中。
- **ECI主机图像**撤销数据，以桶的形式构建。
- **ECI** AS_setup**ECI客户端**初始化数据，以桶的形式构建。
- 为入口和出口数据结构定义轮播组（参见第9.8节）。
- 为**运营商**专有数据定义轮播组。

所有DSMCC轮播参数都应符合[ETSI EN 301 192]的要求。

一个**运营商**可以在单独的多路复用上使用多个轮播来传送所有需要的数据。但是，对于任何特定的**ECI客户端**，**ECI主机**都只需监控数据轮播的单个位置DII的更新。

7.7.2.2 将证书和撤销数据移交给运营商

将证书和撤销列表传送给**运营商**的数据格式和协议不是**ECI**规范的组成部分。

7.7.2.3 安全供应商向运营商的移交

将内容从**安全供应商**转送给**运营商**的数据格式和协议不是本建议书的组成部分。

7.7.2.4 PSI信令

轮播应使用PMT中的stream_identifier_descriptor [ETSI EN 300 468]来标记用于传送轮播的流，以便允许通过SI中的data_broadcast描述符来引用。

轮播应使用data_broadcast_id_descriptor和data_broadcast_id，如表7.7.2.4-1中所定义。

表7.7.2.4-1 – ECI特定轮播的数据广播ID值

Data_broadcast_id值	含义
由DVB项目办公室分配，参见[ETSI TS 101 162]中定义的broadcast-id值。	**ECI运营商**特定的客户端支持数据轮播。

data_broadcast_id_descriptor的选择器字节应遵循表7.7.2.4-2中定义的结构。

表7.7.2.4-2 – ECI DVB DSMCC数据轮播的轮播ID结构

语法	位数	助记符
ECI_carousel_id_structure {		
version	8	uimsbf
if (version == 0x01){		
operator_id	20	uimsbf
platform_**operation_id**	20	uimsbf
}		
}		

语义：

version：整数	结构的版本；目前只定义了0x01。所有其他值都予保留。遇到0x01之外版本的**CPE**将忽略该描述符。
operator_id：ECI_Operator_Id	轮播**平台操作运营商**的ECI ID（定义用于任何**运营商证书**）。
platform_operation_id：ECI_Platform_Operation-Id	根据**平台操作证书：平台操作**的ID。

7.7.2.5 SI信令

7.7.2.5.1 通过数据位置链接描述符的数据轮播位置信令

ECI客户端数据位置链接描述符是一个**ECI**专有DVB链接描述符[ETSI TS 101 162]。该链接描述符通过多路复用位置为**CPE**提供协助，承载一个用于特定**平台操作**的**ECI客户端**数据轮播。该链接描述符承载于NIT或BAT中。**ECI客户端**数据位置链接描述符应通过DVB专有数据说明符描述符[ETSI TS 101 162]始终位于表格部分之前，其中private_data_specifier字段值等于[ETSI TS 101 162]中定义的“ECI”。该描述符可多次出现在NIT或BAT中。该链接描述符应承载于网络和具有4个以上多路复用的业务群中。

参照[ETSI EN 300 468]和[ETSI TS 101 211]中定义的链接描述符定义，**ECI客户端**数据位置链接描述符的字段具有以下特定应用程序：

- **service_id**：可设置为0x0000，以表示没有发送任何特定的service_id信号。
- **linkage_type**：值0x80，表示**ECI**客户端数据位置链接描述符。

ECI客户端数据位置链接描述符的私有数据字节字段应承载表7.7.2.5.1-1中定义的结构。

表7.7.2.5.1-1 – ECI客户端数据轮播位置链接描述符的私有数据结构

语法	位数	助记符
ECI_client_data_location {		
version	8	uimsbf
if (version==0x01){		
for (i=0;i<n; i++){		
operator_id	20	uimsbf
platform_**operation_id**	20	uimsbf
}		
}		
}		

语义：

version：整数	结构的版本；目前只定义了0x01。所有其他值都予保留。遇到0x01之外版本的**CPE**将忽略该描述符。
operator_id：ECI_Operator_Id	轮播**平台操作运营商**的ECI ID（定义用于任何**运营商证书**）。值0x00000表示任何**运营商**。
platform_operation_id：ECI_Platform_Operation-Id	根据**平台操作证书：平台操作**的ID。值0x0000表示任何**平台操作**。

网络和业务群运营商可以使用通配符说明符（值0x00000）作为operator_id或platform_operation_id，以链接至一个承载一个或多个**ECI客户端**数据轮播的多路复用上。出于效率的原因，建议这种信令限制于帮助**CPE**根据需要检查尽可能少的多路复用，以便定位某个特定的**平台操作**轮播。

建议在NIT或BAT中仅使用一个多路复用的单个**ECI客户端**数据轮播位置链接描述符，且位于该多路复用中的所有适用轮播都列于一个ECI_Client_data_location结构中。

7.7.2.5.2 ECI客户端紧急下载描述符

为了表明需要紧急替换**ECI客户端图像**，可以将一个或多个ECI_client_emergency_download描述符置于NIT、BAT中，或者置于服务的某个SDT条目中，对该服务，标记的**ECI客户端**可以提供访问。**ECI主机**应能够从任何表中检索该描述符，它出现于任何当前调谐的多路复用中，执行关联的处理并使用任何备用调谐器来访问相关的多路复用，以获取该描述符，最坏的情况是30分钟间隔期。

ECI_client_emergency_download_descriptor允许针对特定的操作平台和特定的主机类型，以便最大限度地减少因紧急更新而造成的干扰。

当**ECI主机**发现一个新的ECI_client_emergency_download描述符（由table-origin和emergency_id字段验证）时，它应将其主机和客户端配置与描述符中的目标信息进行匹配。如果发现一个目标匹配且当前安装的客户端图像的版本需要更新，则主机应根据emergency_indicator执行该更新。在出现资源冲突的情况下，这可导致**CPE**中正在进行之**用户**活动的中断。

ECI操作描述符是一个DVB专有描述符，并应通过使用**ECI**private_data_specifier_field的DVB private_data_specifier_descriptor，始终位于出现它的表格之前（参见[ETSI EN 300 468]）。描述符的语法在表7.7.2.5.2-1中定义。

表7.7.2.5.2-1 – ECI_Client_Emergency_Download_Descriptor

语法	位数	助记符
ECI_client_emergency_download_descriptor{		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
/* main loop */		
main_loop_nr	8	uimsbf
for (i=0; i<main_loop_nr; i++){		
/* target platform */		
platform_operation_tag	8	uimsbf
/* host target loop */		
host_nr	8	uimsbf
/* host id target loop */		
for (j=0; j<host_nr; j++){		
manufacturer_id	20	uimsbf
cpe_type_id	20	uimsbf
host_version	8	uimsbf
}		
/* client image loop */		
client_nr		
for (j=0; j<client_nr; j++){		
emergency_indicator	4	uimsbf
client_tag	4	uimsbf
min_client_version_major	8	uimsbf
min_client_version_minor	8	uimsbf
}		
}		
/* private data till end of descriptor*/		
for (i=0; i<n; i++){		
private_data_byte	8	
}		
}		

语义：

descriptor_tag	descriptor_tag的**ECI**专有标签值：参见[b-ITU-T J Suppl. 7]。
descriptor_length	参见[ETSI EN 300 468]。
main_loop_nr	主循环中的条目数。单独的主循环条目应由**ECI主机**单独进行评估，即具有OR语义。一个循环条目中的各种各样元素都应具有AND语义。
platform_operation_tag	**ECI**平台的标签值，如在NIT/BAT的ECI_platform_operation_descriptor中所列。如果platform_operation与其中一个已安装**ECI客户端**的platform_operation相匹配，则**ECI主机**应考虑紧急更新。
host_nr	主机目标循环中的条目数；值0表示所有**ECI主机**都是有针对性的。循环条目应具有OR语义；即：如果任何主机目标规范与主循环中的目标条件相匹配，则具有匹配状态。
manufacturer_id	由紧急更新定位的主机Manufacturer_id。如果该字段的值与**ECI主机**的manyufacturer_id相匹配，则主机应考虑紧急更新。
cpe_type_id	值如表6.2.2.1-2中的ECI_CPE_Type_ID所定义。如果主机的cpe_type_id与该字段的值相匹配，则**ECI主机**应考虑紧急更新。Cpe_type_id.cpe_type等于0x000表示任何**ECI主机**cpe_types是匹配的（且cpe_model和主机版本将被忽略）。cpe_type_id.cpe_model等于0x00表示任何**ECI主机**cpe_model是匹配的（且主机版本将被忽略）。
host_version	当且仅当其主机版本小于或等于该字段中的值时，**ECI主机**才会考虑紧急更新。参见注释。
client_nr	客户端图像循环中的条目数。循环条目应具有OR语义，并且所有匹配的客户端图像都应考虑用于紧急更新。
emergency_indicator	**ECI主机**应使用该字段的值来选择适当的行为，以启动下载和客户端的后续更新，如表7.7.2.5.2-2中所定义。

client_tag	在NIT/BAT的ECI_platform_operation_descriptor中列出的、标识**ECI客户端**的标记值，匹配相同主循环中的platform_operation_tag字段。如果引用的vendor_id和client_id与**ECI主机**中已安装的客户端之一相匹配，则**ECI主机**应考虑进行紧急更新。
min_client_version_major	该字段表示客户端图像最低可接受的主版本号。如果安装的客户端与client_tag相匹配，其主版本低于该字段的值，则**ECI主机**会考虑执行紧急更新。
min_client_version_minor	该字段表示客户端图像最低可接受的次版本号。如果安装的**ECI客户端**与client_tag相匹配，并且其次要版本小于该字段的值，主版本等于min_client_version_major，则**ECI主机**应考虑执行紧急更新。
client_id	**ECI客户端**的客户端标识符，它通过platform_operation_tag为服务提供解密服务，如表7.4.4-1中所定义。
private_data_byte	私有数据：内容可由管理该描述符广播的**运营商**来定义。
注： 字段值等于0xFF意味着所有主机版本匹配。	

表7.7.2.5.2-1定义了主循环中需要满足的许多条件（具有AND语义），以便**ECI主机**考虑执行紧急更新。如果符合所有这些条件，则**ECI主机**应根据该客户端的emergency_indicator字段来执行紧急下载并安装一个或多个客户端图像。

表7.7.2.5.2-2 – ECI_Client_emergency_download_descriptoremergency_indicator字段值

名称	值	描述
系统紧急	0×01	**ECI主机**应下载新的客户端图像并尽快安装，以便在需要时中断正在进行的**用户**激发活动（参见注1）。
客户端紧急	0×02	**ECI主机**应下载新的客户端图像并在该客户端的任何媒质句柄会话打开之前进行安装。该客户端任何正在进行的媒质句柄会话都应首先终止（参见注2）。
客户端紧迫	0x03	**ECI主机**应第一时间下载并安装新的主机图像，这不会对任何**用户**激发的活动造成任何干扰。**ECI主机**应在下一次启动事件期间下载最新的主机图像（参见注3）。
RFU	其他	保留以供未来使用。
注1 – 例如，如果当前的**ECI客户端**可能对**ECI主机**与/或其他**ECI客户端**造成损害，则**运营商**可以使用此项，并且必须立即予以更换。 注2 – 例如，如果当前的**ECI客户端**对解密服务的性能非常差，则**运营商**可以使用它。 注3 – 例如，如果当前的**ECI客户端**有严重的解密服务缺陷，但可以合理执行常规用例，则**运营商**可以使用它。		

7.7.2.6 轮播兼容性描述符

DVB DSMCC数据轮播[ETSI EN 301 192]中使用的compatibilityDescriptor应在DSI DII消息中使用。

compatibilityDescriptor提供有关在轮播组中传输的数据类型的信息。specifierData()应包含**ECI** OUI。表7.7.2.6-1定义了**ECI客户端**数据轮播中compatibilityDescriptor的适用字段。

表7.7.2.6-1 – ECI数据轮播内容类型

描述类型字段	组目的	模型字段	版本字段	桶索引以计算模块ID
0xA0	一个供应商的**ECI客户端图像**和证书文件	图像**安全供应商**的Vendor_id		自由分配
0xA2	**ECI客户端**撤销数据文件（作为桶）	platform_operation_id		= Vendor_id + <Client_type，client_version_major>（见注释）
0xA3	**平台操作**链文件	platform_operation_id，platform_operation_version		自由分配
0xA4	**平台操作**撤销数据文件（作为桶）	platform_operation_id		= Operator_id + provider_id
0xA5	**ECI主机**撤销数据文件（作为桶）	platform_operation_id		= Manufacturer_id + cpe_type_id
0xA6	AS_setup文件（如桶）	platform_operation_id		CPE的target_id
0xA7-0xAA	UI应用程序容器（参见第9.4.3.4.2节）	由运营商定义		自由分配
0xB0	出口树文件	（出口**ECI客户端**的）platform_operation_id		自由分配
0xB1	入口链文件	（入口**ECI客户端**的）platform_operation_id		自由分配
0xB2	入口身份验证链文件	（入口**ECI客户端**的）platform_operation_id		自由分配
0xB8-0xBF	运营商专有格式	由**运营商**定义		由**运营商**定义
其他值	保留			
注： 两个字段的串接，最重要的一个作为第一个参数，组成一个20位的数字。				

桶索引计算应使用32位模块整数算术，并在第7.7.2.7节中进行定义。

7.7.2.7 轮播DSI

如果轮播是双层轮播，则DSI应包含轮播中组的完整索引（即每个DII一个循环条目）。

compatibilityDescriptor在表7.7.2.6-1中定义。DII非循环字段应符合以下约束条件：

- 块大小：至少512字节，对于模块较大的组，推荐使用至少2k字节；
- tCDownloadScenario：至少是组中最慢的DDB重复消息的4倍。TCDownload还应符合表B.4-1中的最大限制条件；
- numberOfModules：反映常规轮播模块的数量和桶化数据的桶数（每个映射到一个模块）。对于**平台操作证书链**数据，其值应为1。

以下tCDownloadScenario的值反映**CPE**获取一个完整数据项目的超时期限。它至少是组中任何模块的最慢DDB重复时间的四倍。各条目的值在第B.4节中进行定义。

以下模块循环字段应满足以下约束条件：

- moduleId：位15到位8应与DSI中对应groupInfo结构中的groupId的LSB相同。位7到位0按照表7.7.2.7-1进行分配。
- moduleVersion：应用程序取决于轮播类型，应符合表7.7.2.7-1的要求。
- 所有**ECI**轮播的moduleInfoLength：0。

表7.7.2.7-1 – ECI轮播组参数

组类型	ModuleId位7..0	ModuleVersion	ModuleInfo
客户端图像	client_type	client_version	无
客户端撤销数据	bucket_number	每次更新都会递增	无
平台操作客户端链	由**运营商**指派	每次更新都会递增	无
平台操作撤销数据	bucket_number	每次更新都会递增	无
ECI主机撤销数据	bucket_number	每次更新都会递增	无
ECI AS_setup数据	bucket_number	每次更新都会递增	无

对于桶化的数字，桶号（等于module_id位[7..0]）应通过简单的模操作从索引计算得出：

bucket_number = bucket Index % numberOfModules

7.7.2.8 轮播DDB

无特定要求。

7.7.2.9 动态轮播行为

轮播版本编号和DSI、DII更新应符合[ETSI TR 101 202]的要求。这意味着模块的任何更新都将反映在模块的版本号、其DII中，并级联至DSI（如果有的话）。

CPE实施方案可以监控其目标模块中的变化，以便在正常操作期间追踪出现的任何动态的更新。

7.7.3 万维网传输协议

7.7.3.1 引言

ECI主机可以从**运营商**指定的服务器上检索各种所需的数据项。

接口应使用第9.4.4.6节中规定的直接HTTPS请求，并遵循RESTfull设计原则[b-Richardson]，将请求编码为URL扩展和查询参数的组合，并将响应编码为二进制文件。

HTTP服务器应以下列状态码之一进行响应：

- 200：OK（返回请求的文件）。
- 302 FOUND：重定向将请求延迟到另一个服务器；http请求在返回的URL上重复。
- 404：项目不在服务器上。
- 500 .. 599：服务器错误。

用于请求的URL规范使用“BachusNaur”风格规范。对应**ECI**数据结构中字段的符号名称应表示为其值的十六进制表示（字符串'0'..'9'，'A'..'F'），两倍于数字的数，作为字节，用于表示**ECI**内部二进制数据结构中的数。服务器应忽略任何它无法识别的其他查询参数。

7.7.3.2 ECI万维网API概述

运营商应支持一个在线服务器，该服务器根据以下URL语法和语义响应以下HTTP1.1 [IETF RFC 7231] GET请求：

URL ::= base-url '/' 'eci' major '_' minor '/' tail.

主要和**次要**应以十进制表示来反映协议版本的主要和次要编号，而不包含前导的零。目前的版本为1.0。表7.7.3.2-1中给出了报尾的定义。

表7.7.3.2-1 – 报尾的定义

```
tail                ::=host_version                |
host_images                                        |
host_image_version                                 |
host_image                                         |
po_check |
po_client_checkpo_certchain                        |
po_revocation                                      |
client_version                                     |
client_credential_version |
client_image                                       |
          client_revocation                        |
          as_request                               |
          tail_extension*.
```

tail_extension指示本建议书中定义的**ECI**万维网API的各种各样扩展选项。

7.7.3.3 万维网API ECI主机相关的请求

定义了以下与**ECI主机**相关的万维网API请求：

- host_version ::= 'host-version ' '?target-id=' target_id.这将返回由**target_id**标识的、**CPE**最新版本的**ECI主机图像**集。
- host_images ::= 'hi-images' ' ?target-id=' target_id.这将返回由**target_id**标识的、**CPE**最新数量的**ECI主机**图像。
- host_image_version ::= 'hi-version' ' ?target-id=' target_id '&image-id=' image_id .这将返回由**target_id**标识的、**CPE**最新版本的**ECI主机图像**文件**image_id**。
- host_image ::= 'host-image' ' ?target-id=' target_id '&image-id=' image_id. 这将返回由**target_id**标识的、**CPE**最新**ECI主机图像image_number**。image_number ==“FF”将返回**ECI主机图像**的**ECI主机**证书文件，包括最新的撤销数据。

对于与**ECI主机**相关的请求，**平台操作**的服务器可以支持**ECI主机**以获取其所需的任何**CPE**类型。如果它支持**CPE**类型，则它应支持最新的**ECI主机图像**全集以及相应的**host_image_version**、**host_images**和**host_revocation**查询。返回的文件格式如表7.7.3.3-1中定义的ECI_Host_Version_File。

表7.7.3.3-1 – ECI主机版本文件定义

语法	位数	助记符
ECI_Host_Version_File {		
magic = 'RHVE'	32	uimsbf
host_version	8	uimsbf
}		

语义：

magic：字节[4]	字符串“RHIM”的8位ASCII表示。
host_version：整数	**ECI主机证书**的版本号。

返回文件的格式如表7.7.3.3-2中定义的ECI_Host_Images_File。

表7.7.3.3-2 – 主机图像文件定义

语法	位数	助记符
ECI_Host_Images_File {		
magic = 'RHIM'	32	uimsbf
host_images	8	uimsbf
}		

语义：

magic：字节[4]	字符串“RHIM”的8位ASCII表示。
host_images：整数	请求中标识的、**CPE**类型支持的**ECI主机图像**数量。

返回文件的格式如表7.7.3.3-3中定义的ECI_Host_Image_Version_File。

表7.7.3.3-3 – 主机图像版本文件语法

语法	位数	助记符
ECI_Host_Image_Version_File {		
magic = 'RHIV'	32	uimsbf
host_image_version	16	uimsbf
}		

语义：

magic：字节[4]	字符串“RHIV”的8位ASCII表示。
host_image_version：整数	由请求标识的、**ECI主机图像**的**ECI主机图像**版本。

7.7.3.4 万维网API平台操作相关的请求

平台操作的服务器应代表其支持的**平台操作**ID支持以下请求：

```
po_check ::= 'po_check' '/' operator_id '/'
platform_operation_id .
```

这将以表7.7.3.4-1中定义的文件格式返回为**operator_id**、**platform_operation_id**颁发的**证书**撤销状态。**平台操作**的服务器至少应通过该接口在操作中支持其自身的**平台操作证书**。

```
po_client_check ::= 'po-client-check' '/' operator_id '/'
platform_operation_id '?cosignature-id=' cosignature_id .
```

这将根据最新的平台操作客户端撤销列表，为cosignature_id返回**ECI客户端图像**的平台撤销状态。参见表7.7.3.4-2。

```
po_certchain ::= 'po-chain' '/' operator_id '/'
platform_operation_id .
```

这将为表7.6.2-1中定义的**operator_id**、**platform_operation_id**标识的平台操作，返回最新的**ECI客户端**链。**平台操作**的服务器至少应通过该接口在操作中支持其自身的**平台操作证书**。

```
po_revocation_::= 'po-revoc' '/'
operator_id .
```

这将返回最新的**平台操作**撤消数据文件，其中包含由**operator_id**标识的**运营商**的撤销列表。服务器应至少支持其自身**平台操作**的**运营商**的最新撤销数据。**ECI主机**应使用该API来尝试获取所有存储的**ECI客户端**的最新撤销数据。

表7.7.3.4-1 – 平台操作校验文件语法

语法	位数	助记符
ECI_PO_Check_File {		
magic = 'RPCH'	32	uimsbf
non_revoked_certificate_flag	8	uimsbf
}		

语义：

magic：字节[4]	字符串“RHIV”的8位ASCII表示。
non_revoked_certificate_flag：字节	如果请求标识的**平台操作**ID的**证书**已被撤销，则值为0x00，否则为0x01。

表7.7.3.4-2 – 平台操作客户端检查文件语法

语法	位数	助记符
ECI_PO_Client_Check_File {		
magic = 'RPCC'	32	uimsbf
non_revoked_certificate_flag	8	uimsbf
}		

语义：

magic：字节[4]	字符串“RHIV”的8位ASCII表示。
non_revoked_certificate_flag：字节	如果根据平台操作的最新平台操作客户端撤消列表撤消与请求的cosignature_id字段相关联的客户端图像，则值为0x00值，否则为0x01。

7.7.3.5 万维网API客户端请求

运营商服务器应代表其**平台操作**id所需的客户端支持以下请求：

```
client_version   ::=   'client-ver'   '/'   vendor_id   '/'
                       client_type '/' client_version_major .
```

- 这将返回一个客户端版本文件（参见表7.7.3.5-1），其中包含由**vendor_id**、**client_type**标识的客户端的最新**ECI客户端**图像版本。服务器应至少支持用于操作其自身**平台操作**服务的客户端。

```
client_credential_version    ::=    'client-ver'    '/'    vendor_id    '/'
client_type '/' client_version_major .
```

- 这将返回一个客户端证书版本文件（参见表7.7.3.5-2），其中包含由**vendor_id**、**client_type**标识的客户端的最新**ECI客户端**证书版本。服务器应至少支持用于操作其自身**平台操作**服务的客户端。

```
client_image   ::=   'client-img'   '/'   vendor_id   '/'
client_type                        '/'client_version_major
                  ['? &target-id=' image_target_id] .
```

- 这将返回由<vendor_id，client_type，client_version_major>标识的客户端的最新**ECI客户端图像**文件。在类型image_target_id的**图像**情况下，提供ECI_Image_Target_Id作为查询参数。服务器应至少支持用于操作其自身**平台操作**服务的**ECI客户端**的**供应商**。**ECI主机**应使用该API来尝试获取所有存储的**ECI客户端**的最新撤销数据。

```
client_revocation_data ::= 'client-revoc' '/' vendor_id .
```

- 这将返回由**vendor_id**标识的客户端的最新**ECI客户端**撤销数据文件。服务器应至少支持用于操作自身**平台操作**服务的客户端。

表7.7.3.5-1 – 客户端版本文件语法

语法	位数	助记符
ECI_Client_Version_File {		
magic = 'RCVE'	32	uimsbf
client_version	16	uimsbf
emergency_download_descriptor		
}		

语义：

magic：字节[4]	字符串“RCVE”的8位ASCII表示。
client_version：整数	请求中标识的客户端类型的最新客户端版本。
emergency_download_descriptor	一个 **ECI_client_emergency_download_descriptor**，当中 **ECI 主机**应假设platform_operation_tag应与客户端万维网API提供商的平台操作相匹配，并且client_tag应匹配万维网API参数中请求的客户端图像。

表7.7.3.5-2 – 客户端证书版本文件语法

语法	位数	助记符
ECI_Client_Credential_Version_File {		
magic = 'RCCV'	32	uimsbf
root_version	8	uimsbf
vendor_rl_version	24	uimsbf
eci_vendor_id	32	uimsbf
padding(4)		
client_rl_version	24	uimsbf
eci_client_id	32	uimsbf
}		

语义：

magic：字节[4]	字符串“RCCV”的8位ASCII表示。
root_version：整数	最新**ECI客户端**证书的根版本（如表5.3-1中所定义）。
vendor_rl_version：整数	最新**ECI客户端**证书的安全供应商撤销列表版本号。
eci_vendor_id：ECI_Vendor_Id	最新**ECI客户端**证书的ECI_Vendor_Id（如表7.6.1-2中所定义）。
client_rl_version：整数	最新**ECI客户端**证书的客户端撤销列表版本号。
eci_client_id：ECI_Client_Series-Id	最新**ECI客户端**证书的ECI_Client_Series_Id（如表7.6.12所定义）。

7.7.3.6 万维网API AS_setup请求

如果**运营商**支持加密模式**ECI客户端**的在线注册，则应支持以下请求：

```
as_request ::= 'as_request' '/' vendor_id '/' eci_client_id
               '?&image-target-id=' target_id '&nonce=' nonce].
```

该请求返回指定客户端的<as_setup文件（<vendor_id，eci_client_id>）以及由ECI_Image_Target_Idtarget_id指定的**CPE**。eci_client_id的类型可以是ECI_Client_Id或EI_Client_Series_Id。随机数是由**ECI客户端图像**解密协议指定的随机数值。更多的详细信息，请参阅第7.8.4.2节。

7.8 平台操作ECI客户端安装

7.8.1 范围和配置

平台操作可以选择**ECI客户端**安装的安全选项，并使用image_encrypted_flag和**ECI客户端图像**文件中的在线标志来发出信号（参见表7.6.1-2）：

- “具有未加密**ECI客户端图像**文件的**ECI客户端**安装模式”，其中在第7.2节中定义由信令建议的（最新版本的）**ECI客户端**，以便下载解码业务，并启动**ECI客户端**。
- “具有加密**ECI客户端图像**文件的**ECI客户端**安装模式”，除第一种模式外，允许**平台操作**加密**ECI客户端图像**并根据[ITU-T J.1014]中的定义进行验证。**ECI客户端**解密是**ECI主机**特定的，并包含**ECI主机**版本验证，从而通过不允许在未知或受损的**ECI主机**上进行解密而进一步确保解密后的**ECI客户端**的机密性。如果**CPE**未连接到在线

网络，则需要一个ECI_Image_Target_Id。在该用例中，ECI_Image_Target_Id需要手动地发送到安全头端。

ECI客户端启动的两个版本的协议在本条款的其余部分中定义。

使用加密安装模式操作在线**CPE**的**平台操作**，可以通过在**ECI客户端**的**平台操作**服务器的解密协议中使用AS生成的随机数来强制使用最新的**ECI客户端**（参见第7.7.3.6节）。

配置规则：

- 如果在线注册由**平台操作**提供（信令在第7.2节中定义），并且**CPE**能够访问在线服务，则**CPE**应使用在线注册协议。
- 支持广播接收的**CPE**应能执行广播注册协议。广播模式要求在初始**平台操作**注册时注册**CPE**。
- 支持广播网络、支持**CPE**而不同时在线连接的在线注册应支持广播模式注册。用户输入**CPE**注册信息的细节应遵守适用的格式规则。

7.8.2 带未加密ECI客户端图像文件的ECI客户端安装模式

在**ECI客户端**初始化之初，**ECI主机**为**平台操**作保留一个**AS时隙**，重置**AS时隙**并将**平台操作**公钥加载到[ITU-T J.1014]中定义的AS时隙中。

如果需要，**ECI主机**下载**ECI客户端**，将其存储在NV RAM中以供未来检索并启动。**ECI客户端**将通过安装进一步指导**用户**。在**CPE**没有用于广播系统安全注册的在线连接的情况下，安装可能涉及**用户**手动地将**CPE**ECI_Image_Target_Id值target_id发送给头端。

在任何后续的重新启动时，**ECI主机**将重新初始化**ECI客户端**。

7.8.3 带加密ECI客户端图像文件的ECI客户端安装模式

该操作模式使用**运营商**选择的密钥对**ECI客户端图像**进行加密下载。在**as_setup**结构中加密和承载该**运营商**选择的密钥。

在**ECI客户端**启动之初，**ECI主机**为平台操作保留一个**AS时隙**，重置AS时隙并将**平台操作**公钥加载到**AS时隙**中：

- **ECI主机**应区分用于**as_setup**检索的两种模式：**注册模式**：如果**ECI客户端**首次启动，或者POPK或**ECI客户端**版本已更改，或者客户端工作于在线重新注册模式，对每次重新注册都使用一个唯一的随机数。应从**平台操作**网络检索**CPE**的**as_setup**结构。
- **注册模式**：先前的**as_setup**结构从NV存储器检索而来。如果有任何未决的**ECI客户端**或**ECI主机**版本更改，则**ECI客户端**应警告**用户**启动或取消阻止此类下载（在默认下载设置下，这些设置通常应在合理的时间范围内自动进行）。下载一个新的**ECI客户端**还需要一个新的**as_setup**结构。

在注册模式下，**ECI主机**应执行以下操作来检索新的**as_setup**结构：

1) **ECI主机**初始化AS时隙并检索：
   - **CPE**的ECI_Image_Target_Id值target_id；

– 在在线注册的情况下，通过应用**getAsSlotRk**函数（参见[ITU-T J.1014]），从**AS时隙**检索到的一个随机数（128位）。

2) **ECI主机**应发送上述信息，以检索**平台操作**中的**as_setup**消息：

– 在**广播注册**的情况下，**ECI主机**应在带**平台操作**注册对话框的屏幕上显示**target_id**。**ECI主机**应从AS设置轮播中检索as_setup结构（参见第7.7.2节）。

注1 – 如果平台提供多种类型**ECI客户端**，**平台操作**可要求**用户**也提供一些额外的信息，以便为相应的**ECI客户端**类型提供**as_setup**。

注2 – 平台操作可以假定CPE已下载最新的ECI客户端图像版本，并仅为该ECI客户端图像提供as_setup结构。

– 在**在线注册**的情况下，**CPE**应使用第7.3.3节中的万维网API注册客户端标识、**CPE**的**target_id**和随机数。

注3 – **平台操作**可以决定运用随机数来确保在每个**ECI主机**重新初始化事件中的重新注册。

在注册模式下**as_setup**获取序列后，或者在注册模式下从NV存储器恢复**as_setup**结构后，**ECI主机**应初始化AS并尝试加载加密的**ECI客户端**：

1) 使用**reqAsClientImageDecrKey**消息在AS中加载**as_setup**结构。将**ECI客户端**证书客户端链加载到AS中。加载**平台操作**客户端撤销列表和**平台操作**客户端共同签名。以下故障情况至少应以可理解的方式向**用户**报告或做自动处理：

a) 旧的**ECI主机**版本 - **ECI主机**或其证书需要更新。

b) 旧的**ECI客户端**版本 - **ECI客户端**或其证书需要更新。

2) 使用AS计算得到的客户端**图像**密钥解密图像（如果需要的话），并使用**ECI客户端**签名和**平台操作**共同签名验证**ECI客户端图像**。

3) 验证错误时失败。

as_setup结构和as_setup_file格式应符合表7.8.3-1中的定义。

表7.8.3-1 – AS-Setup结构、文件和桶文件

语法	位数	助记符
ECI_As_Setup {		
as_version	8	uimsbf
if (as_setup_version == 0x01) {		
vendor_id	20	uimsbf
if (/* client image regular */){		
ECI_Client_id**client_id**		
}		
if (/* client image series */){		
ECI_Client_Series_Id**series_id**		
}		
ECI_Image_Target_Id**target_id**		
as_tag	16	uimsbf
online	1	uimsbf
padding(4)		
EciRootState**min_root_state**	32	
InputVinputV		

表7.8.3-1 – AS-Setup结构、文件和桶文件

语法	位数	助记符
symKeyeKey		
Extension **extension**		
}		
}		
ECI_As_Setup_File {		
magic file = 'AES'	24	uimsbf
as_setup_file_version	8	uimsbf
if (as_setup_version == 0x01){		
ECI_As_Setup**as_setup**		
}		
}		
ECI_As_Setup_Bucket_File {		
magic_bucket_file = 'AEB'	24	uimsbf
as_setup_bucket_version	8	uimsbf
if (as_setup_version == 0x01){ for (i=0; i<n; i++) {		
ECI_As_Setup**as_setup_item**		
}		
}		
}		

语义：

vendor_id：整数	该as_setup计划针对的**ECI客户端**的安全供应商。
client_id：ECI_Client_Id	该as_setup计划针对的**ECI客户端**的ID。前面的if语句使用type-fieldclient_id：它应对应于“常规客户端图像”。
series_id：ECI_Client_Series_Id	该**as_setup**计划针对的**ECI客户端系列**的ID。前面的if语句使用type-field client_id：它应对应于“客户端图像系列”。
target_id：ECI_Image_Target_id	ECI_Image_Target_Id标识该消息计划针对的**CPE**。
as_tag：整数	用于指示上述目标as_setup结构的版本的标记。该值应随该目标**as_setup**结构的任何更改而改变；如增量。
online：布尔	如果为真，则该消息要求在AK机制中使用时隙随机数；如果为假，则不需要随机数。注：只有在工作在线连接的情况下才能设置此位。
min_root_state：minEciRootState	最小根状态（最小根版本号、最小根撤消列表号），用于验证已加载的**ECI主机**和**ECI客户端**。该字段按照[ITU-T J.1014]中定义的字节序列进行编码。
inputV：InputV	用于AS系统的InputV消息。该字段按照[ITU-T J.1014]中定义的字节序列进行编码。
eKey：SymKey	用于解密图像的加密对称密钥。该字段按照[ITU-T J.1014]中定义的字节序列进行编码。
Extension：扩展	扩展数据，后向兼容。广播应用不应超过256字节，以便保持广播轮播紧凑。没有为该数据定义任何应用程序。
magic_file：字节[3]	字符串“AES”的8位ASCII表示。
as_setup_file_version：整数	ECI_AS_Setup_File格式的版本。值0和0x2..0xff保留。值0x01用于此处定义的格式。
as_setup：ECI_As_Setup	**平台操作**的**as_setup**结构，用于在特定的**ECI主机**上加载特定的加密**ECI客户端**。
magic_bucket_file：字节[3]	字符串“AEB”的8位ASCII表示。
as_setup_item：ECI_As_Setup	该桶中的as_setup结构。任何新的as_setup结构都应添加在桶顶；因此排序是在桶底最早的as_setup结构。必要时，只能从桶底删除as_setup结构。这样可以更快速地检查**CPE**的更新。即在第一次检查后，只需自上而下对as_setup结构进行检查，直到遇到先前系列检查中的第一个。

更新**as_setup**结构的最低检查频率应与第7.3.1节为其他**ECI客户端**定义的频率相同。请注意，更新通常意味着**CPE**的**ECI客户端**与/或**ECI主机**软件的更新；因此，对它们的任何更新也都应予下载，以确保可以完成一致的**ECI客户端**初始化序列。如果这种连贯的新集合不可用，则可以使用先前的连贯集合。

当**ECI主机**处于试图完成新的或更新的**ECI客户端**的广播模式（手动）注册状态时，**ECI主机**应以尽可能高的频率对**as_setup**文件轮播更新进行检查。

7.8.4 传输协议

7.8.4.1 广播协议

as_setup结构的广播协议应符合第7.7.2节的规定。

需要在**ECI客户端**版本更改上更新的as_setup结构数量可能非常大。为了在大型的、只有广播的操作中限制**ECI客户端**版本更改中新的在线as_setup消息数量，**平台操作**可以使新的**ECI客户端**可用，并展示新的证书投入使用，从而替换**CPE**上的**ECI客户端**组；并可多次重复此操作，以便在使用安全系统强制使用新的**ECI客户端**之前，捕获尽可能多的**CPE**。

7.8.4.2 在线协议

在线协议依赖于在第7.7.3节中定义的、**CPE**与**ECI客户端**之间直接的请求-响应协议，将**CPE target_id**和**nonce**作为请求的一部分予以传送，返回的是**ECI_As_Setup_File**。

7.8.5 目标标识呈现给用户

如果需要的话，**ECI主机**和**ECI客户端**都必须能够在广播网络上向**用户**呈现**CPE**的target_id，以防在没有在线连接的情况下允许生成解密**ECI客户端图像**所需的CPE特定信息，并允许生成**ECI客户端**的AS系统InitV消息（这些消息的传输协议由**ECI客户端**来定义）。此外，target_id可以作为**CPE**外部或随附文档中的印刷品来读取。本条款定义了至**用户**的target_id表示。

target_id是一个64位整数，应按照第6.2.2节中的规定提供给**用户**，使用9位校验以及添加9位子串而不是5位子串。因此，target_id被表示为6个4位数字的序列，其数字在0与7之间。

允许**CPE**和**ECI客户端**在其**用户**接口中使用自定义的表示（例如，基于专用**CPE**编号方案），但应总是基于上述表示格式来提供**ECI客户端**注册功能。

8 撤销

8.1 引言

所有参与方及其对**ECI生态系统**贡献的项目都将通过**ECI TA**进行认证。通过该认证，将有可能提供适合实施功能性和稳健性的基础质量，以及由贡献方采取的适当的更新措施。该认证过程还可以防止使用**ECI**生态系统的黑客和盗版操作。

ECI提供的功能可以根据加载的**CPE**硬件、**ECI主机**、其他**平台操作**和**ECI客户端**的**ECI TA**状态，选择性地排除向**CPE**交付的服务。

如果这些操作不遵守共同商定的、关于在共享**CPE**上不对其他**平台操作**造成干扰或者不通过**ECI**提供盗版服务的规则，则**ECI TA**可以撤销平台操作。同样，如果**ECI客户端**不遵守共同商定的、关于在共享**CPE**上不对其他**ECI客户端**造成干扰或者不实施黑客行为的规则，则**ECI TA**可以撤销**ECI客户端**。如果这些版本存在明显的不足之处而暴露**ECI客户端**秘密或允许操纵，则**ECI TA**可以进一步撤销**ECI主机**软件版本。

在所有上述情况下，负责撤销项目的组织可以修复缺陷，通常用新项目替换被撤销的项目。**安全供应商**可以用一个新的版本来替换**ECI客户端**，**CPE制造商**可以为**ECI主机**提供安全补丁，**运营商**可以通过其新版本的**平台操作证书**改进其操作。所有这些操作都具有协同性，建议在受影响各方与**ECI TA**之间达成合约协议后进行。

如果参与**ECI**的各方导致系统性地违反**ECI TA**协议，对其他方或**用户**造成不利影响，则可能会从**ECI TA**中撤销所有其出了力的项目。

如果某些**CPE**不再拥有有效的**ECI主机**，并且预计不会从其**CPE制造商**处收到更新，则可能会被撤销。如果**CPE**的引导加载程序受到攻击并允许加载不兼容的**ECI主机**软件，则也会发生这种情况。

如果可用，**CPE**应尝试用更新版本自动地替换被撤销的版本。不过，新的下载和**撤销列表**可以被阻止。在这种情况下，**平台操作**可以拒绝向这种**CPE**上的本地存储内容提供服务或拒绝提供补偿。

8.2 CPE撤销

ECI允许**平台操作**通过使用CA或DRM系统的选择性权利交付功能来排除向特定**CPE**提供的服务。**平台操作**可以从**ECI TA**检查**CPE**的最新**ECI TA**状态。如果**ECI TA**认为有必要撤销**CPE**，那么**平台操作**可以通过CA或DRM系统交付服务，根据其注册的芯片组ID，停止向**CPE**提供服务。

本建议书还有助于**平台操作**排除向运行已撤销**ECI主机**的CPE提供服务。根据第8.3节中定义的最新**ECI主机**撤销列表，**平台操作**可以使用高级安全系统来要求**ECI主机**的最低版本号。

如果认为合适，**ECI主机**撤销机制还可以用于**CPE**撤销，方法是指定最低**ECI主机**版本，它应高于目前已发布的版本。

8.3 通用撤销过程

本条款将最小**根**版本和最小**根**撤销列表版本组合为“最小**撤销列表**版本”。

ECI主机的最终撤销执行机制是服务不足：如果在**ECI主机**上存在撤销项目（尽管应用了（可假定为旧的）**撤销列表**），**平台操作**可能会决定停止向该**ECI主机**提供服务。**平台操作**所需的最低可接受**撤销列表**的交付受**AS系统**保护：其操作本身会导致服务不足。**平台操作**因此可以强制检查用于安装**ECI主机**和所有其他**平台操作**与**ECI客户端**的**证书**的版本。

平台操作应为上述任何项目提供**撤销列表**的下载服务。（**ECI主机**、**ECI客户端**和**平台操作**）。这可确保在**ECI主机**上加载的、所有**ECI客户端**和平台操作的最新撤销列表是可用的。

AS系统初始化[ITU-T J.1014]允许**ECI主机**为所有项目指定此最小期望**撤销列表**版本。它用于回顾性地验证**ECI主机**使用的撤销列表版本。**ECI主机**应使用其希望加载的、**ECI客户端**项目的最小**根撤销列表**值以及其已经加载的**ECI主机图像**。

注 – 建议**ECI主机**不加载会导致撤销生效的项目，而是通知**用户**。

防止过度服务不足需要最新的证书（必要时还有最新版本），以便所有要加载的项目在**ECI主机**中可用。为了防止因存在已撤销的**ECI主机平台操作证书**或**ECI客户端**所导致的安全隐患而造成**ECI客户端**无法正常运行，**ECI主机**应提供以下功能，以确保最新的证书和（如有必要）项目可用来防止出现过度的服务匮乏：

- 它应使用**CPE制造商**及其**ECI客户端**的**平台操作**的证书和**撤销列表**下载服务，保留在其当前**ECI主机**、**平台操作**和**ECI客户端**配置中验证的、每个项目的最新**ECI TA撤销列表**链。
- 所有相关**CPE**模式的默认设置应能够进行这种下载。

  除了未连接电源或禁止下载网络访问（不是由于**CPE**状态或操作模式），**CPE**不应有任何永久禁止下载的操作模式。
- 通过简单的**用户**操作，它应有可能恢复有关下载的默认设置以及和有关**ECI客户端**与**平台操作**的默认撤销。

本建议书允许**用户**覆盖默认的主机行为，以撤销会引起其他服务不足的项目。在**用户**这样做的情况下（例如，保持旧客户端运行），它们可能会遇到越来越难以呈现现代服务。

8.4 基于ECI主机撤销的撤销列表

没有正确维护的**CPE**可能有一个已被撤销的**ECI主机**。**CPE制造商**将提供更新的证书，包括最新的、适用的**ECI撤销列表**。另外，希望在**ECI主机**上运行**ECI客户端**的**平台操作**能够为**ECI主机证书**相关的**撤销列表**提供下载服务，并可以为选定的**ECI主机**提供下载服务。**ECI主机**应根据[ITU-T J.1014]中定义的通用**撤销列表**处理规则，为**ECI主机**证书（**根证书**和**制造商证书**）应用**撤销列表**。

ECI主机撤销数据文件的格式在第5.3节中定义。

8.5 ECI平台操作撤销

希望在**ECI主机**上运行**ECI客户端**的**平台操作**可以为与其他**平台操作**有关的**撤销列表**提供下载服务。**ECI主机**应根据[ITU-T J.1014]中定义的通用**撤销列表**处理规则，将**撤销列表**应用于所有安装的**平台操作**证书。

ECI平台操作撤销文件的格式在第7.6.3节中定义。

8.6 ECI客户端撤销

希望在**ECI主机**上运行**ECI客户端**的**平台操作**可以为与其他**ECI客户端**有关的**撤销列表**提供下载服务。**ECI主机**应根据[ITU-T J.1014]中定义的通用**撤销列表**处理规则，将**撤销列表**应用于所有已安装的**ECI客户端**证书。

ECI客户端撤销文件的格式在第7.6.3节中定义。

9 ECI客户端接口

9.1 引言

9.1.1 ECI客户端接口的体系结构

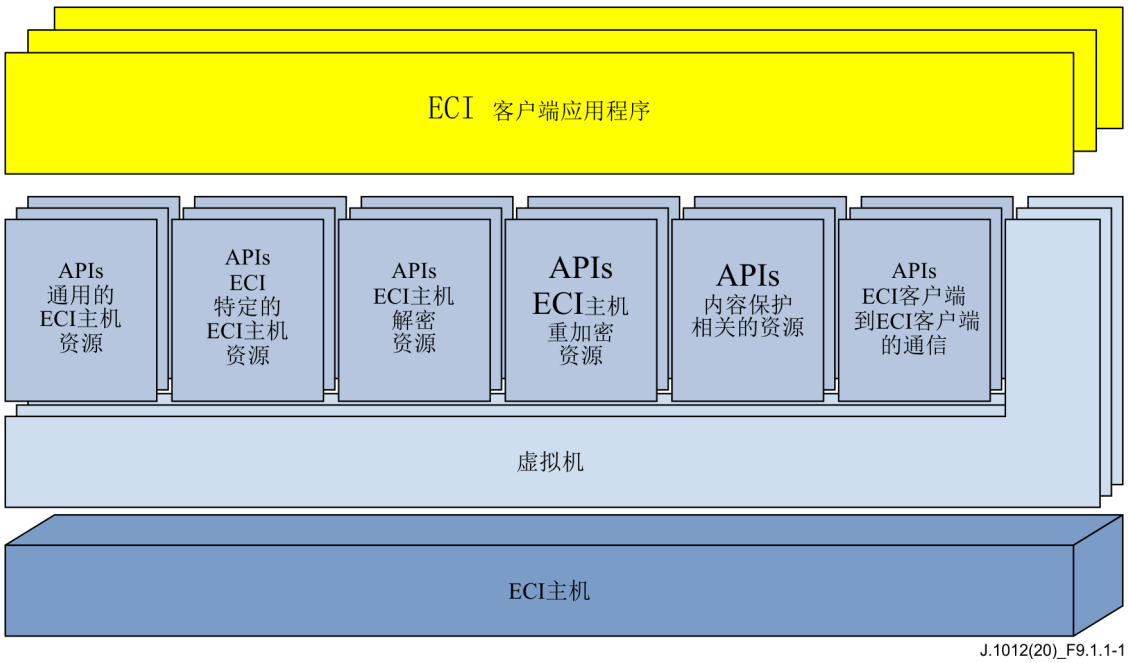


图9.1.1-1 – 第9节中定义的API结构

图9.1.11概述了**ECI**系统API的结构。它显示了**ECI客户端**可以使用的6块API。这些API块在第9.4节到第9.9节中进行规定。表9.1.1-1列出了本建议书第9节中定义的API；也可参考[b-ETSI GS ECI 002]。

表9.1.1-1 – 本文件中定义的API列表

条款号	API类别	描述
9.4	有关通用的**ECI主机**资源的API	支持**ECI客户端**通用功能的API
9.5	有关**ECI**特定的**ECI主机**资源的API	支持**ECI客户端ECI**特定功能的API
9.6	用于访问**ECI主机**解密资源的API	允许**ECI客户端**利用**ECI主机**解密资源的API
9.7	用于访问**ECI主机**重加密资源的API	允许**ECI客户端**利用**ECI主机**重加密资源的API
9.8	有关内容属性相关资源的API	支持**ECI客户端**内容保护功能的API
9.9	有关**ECI客户端**到**ECI客户端**通信的API	支持**ECI客户端**与**ECI客户端**之间直接通信的API

9.1.2 媒质句柄

媒质句柄是主机环境中的对象标识符，它提供了提供给**ECI客户端**的所有**ECI主机**接口的情形，用于控制内容项目的解密过程。**媒质句柄**还允许**ECI客户端**规定它想从内容容器获取的数据，以便能够对内容进行解扰。在广播网络传送的情况下，它还提供对要解码之节目选择以及来自传送网络之流选择（调谐功能）的控制。**ECI客户端**还可以请求**媒质句柄**访问调谐器，以便访问为从网络流操作**ECI客户端**所需的数据，应用程序/主机不会出于内容获取目的访问这些网络流。对于基于文件和OTT流的传送，**媒质句柄**为**ECI客户端**提供了一种手段，以访问未在标准位置中指定的文件/流中的安全数据。

媒质会话解扰直接在**ECI客户端**的控制下运行。CW应用与TS的同步基于TS中的加扰控制信息。CW（在此情形下通常称为密钥）与ISOBMFF CENC文件[ISO/IEC 23001-7]的同步应基于CENC KeyID标识符。

表9.1.2-1中列出了使用**媒质句柄**的会话。

表9.1.2-1 – 媒质句柄类型

名称	值	描述
MhDvbTs	0x01	TS应符合[ISO/IEC 13818-1-1]的要求。
MhIsobmffCenc	0x10	ISO BMFF文件应符合[ISO/IEC 23001-9]和[ISO/IEC 14496-12]的要求。
RFU	其他	保留以供未来使用。

9.2 ECI虚拟机接口

9.2.1 原则

应为每个**ECI客户端**创建一个单独的虚拟机实例。在第7节中定义了如何将**ECI客户端**的数据和指令加载到一个虚拟机（VM）中。

虚拟机的操作在[ITU-T J.1013]中进行定义；也可参考[b-ETSI GS ECI 001-4]。

ECI客户端与外部世界的所有交互都应使用第9.2.3节中定义的消息接口进行。

9.2.2 指令和数据（静态资源）

虚拟机将执行由**ECI客户端**加载程序提供给它的指令，作为**ECI客户端图像**代码段的一部分。

这些指令是非自我修改的，它由VM来保证。任何易于导致非期望与/或容易操控**ECI客户端**（例如口译员）行为的代码，都被认为是不适当的，且必须确保作为**ECI客户端**认证过程的一部分。

ECI客户端所需的最大代码和静态数据空间在[b-ITU-T J Suppl. 7]中提出。

9.2.3 与ECI主机的交互

ECI客户端与**ECI主机**的所有交互是根据本节中的消息模型来定义的。**ECI客户端**与**ECI主机**之间没有任何共享的数据，除了：

- 消息中包含的数据；
- 代表**ECI客户端**存储在**ECI主机**NV存储器中的任何数据；或者
- 至或自其他**ECI客户端**之通信信道中的消息中的任何数据。

请注意，该数据也通过消息进行交换。

消息模型基于从**ECI客户端**到**ECI主机**的三种不同类型的交换：

1) **同步客户端**发起的交换：**ECI客户端**调用**ECI主机**功能，它在很短的时间内做出反应。当**ECI主机**处理消息并提供返回消息时，**ECI客户端**的线程（执行流程）被阻止。

2) **异步客户端**发起的交换：**ECI客户端**向**ECI主机**发送一条客户端请求消息，它将在适当的时候由**ECI主机**进行排队和处理。异步调用将提供仅带有一个基本结果（消息标识符或错误）的立即返回。**ECI主机**稍后将提供一条**主机响应**消息，报告由**ECI客户端**发起的**ECI主机**的操作状态和结果。

3) **异步**主机发起的交换：**ECI主机**向**ECI客户端**发送一条消息，它将在适当的时候由**ECI客户端**进行排队和处理。异步调用将提供近带有一个基本（标准）结果的立即返回消息。该消息的类型和格式（如**ECI主机**中所示）不在本建议书的讨论范围内，因为这是一个**ECI主机**内部问题。

   只定义了ECI客户端的表示。ECI客户端稍后将提供一个响应消息，报告由ECI主机发起的ECI客户端的操作状态和结果。

图9.2.3-1显示了ECI主机与ECI客户端之间不同类型的消息交换。

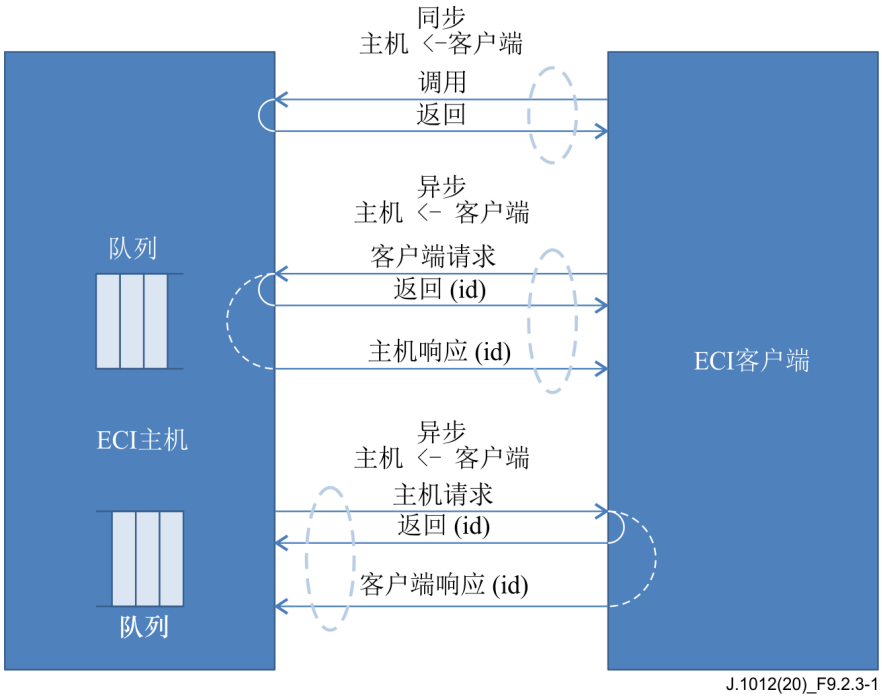


图9.2.3-1 – 客户端和主机之间的消息交换

ECI客户端必须确保有效载荷根据需要受到保护，例如，控制字和内容属性。此外，接口非设计和计划用于内容交换之目的。

ECI客户端应根据第9节中介绍的API定义，使用**响应**中的**请求**标识符，对其支持的**ECI主机请求**做出响应。

ECI主机应根据第9节中介绍的API定义，使用**响应**中的**请求**标识符，对其支持的**ECI客户端请求**做出响应。

异步**请求**可以有选择地指出不需要任何**响应**。例如，当移去许多数据项目时，假设所有的中间数据项目均被正确处理，则发起者仅需要在最后的**请求**上有一个**响应**。

所有异步**ECI主机请求**和**ECI主机响应**都按“其出现的顺序”进行排队。

9.2.4 为ECI客户端提供的动态资源

[b-ITU-T J Suppl. 7]中提出了**ECI客户端**所需最小动态资源的技术参数。涵盖以下项目：线程、栈空间、堆空间、执行时间、NV存储和客户端之间的通信。

9.2.5 API版本管理

允许本建议书中定义的API具有多个版本，例如，出于提供增强功能的目的，以替代先前的功能或解决规范的缺陷。在初始化时，**ECI客户端**及其**ECI主机**需要确定其对应端支持哪些API，并选择在**ECI客户端**生命周期的其余部分时间将使用对应端各可用API的哪个版本。**ECI客户端**在初始化阶段不能使用发现API以外的其他API，因为消息版本（即其可用性、长度和语法）在发现过程完成之前尚未定义。

API版本在其语义中是自包含的：即，通过API版本的**ECI客户端**与**ECI主机**之间的消息交互既不依赖于**ECI主机**中对该API其他版本的支持，也不依赖于**ECI主机**与使用该API其他版本的其他**ECI客户端**之间的交互。

注 – 由于实际原因，定义新API版本的条款文本可以参考本建议书中定义旧API版本的文本。

API是强制的、可选的或有条件的（即强制性受限于某个条件）。一个条件性的示例是，PVR相关的API需要支持CPE（它支持PVR功能）。本建议书的未来版本可以定义由**ECI主机**和**ECI客户端**（引用配置文件名称和规范版本号）支持的API配置文件。

为了符合本建议书的要求并确保后向兼容性，支持API的**ECI主机**或**ECI客户端**应支持该API的所有版本（包括最新版本），除非在本文件的（未来版本）中明确弃用旧版本或另有明确声明。

注2 – 创建本建议书的未来版本并不意味着部署的或新的**ECI客户端**和**ECI主机**必须符合要求。任何将**ECI主机**和**ECI客户端**升级到新规范版本或者要求实施新规范版本（适用于新的**ECI主机**和**ECI客户端**）的规则的政策均超出了本建议书的讨论范围。

ECI客户端应该选择其能处理的、**ECI主机**中可用的API最高版本号，反之亦然，**ECI主机**选择其能处理的、**ECI客户端**中可用的API最高版本号。这鼓励向更成熟的API做前向版本迁移，并鼓励在弃用（旧）API版本的情况下避免遗留问题。

考虑到**ECI主机**的生命周期通常较长且相对容易更新**ECI客户端**，因此**ECI客户端**应该能够支持反映已安装基本情况的旧**ECI主机**API版本（可能需要达成进一步协议，这超出了本文件的讨论范围）。反之亦然，新的**ECI主机**应该支持反映**ECI客户端**部署的旧**ECI客户端**（可能需要达成进一步协议，这超出了本建议书的讨论范围）。

ECI客户端-ECI主机发现API在第9.4.2节中进行定义。

9.2.6 响应性监控

ECI主机应部署一些基本的自动**ECI客户端**重启功能，以便提供整体**CPE**功能之外的稳健性。**ECI主机**应检测**ECI客户端**中的致命错误条件，并应自动在这些事件中重新初始化**ECI客户端**。**ECI客户端**使用的所有资源都将在重新初始化之前予以释放，包括**媒质句柄**、mmi会话、文件、IP连接等。

定义了以下错误条件：

- **ECI主机**应监控**ECI客户端**代码执行的任何非法指令，如未定义的指令操作码、寻址非法数据或寻址不存在的代码、溢出或寄存器堆栈溢出等。
- **ECI主机**应在**ECI客户端**接受新消息时使用超时。有关该参数的提议数据在[b-ITU-T J Suppl. 7]中给出。

在反复重新初始化的情况下，**ECI主机**可以使用可能涉及用户设置或用户输入的策略进行解码，以更持久地排除重复发生故障的**ECI客户端**。

注 – **ECI客户端**执行sys_exit系统调用（参见[ITU-T J.1013]）将被理解为**ECI客户端**的常规终止。通常这意味着**ECI客户端**可以被删除或被更新版本替代。**ECI主机**不会根据此类事件自动删除**ECI客户端**，而是等待，直至通过**ECI客户端**的其他管理策略调用一个适当的替换或删除过程。

9.3 ECI客户端API的机制

9.3.1 异步消息语法

所有消息结构都是根据其在**ECI** VM中的外观来定义的。在表9.3-1中，所有异步消息的消息缓冲区结构都是以其在VM存储器映射中的外观来表示的。注意，所有的消息缓冲区都是32位对齐的。

表9.3-1 – 异步消息语法

C风格的语法	位数
struct messageBuffer { uint32 **msgTag**; uint16**msgId** uint16**payloadLen;** uint32**payload[];** } MessageBuffer;	 32 16 16 n*32

msgTag：

该字段表示以下值：

- 位0-15：**msgApiTag**。消息的API标识（定义见附录C）。
- 位16-23：**msgCallTag**。API调用标识，由接收方在**msgTag**值和商定的API版本情形下进行解释。
- 位24-31：**msgFlags**：用于限定消息的附加标志。以下定义适用：
  - 位24：**msgNoResFlag**：用于**请求**和调用消息：如果0b1，则无需任何**响应**或应答；如果0b0，则需要**响应**或应答。该位在应答和响应消息中没有任何意义。
  - 位25-31保留以供未来使用；这些位应由消息的发起者设置为0b0。

对**请求**消息的**响应**和调用消息的应答，消息标记应相同。

msgld：

- 由**ECI主机**分配的消息的消息标识符值。对于响应消息，这应对应于原始请求消息的值。通过**ECI客户端**发送请求，可以使该字段不被初始化（该值将由**ECI主机**进行分配，并作为SYS_PUTMSG系统调用的结果值予以返回）。

payloadLen：

- 有效载荷长度字段表示有效载荷缓冲区以字节为单位的大小。实际分配的有效载荷字段大小应为该值四舍五入至4的下一个倍数或更大值。**ECI主机**应在解释接收到的消息的**有效载荷**字段时，验证数据没有超出**payloadLen**；否则应返回一个错误。**ECI客户端**可以假定**ECI主机**提供适当尺寸的消息缓冲区。

有效载荷字段：

- 有效载荷字段用于承载消息参数。有效载荷的结构使用函数调用签名的c语法来定义，该函数使用了第9.3.2.3节中定义的特定映射规则。

9.3.2 异步消息布局定义约定

9.3.2.1 消息定义的语法

异步消息使用c风格的函数签名声明来定义。该记法对应于通过本节中定义的规则的消息布局。下面是函数签名声明的一个例子：

```
reqSetTimer(uint32 time, uchar priority)
```

9.3.2.2 基本消息参数类型

语法应使用表9.3.2.2-1中规定的参数定义的基本类型。

表9.3.2.2-1 – 用于消息参数定义的基本类型

基本类型	代表
uint8, uchar, byte:	8位无符号整数
int8, char, bool:	8位有符号整数
uint16, ushort:	16位无符号整数
int16, short:	16位有符号整数
uint32, uint:	32位无符号整数
int32, int:	32位有符号整数
uint64, ulong:	64位无符号整数
int64, long:	64位有符号整数
char *, … ,long *（客户端内存）	32位；只允许同步消息

对于布尔类型的参数，使用符号值“**真**”和“**假**”。根据c语言定义，“**假**”由0x00来表示，“**真**”由0x00以外的任何值来表示。

9.3.2.3 消息有效载荷到消息参数的映射

有效载荷字段包含消息的所有参数。**msgId**消息标识符参数和**msgResult**结果参数是隐含的，即它们不出现于函数签名声明性语法描述中。它们的存在由消息类型隐性地来定义。

ECI主机应将**msgId**与**ECI主机**和**ECI客户端请求**消息相关联，以便将**请求**与相应的答复关联起来。msgId的类型是uint32。msgId值的管理由**ECI主机**来负责。在传送**响应**消息之前，msgId值不得重新发布。

响应应包含int32类型的**msgResult**参数。

这些隐含参数是消息缓冲区有效载荷字段中的第一个参数。表9.3.2.3-1从**ECI客户端**角度（**ECI主机**角度超出了**ECI**中的讨论范围）提供了每种消息类型的有效载荷字段参数序列。

表9.3.2.3-1 – 消息类型和“隐藏”参数（客户端透视图）

消息类型	隐含参数	有效载荷字段
客户端请求，C→H	无	p_1, .. , p_n
主机响应，H→C	msgId, 结果	msgId, 结果, p_1, .. ,p_n
主机请求，H→C	msgId	msgId, p_1, .. , p_n
客户端响应，C→H	msgId, 结果	msgId, 结果, p_1, .. ,p_n

应使用以下规则来将参数（它为结构、字节和短数组等）转换为**ECI客户端**内存空间中消息缓冲区的有效载荷布局：

- 参数映射到内存中，其最低地址在先，除了变长数组的数据字段。
- 任何8位或16位数据类型都将使用适合其类型（有符号或无符号）的扩展名来扩展到32位。
- 结构（不包括位字段）：所有字段都应按照其定义的顺序进行映射，字段大小对齐（对于16位和32位实体）最低地址上的第一个字段，在紧随其后的较大字段前填充字段。该结构始终填充至下一个32位边界。联合结构应填充至替代者的最大尺寸。

- 字节（8位）、短（16位）和int（32位）数组：应包含在消息缓冲区中（而不是指向**ECI客户端**存储器的指针）。固定长度数组应使用以下记法<type>、<array_identifier>、'['<constant>'] '。这些应按它们在参数列表中出现的顺序予以映射。变长数组应使用记法<type>、<array_identifier>、'[']'。所有可变长度数组应映射到两个32位字段。第一个字段包含消息缓冲区中（阵列第一个元素之所在）的偏移量。第二个字段包含数组的长度（以字节为单位）。
- 首先将64位实体存储为最重要的32位（遵循用于在32位小端机中映射64位实体的典型约定）。
- 所有32位和16位实体在内存中都应具有自然的（未知的 - 由底层CPU架构定义）字节顺序表示。
- 指向可打印字符的任何（char *）都应使用UTF-8表示[ISO/IEC 21320]作为实际的“代码点”，除非另有明确定义。字符表示可以是1到4个字节（取决于代码点）。本规范没有定义**CPE**中哪些代码点是可打印的（对不同区域可能有不同的实现）。

注 – 并在发现过程中，**ECI主机**负责解释消息标签以及**ECI客户端**同意的API版本。同样，并在发现过程中，**ECI客户端**负责解释消息标签以及**ECI主机**同意的API版本。

9.3.2.4 异步消息的命名约定

函数名称的约定：

所有函数名称都应以反映消息类型的三个字母指示开始。函数的<name>应以一个大写字母开头。以下按其类型定义消息的名称约定：

```
req<name>():   request message; res<name>():   response message;
```

例 1：reqIpTcpSend().

消息对记法的约定：

请求和**响应**消息被定义为一对，并且类似地用于调用和应答消息。以下记法用于指代这样的消息对：

```
<requestMessage>   →   <responseMessage>
```

例 2： reqIpTcpSend(socket,buffer) → resIpTcpSend(socket).

简洁起见，函数签名可出现在这些记法和其他记法中，而无需参数输入。

表9.3.2.4-1提供了一些实际消息名称的例子，映射至使用过程样式、JavaScript类似事件预约/回调类型编程方法或调度循环中可能的c函数。**subscr**函数允许在接收带有标签的消息时进行函数调用。提供了两个示例：第一个示例对**msgId**标识符具有选择性，并包含该函数的**cntxt**结构；第二个示例不会在**msgId**上进行过滤，也不会在回调/调度时提供**cntxt**结构。

表9.3.2.4-1 – 每个消息类型的有效载荷字段中的参数，参数p1，...，参数pn

消息	程序如记法	客户端事件回调订阅	客户端回调/调度记法或调用
Req, C→H	id = reqName([tag],p_1,..p_n)		
Res, H→C	res = resName([tag],id,p_1,..p_n)	subscr(tag,id,resName,cntxt) subscr(tag,resName)	resName(cntxt,res,p_1,..p_n) resName(id,p_1,..p_n)
Req, H→C	[tag =] reqName([id],p_1,..p_n)	subscr(tag,invName)	invName(id,p_1,..p_n)
Res, C→H	resName([tag],id,res,p_1..p_n)		

9.3.3 同步消息

同步消息采用相同的记法约定，使用函数名称作为异步消息。同步消息参数不应串行化，以适应消息缓冲区，而应使用通用的c约定进行函数调用，并使用VM应用程序二进制接口定义来过程映射至VM内存和寄存器状态。这允许同步消息直接映射至常规C函数，作为**ECI客户端**库的一部分。

有三种预定义类型：**get**（读取）以读取**ECI主机**域中的变量，**set**（设置）以在**ECI主机**域中写入变量，并使用负错误代码或非负函数值返回的通用函数调用（**call**），如表9.3.3-1所示。

表9.3.3-1 – 同步函数类型

类型	适用于	记法	结果	语义
获取	主机变量	getVariable((**i1**..**in**)	变量类型	读取由**ECI主机**域中参数i1..in索引的变量（对该**ECI客户端**）（参见注释）。
设置	主机变量	setVariable((**i1**..**in,** value)	空	将值分配给由**ECI主机**域中参数i1..in索引的变量（对该**ECI客户端**）（参见注释）。
调用	主机	callFunc(p1..pn)	整数或空	对**ECI主机**域中的函数进行（通用）同步调用。返回值与异步消息的结果值类型相同：即负值表示发生了错误。一些函数可能有一个void类型 - 不允许错误信号。
注 – 除了作为Get函数调用的结果返回请求的对象外，还可以触发**ECI主机**执行操作。				

同步消息定义的示例：

```
uintgetClock();
void setPwrWakeup (int timeout);
void memcpy(char *p1, char *p2; int len) ;
```

使用示例：

```
uint clock = getClock() ;        /* read clock */
setPwrWakeup (1000);                /* set wakeup timer; triggers
invocations */
(void) memcpy(ptr1,ptr2,100*1000)   /* copy client memory efficiently */
```

9.3.4 返回的错误代码

响应、**答复**和（如适用）**调用**的返回代码参数应包含一个32位有符号整数。如果返回值为零或正数，则代码的执行成功。如果发生错误，则返回负值。错误是通用的（请参阅表9.3.4-1）或是**请求**特定的（请参阅每个**请求**的特定错误代码）。

表9.3.4-1 – 返回消息的错误代码

名称/常量	值	描述
	1..最大整数	成功的**请求**，由消息定义定义的值。
ErrReqOkNoId	0	成功的**要求**。
ErrReqApiErr	-1	不支持由msgApiTag指定的API。
ErrReqCallErr	-2	在不支持的msgApiTag指定的API内调用。
ReqQueueErr	-3	排队消息时出现问题，**ECI**缓冲区队列溢出。
ReqResource	-4	处理**请求**时发生资源问题（例如，由于消息过多而导致的内存问题）。
RFU	-5..-15	保留以供未来使用（通用错误类型）。
ReqParam<N>Err	-16..-48	参数**N**= -Result-15中的错误。
保留用于VM错误	-49..-64	如[[ITU-T J.1013]]中定义的那样，错误代码被保留用于VM特定的错误。
RFU	-65 .. -256	保留以供未来使用。
API specific error	-256 .. -511	由API错误代码表定义的API特定错误。
RFU	-512..最小整数	保留以供未来使用。

注 – 典型地，**ECI客户端**可以依赖**ECI主机**来支持第9.2.5节中定义的API特定的配置文件，并将消息的缓冲区排序为自由。因此通常不需要智能错误处理；错误代码通常仅用于**ECI客户端**调试场景。

API特定的错误代码或ReqParamNErr不能作为返回的一部分返回，但这种错误应作为**响应**的一部分发出信号。

9.3.5 安全认证通道（SAC）

高级安全API（参见第9.5.2节）提供了在**ECI客户端**与任何其他适当设备之间建立**安全认证信道（SAC）**的工具。在**ECI客户端**需要与另一个**ECI客户端**或任何外部设备进行安全认证通信的情况下，它需要定义一个专有机制，该机制可以利用可用的API，尤其是高级安全API。

9.3.6 通过ECI主机进行的消息验证

为了避免由于不合适的**请求**或**响应**而导致的错误条件或不当行为，**ECI主机**应对从**ECI客户端**收到的任何消息进行全面检查。应做以下检查：

- 支持msgApiTag。
- 在API消息空间内支持msgCallId（在发现时建立API版本的情形下）。
- 验证关于有效载荷的约束、特别msgLength是否与消息的语法规则相匹配，并且ECI主机读取或写入的ECI客户端地址空间的消息缓冲区（用于异步消息）和任何内存，限制于ECI客户端地址空间中已做定义的部分。
- 验证是否有任何消息特定的预置条件失败（预置条件对请求或相应的完整性而言至关重要）。
- 验证消息中是否有任何指针或内存是分配给ECI客户端的内存。

9.3.7 通过ECI客户端进行的消息处理

任何分配用于发送**请求**的内存都可以在返回时重新使用，除非另有明确说明（通常避免复制的大消息是重要消息）。同样，任何分配用于发送**响应**的内存都可以在发送事件之后立即重新使用。

ECI客户端不应依赖**ECI主机**为每个**请求**返回一个**响应**。

ECI客户端可以对任何**ECI主机请求**或**响应**的正确语法进行验证。在格式错误的**请求**或**响应**情况下，**ECI客户端**没有义务响应以向**ECI主机**提供任何反馈。

9.4 通用ECI主机资源的API

9.4.1 第9.4节中定义的API列表

图9.4.1-1 – 第9.4节中定义的API的框图

表9.4.1-1 – 第9.4节中定义的API列表

条款	API名称	描述
9.4.2	主机接口发现	允许**ECI客户端**识别**ECI主机**提供的接口
9.4.3	用户接口	允许**ECI客户端**与用户建立通信
9.4.4	IP堆栈	允许主机建立到外部IP设备的IP链路
9.4.5	文件系统	允许**ECI客户端**将数据存储在**ECI主机**的RAM内存中
9.4.6	时间 /时钟	允许**ECI客户端**访问**ECI主机**的时间和日期信息
9.4.7	能源管理	允许**ECI客户端**与**ECI主机**电源管理系统进行通信
9.4.8	国家和语言设置	允许**ECI客户端**读取**ECI主机**中的国家和语言设置

表9.4.1-1显示了第9.4节和图9.4.1-1中定义的API，它说明了使用**ECI体系结构**在第9.4节中定义的API的位置。

以表9.4.1-2中所示的结构，在表中概述了每个API与不同API有关的表示消息。

表9.4.1-2 – 汇总各个API消息功能的表的结构

信息	类型	方向	标签	描述
消息的名称	参见表9.4.1-3	C→H or H→C	标签值	消息功能的简短描述

表9.4.1-2中的列类型给出了相关消息的类型，它可以是同步的也可以是异步的。更多细节在表9.4.1-3中给出。附录I给出了**ECI客户端**可用的、有关所有API消息的一个完整列表。

表9.4.1-3 – Type列的可能值

消息类别	类型列中的表示法	评论
异步消息	A	可能的消息类型：参见表9.3.2.3-1.
同步消息	A	可能的消息类型：参见表9.3.3-1
	设置	
	获取	
	调用	

9.4.2 用于访问ECI主机接口发现资源的API

9.4.2.1 引言

本节定义**ECI客户端**可用于发现**ECI主机**支持之API和API版本的API，并在**ECI客户端**与**ECI主机**的会话期间选择最合适的版本。API版本管理机制允许以API为基础在API上选择API。一旦选择了一个API版本，它将一直保持使用状态，直到**ECI主机**发生下一次**ECI客户端**初始化事件。

第9.2.5节讨论了有关API可用性的策略。第10节中定义了强制性API。

ECI客户端在初始化后应立即启动版本管理：没有（相互）建立的版本，不能使用任何API。

API的版本应由16位数字表示。API版本编号从0x0000开始。新版本的定期分配是递增式的（每次递增1）。

表9.4.2.1-1列出了API消息。

表9.4.2.1-1 – ECI主机接口发现API

消息	类型	方向	标签	描述
getApis	获取	C→H	0x0	获取可用的主机API
getApiVersions	获取	C→H	0x1	获取可用的主机API版本
setApiVersion	设置	C→H	0x2	设置要使用的主机API版本

9.4.22 getApis消息

C→Huint[] **getApis (**uint**maxNrApis**)

- 该请求返回一个指示**ECI主机**支持之API的maxNrApis位数组。

属性定义：

- 带标签a（a<maxNrApis）的API的主机API可用性被认为是((*result*[a/32]>>(a%32))&0b1 == 0b1)。

参数定义：

maxNrApis： ushort	返回结果的最高API编号加1。

9.4.23 getApiVersions()消息

C→Huint[] **getApiVersions (**uhort**api,** ushort**maxNrVersions**)

- 该请求返回一个maxNrVersions位数组，指示**ECI主机**支持的api版本。

属性定义：

- 版本v（v<maxNrVersions）带标签**api**的API版本可用性被认为是(**(*result*[v/32]>>(v%32))&0b1 == 0b1**)。

参数定义：

maxNrVersions：ushort	结果中返回的最高版本号加1。

9.4.2.4 setApiVersion()消息

C→HsetApiVersion (ushort**api,** ushort**version**)

- 该消息设置在**ECI客户端**与**ECI主机**之间使用的**api**版本。应该只调用一次（后续调用不起作用）。

参数定义：

api：ushort	要为其设置版本的API的标签。
version：ushort	在客户端与主机之间后续会话中使用的**api**的版本号。

详细的语义：

- 如果**版本**不是**api**支持的现有API版本，则应将API版本设置为API支持的第一个更高API版本（如果可用的话）或者最高API版本。
- 在对该API版本执行初始化之前，**ECI客户端**应检查API版本的可用性。

注 – 不明确检查可能会发生意外的API行为或错误条件。

9.4.3 用于访问ECI主机用户接口资源的API

9.4.3.1 引言

本节定义了**ECI**应用程序的应用环境，允许**ECI客户端**与**用户**建立一个交互接口。**ECI**应用程序由**ECI客户端**托管，并在**ECI主机**上执行。这些应用程序使用HTML浏览器，该浏览器可在电视设备中用于设备供应商和广播公司的多种平台。

图9.4.3.1-1描述了**ECI**应用程序环境中的各个实体。**ECI客户端**不会控制以及直接与它启动的**ECI**应用程序进行通信；它使用**ECI主机**提供的一个代理。该代理实现第9.4.3.4节中定义的API，它允许**ECI客户端**启动和停止**ECI**应用程序，并与正在运行的**ECI**应用程序进行通信，例如处理**用户**输入。**ECI**应用程序与**ECI客户端**的通信由代理来处置，方法是将浏览器的HTTP GET请求转码为来自应用程序容器的资源，或者对第9.4.3.4.8节中定义之**ECI客户端**的reqUiClientQuery API请求。后者可以向**ECI客户端**提供**用户**的输入，并允许**ECI客户端**提供具有动态内容的响应。应用程序容器提供（更大的）静态资源，以构建UI屏幕；**ECI客户端**向UI屏幕提供自定义的输入，并接收**用户**输入。

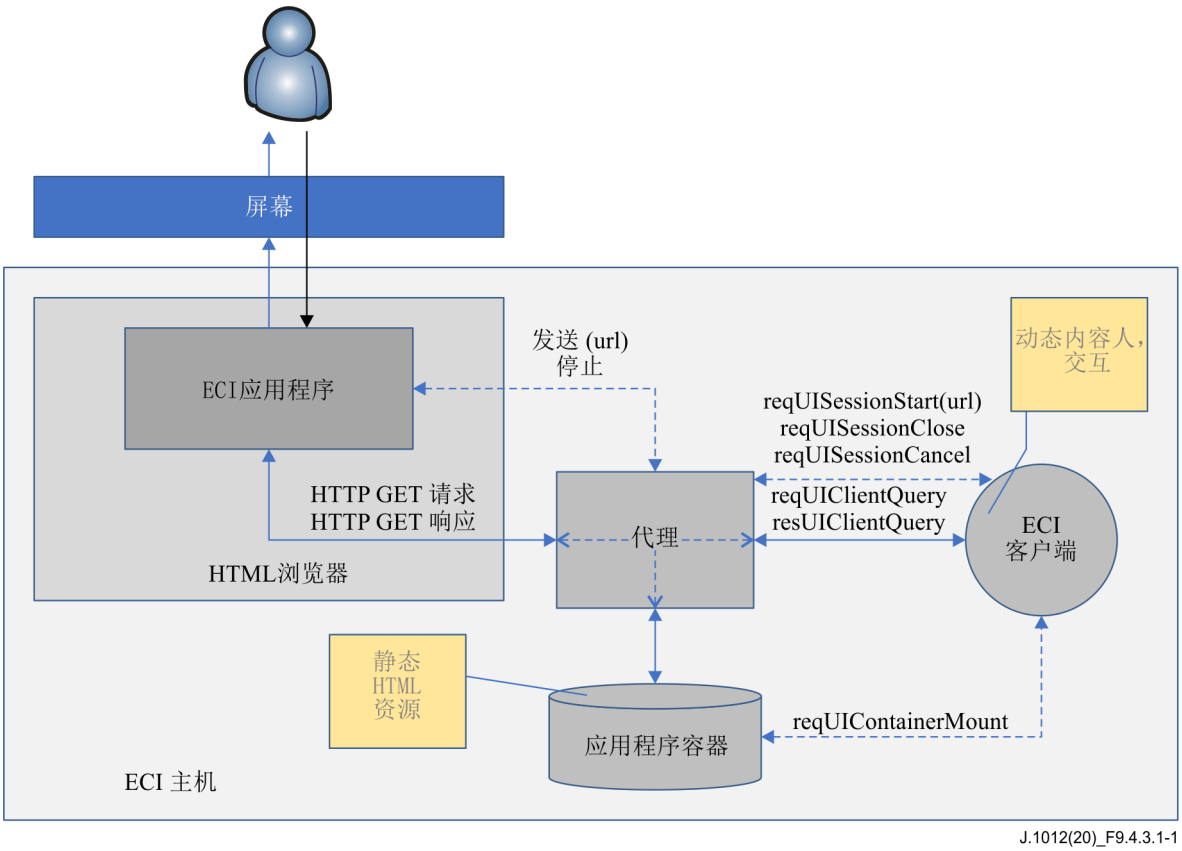


图9.4.3.1-1：用户接口API的框图

9.4.3.2 用户接口环境

9.4.3.2.1 浏览器配置文件

ECI主机应提供一个HTML浏览器，用于实现[IEC 62766-5-2]中定义的万维网标准电视配置文件，符合本建议书中定义的约束和扩展。该配置文件也被HbbTV系统[b-HbbTV]所采用。

9.4.3.2.2 约束条件

ECI主机应拒绝对不源自该**ECI应用程序**会话的任何**ECI应用程序**会话资源的HTTP请求。

用于将**ECI应用程序**资源加载到浏览器中的URL应是会话唯一的基本URL和相对URL的串接，以处理**ECI客户端**或应用程序容器。例如，如果会话基准URL是：

```
http://localhost:3000/session-x/
```

并且应用程序容器中的资源是：

```
main/pincode.html
```

则浏览器URL是：

```
http://localhost:3000/session-x/main/pincode.html
```

当从HTML浏览器提供请求时，**ECI主机**应从其文件扩展名中推断出**ECI应用程序**资源的内容类型，并应至少支持：

- text/html - .html and .htm

- text/javascript - .js
- text/css - .css
- image/png - .png
- image/gif - .gif
- image/jpeg - .jpg and .jpeg

9.4.3.2.3 浏览器功能

9.4.3.2.3.1 显示模型

浏览器窗口应为全屏。浏览器窗口的尺寸至少应为1 280 x 720像素。应编写一个**ECI**应用程序，以便它适当地扩大尺寸。

显示**ECI**应用程序的图形平面应置于终端应用程序的图形平面之后，并应位于任何其他图形平面之前，包括用于视频、字幕和广播应用程序的图形平面。

用于**ECI**应用程序的平面完全覆盖除终端之外的任何图形平面。浏览器窗口的背景应该是透明的，即如果一个区域没有被**ECI**应用程序的任何HTML元素覆盖；下面的图形平面（其中一个通常包含广播视频）应该是可见的。如果主体元素的CSS属性背景颜色设置为透明，则浏览器的背景窗口应是透明的。

当终端需要临时覆盖**ECI**应用程序时，例如，要在**用户**操作中显示系统菜单或频道信息横幅，**ECI**应用程序将失去输入焦点。如果**ECI**应用程序失去输入焦点，则应以Window对象作为目标来发送模糊事件。

当终端关闭其UI并且**ECI**应用程序仍在运行时，它应重新获得输入焦点。如果**ECI**应用程序获得输入，则应以Window对象作为目标来发送焦点事件。浏览器应支持RGBA32作为彩色格式。

9.4.3.2.3.2 文本和字体

浏览器应包含嵌入的比例字体。**ECI**应用程序可以使用'sans-serif'或'default'作为通用字体系列名称来选择字体，以选择嵌入字体。嵌入字体支持的字符集应该适用于部署设备的区域。**ECI**应用程序可以使用[IEC 62766-5-2]中定义的CSS3万维网字体来使用替代字体和字符集。对每个**ECI**应用程序，浏览器应支持至少一种可下载的万维网字体。

浏览器应支持**ECI**应用程序的所有文本资源（即HTML文档、脚本和样式表）的UTF-8编码。

9.4.3.2.3.3 图形格式

浏览器应支持使用以下格式的图形：GIF [W3C GIF V89a]、JPEG [ITU-T T.871]和PNG [W3C PNG]。

9.4.3.2.3.4 用户输入

浏览器应通过使用DOM3 KeyboardEvents的远程控制来支持**用户**输入。当**ECI**应用程序正在运行且具有输入焦点时，**ECI主机**应允许**用户**启动以下事件：

- 数字键：0-9
- 光标键：左、右、上、下、回车和浏览器返回

不需要对传统属性keyCode和charCode的支持。

9.4.3.2.3.5 存留

浏览器应支持WebStorage API和会话cookie的会话存储。**ECI客户端**应使用其内部存储

器来保存浏览器会话中的信息。

9.4.3.2.3.6 ECI应用程序访问静态HTML资源

接收来自**ECI应用程序**的HTTP请求的代理，应将相对URL（即来自会话基本URL的扩展）映射至由**ECI客户端**挂载的应用程序容器中的相对路径。从相对URL到文件的映射是直接的：相对URL directoryname1 / directoryname2 /.. /directorynameN /文件名被映射至包含在目录directorynameN / directoryname2 /.. / directoryname1中的文件名。

应用程序容器目录结构和文件应符合以下约束条件：

- 所有文件名和目录应由字母数字字符以及字符“.”（点）和“_”（下划线）组成，不得超过40个字符。

[b-ITU-T J Suppl. 7]中提议了有关应用程序容器的更多资源或性能要求。

9.4.3.2.3.7 ECI客户端与ECI应用程序之间的通信

浏览器支持本建议书第9.4.3.2.1节要求的XmlHttpRequest。**ECI**应用程序与**ECI客户端**之间的通信通过**ECI主机**的代理进行路由。**ECI**应用程序可以使用本节中定义的XMLHttpRequest API来执行HTTP GET请求。HTTP请求的URL应构建自第9.4.3.2.2节中定义的**ECI**应用程序会话的基本URL和相对URL'/ client'。任何参数都应作为密钥对查询串的一部分。密钥和值只能由ASCII字符组成。密钥的最大长度为31个字符，值的最大长度为255个字符。

示例： http://localhost:3000/ session-20170303-163100-01/client?id=e4f0&p2=v2。

在接收到HTTP请求代理时，**ECI主机**应按照第9.4.3.4.5节中的定义，以解析查询字符串作为密钥值对，将reqUiClientQuery消息发送给**ECI**应用程序的**ECI客户端**。**ECI客户端**对主机的响应应包含以下参数：

- 类型：符合相关标准定义并在IANA媒质类型数据库[b-IANA]中归档的媒质类型的字符串，例如，由[b-IETF RFC 8259]定义的application/json。
- 状态代码：在GET请求的响应中使用的一个整数，即成功应该是200。
- 主体：最大64 kB的字符串。

ECI主机应通过将内容类型报头设置为类型参数、将HTTP状态设置为错误值、响应主体设置为主体参数值，从而为浏览器构建HTTP GET响应。

与源自**ECI客户端**的HTML应用程序之间的通信超出了本版当前建议书的讨论范围。

9.4.3.3 应用程序生命周期

9.4.3.3.1 启动一个ECI应用程序

电视屏幕是由终端、广播、**运营商**和第三方应用程序构成的共享资源。该版本的当前建议书定义了操作**ECI**模块所需的基本**用户**接口的应用环境，例如，PIN入口、订阅信息等。

从**ECI客户端**发出对**ECI主机**的请求应限于以下情况：

- **ECI主机**即将开始正在由**ECI客户端**处理的媒质呈现（例如，在调谐到广播频道之后）。
- **ECI主机**正在呈现由**ECI客户端**处理的媒质。
- ECI主机请求ECI客户端显示其应用程序菜单。

- **ECI客户端**表明它希望启动一个非内容流相关的**ECI应用程序**，并且**ECI主机**可确保该对话符合**用户**请求或者与屏幕上的内容不冲突：即不存在第三方的移除/停电或屏幕覆盖，第三方内容被选择供**用户**查看。

对于上述情况，**用户**执行第9.8.2.11节中定义之委托父认证交互的启动请求，被认为是**ECI客户端**发起的一个启动请求，它发出第9.8.2.10节中定义的原始父认证请求。

屏幕冲突定义为**ECI客户端**请求**ECI主机**启动**ECI应用程序**（打开一个UI会话）但上述启动条件未得到满足的情况。

如果**ECI主机**能够运行交互式应用程序，则**ECI主机**应能够启动至少一个**ECI应用程序**，同时运行与屏幕上显示的媒质相关的交互式内容。这种**ECI应用程序**应与屏幕上呈现的媒质直接相关。启动**ECI应用程序**不应终止屏幕上显示的交互内容，并且该内容应能够在**ECI应用程序**停止时恢复与**用户**的交互。

ECI主机应传达**ECI客户端**希望启动非内容流相关的**ECI应用程序**，以引起**用户**注意或者允许**ECI客户端**定期启动此类**ECI应用程序**而不发生**屏幕冲突**。例如，可以通过在启动或待机时启动此类**ECI应用程序**，或者使用某些用户操作来响应定期显示的横幅或**ECI主机**菜单屏幕中的关注图标，来完成此操作。**ECI客户端**不应承担频繁启动此类**ECI应用程序**的能力，并应将其目的限制在对**ECI客户端**的持续操作至关重要的事项上。

当由**ECI客户端**启动时，**ECI应用程序**应加载到浏览上下文中，这些上下文无法从广播或任何其他第三方应用程序的浏览上下文来访问。

浏览器窗口应在一秒内可见，并应该已完全加载**ECI应用程序**。

本建议书的未来版本可能提供扩展的生命周期模型和冲突解决机制，并允许与外部启动的HTML应用程序进行通信。

9.4.3.3.2 终止一个ECI应用程序

为停止**ECI应用程序**，**ECI客户端**会向**ECI主机**发送一条reqUISessionStop消息。该请求包含由**ECI主机**在resUISessionOpen响应中返回的uiSessionId。应停止**ECI**应用程序。这是如何实现的取决于实施方案，例如，通过停止或最小化浏览器。在任何情况下，**ECI**应用程序都应放弃输入焦点，并且浏览器不得再向**ECI**应用程序发送KeyboardEvents。

如果任何**用户**动作（如按P +/P-）使终端进入禁止**ECI**应用程序启动的状态，则还应停止**ECI**应用程序。**ECI主机**应向**ECI客户端**发送一条reqUiSessionCancel消息。

9.4.3.4 与用户通信相关的API

9.4.3.4.1 用户通信API消息的列表

用户接口API允许**ECI客户端**安装下载的UI应用程序容器文件，以提供生成**用户**接口所需的大量静态HTML资源。代理自动解析从浏览器到应用程序容器文件的所有非客户端定向的HTTP请求。

ECI主机可以建议**ECI客户端**启动一个应用程序，通过响应请求访问**ECI客户端应用程序菜单**的**用户**，或者通过reqUiSessionCommence消息向**ECI客户端**指示，不存在任何阻止它将非**媒质句柄**相关的**ECI应用程序**呈现给**用户**的冲突。**ECI客户端**可以通过setUiClientAttention消息表明，它有兴趣启动这样一个非**媒质句柄**相关的对话。实际上，当没有任何**屏幕冲突**时，这允许从**ECI客户端**到**用户**的较低优先级通信。

所有**用户**接口会话均由**ECI客户端**使用reqUiSessionOpen消息来打开。呈现第一个UI屏幕的相对URL作为参数提供。**ECI客户端**和**ECI主机**都可以分别使用reqUiSessionClose和

reqUiSessionCancel消息来终止用户接口会话。

reqUiClientQuery消息允许浏览器中的**ECI应用程序**通过代理向**ECI客户端**发送带有参数的请求，**ECI客户端**而后可以响应HTML应用程序的数据。此通信允许**ECI应用程序**呈现特定于**ECI客户端**的数据，并以同HTML应用程序与动态HTTP服务器通信的方式，来向**ECI客户端**提供用户输入。

表9.4.3.4.1-1列出了本节中定义的所有API。

表9.4.3.4.1-1 – 用户接口API消息

消息	类型	方向	标签	描述
reqUiContainerMount	A	C→H	0x0	用HTML资源装载一个UI应用程序容器，以支持UI会话。
setUiClientAttention	S	C→H	0x1	**ECI客户端**表示希望开始一个UI会话而不关联到**媒质句柄**。
reqUiSessionCommence	A	H→C	0x2	**ECI主机**建议**ECI客户端**打开一个UI会话。
reqUiSessionOpen	A	C→H	0x3	**ECI客户端**请求打开一个与**用户**的**用户**接口会话，并在屏幕上显示内容。
reqUiSessionClose	A	C→H	0x4	**ECI客户端**结束一个**用户**接口会话。
reqUiSessionCancel	A	H→C	0x5	**ECI主机**取消一个**用户**接口会话。
reqUiClientQuery	A	H→C	0x6	**ECI客户端**从浏览器中的HTML应用程序接收请求，并提供一个（动态）响应。

9.4.3.4.2 reqUiContainerMount消息

C→HreqUiContainerMount(fileName**filename**, PubKey**pk**) **→H→CresUiContainerMount** (uint**indexFileLen**, uchar**indexFile**)

- 该消息允许ECI客户端指示ECI主机指定一个文件，作为包含其ECI应用程序的HTML资源的**ECI客户端**应用程序容器。如果成功，它将返回应用程序容器主目录中的“EciIndex.txt”文件的内容。

请求参数定义：

filename:fileName	ECI客户端文件系统中的文件名，它将成为指定的应用程序容器。
pk: PubKey	用于验证应用程序容器签名的公钥。

响应参数定义：

indexFileLen: uint	索引文件的长度。
indexFile: uchar	索引文件的内容。

详细的语义：

- 下面使用的、之间带有冗长文本的方括号[和]表示ZIP文件容器中字段和结构的划分。
- 验证容器文件的签名位于[中央目录记录结束]结构的[.ZIP文件注释]字段中，参见[ISO/IEC 21320]中引用的、PKWARE®公司的Zip文件格式规范版本6.3.3。
- [.ZIP注释字段]定义在包含所有ASCII字符的以下字符串中：“ECI_SIGNATURE=”，后跟表5.6-1中定义的ECI_Data_Signature结构的值，使用大写字母编码为十六进制字符串，后跟一个“”（一个结束括号）。

例子：

ECI_SIGNATURE="01000000FCB1F60456719035"

类型-1的ECI_Data_Signature的编码数据签名串长度为520个字符。

- **ECI主机**应验证通过容器文件计算的签名，直到[中央目录记录结束]结构中的[.Zip注释字段]，并使用公钥参数pk将[.Zip注释长度字段]设置为0x0000以及为计算签名而定义的第5.6节中的过程。
- 索引文件被定义为容器文件主目录中名称为“EciIndex.txt”的文件。
- 如果UI会话需要，则**ECI客户端**应安装一个有效的UI应用程序容器。
- **ECI客户端**应能够在无法加载和安装UI应用程序容器时向用户显示一条基本的遇险消息。

应用说明：

- 客户端可以使用HTTP(S)API（参见第9.4.4.6节）从在线服务器将应用程序容器文件加载到其文件系统中，或者使用数据轮播API从广播传输流中将应用程序容器文件加载到其文件系统中。
- “EciIndex.txt”文件可以包含UI的版本信息，通过公钥签名进行验证。

关于reqUiContainerMount消息的错误代码在表9.4.3.4.2-1中定义。

表9.4.3.4.2-1 – reqUiContainerMount错误代码

名称	描述
ErrUiContainerFileNot	参见表9.4.3.4.9-1。
ErrUiContainerNot	
ErrUiContainerSignature	
ErrUiContainerIndexTxtNot	

9.4.3.4.3 setUiClientAttention消息

C→HsetUiClientAttention(uint **clientAttention**)

- 该消息表明**ECI客户端**希望打开一个与**用户**的UI会话，而与**媒质句柄**无关（UI会话类型等于EciUiSessionDiaReq，参见第9.4.3.4.4节）。

属性定义：

clientAttention：uint	定义的值是： 0x0：**用户**不需要注意。 0x1：来自**用户**的关注是可取的。 保留所有其他值。

后置条件响应：

- 如果clientAttention = 0x0，则**ECI主机**不会发出任何reqUiClientSessionCommence（uiSessionType = EciUiSessionDiaReq）消息。
- 如果clientAttention = 0x1，若没有这种类型的未决消息，则**ECI主机**将发出一条reqUiClientSessionCommence（uiSessionType = EciUiSessionDiaReq）消息。

9.4.3.4.4 reqUiSessionCommence消息

H→CreqUiSessionCommence (uint **uiSessionType**) **→C→HresUiSessionCommence ()**

- 该消息允许**ECI主机**建议**ECI客户端**打开特定类型的一个UI会话。

请求参数定义：

uiSessionType：uint	**ECI客户端**文件系统中将成为指定应用程序容器的文件名。这些值在表9.4.3.4.4-1中定义。只有值EciUiSessionAppMenu和EciUiSessionDiaReq是允许的。

表9.4.3.4.4-1 – ECI UI会话类型

名称	值	描述
EciUiSessionDiaReq	0x00	**ECI客户端**通过setUiClientAttention消息（不与特定**媒质句柄**关联）请求与最终**用户**的UI会话，并且**ECI主机**可以向其授予来自**ECI客户端**的reqUISessionOpen。
EciUiSessionAppMenu	0x01	**ECI客户端**的**应用程序菜单**。这允许**用户**发起访问可由用户发起的所有相关设置、信息和功能。
EciUiSessionMh	0x02	**ECI客户端**请求的UI会话与**媒质句柄**的操作相关联。
EciUiSessionParAuthDel	0x03	**ECI客户端**请求UI会话代表**媒质句柄**上的处理内容来执行委托父认证对话。
RFU	其他	保留以供未来使用。

注 – 表9.4.3.4.4-1中的值以建议的优先顺序进行定义。该顺序可以提供解决**ECI主机**设计中UI焦点冲突的建议。

详细的语义：

- ECI客户端应能够呈现应用菜单。应用程序菜单应至少允许用户检查ECI客户端的版本、平台操作参考和ECI客户端的操作状态。

先决条件请求：

- 对一个UI会话，对**ECI客户端**应没有任何之前发出的、未决的reqUiSessionCommence消息。

后置条件响应：

- **ECI**客户端应发出一条相应UI会话类型的reqUiSessionOpen消息，或者报告一个错误。

有关reqUiSessionCommence消息的错误代码在表9.4.3.4.4-2中定义。

表9.4.3.4.4-2 – reqUiClientSessionCommence错误代码

名称	描述
ErrUiResourceError	参见表9.4.3.4.9-1。
ErrUiClientError	

9.4.3.4.5 reqUiSessionOpen消息

C→H reqUiSessionOpen(uint**uiSessionType,** ushort**mH,** uint**relUrlLen,** char**relUrl[]) → H→C resUiSessionOpen(**ushort**uiSessionId)**

- 该消息允许**ECI客户端**从**ECI主机**请求新的UI会话。

请求参数定义：

uiSessionType: uint	表 9.4.3.4.4-1 中定义的UI会话类型。如果值是EciUiSessionMh或EciUiSessionParAuthDel，则mH参数应有相关性，否则它应不相关。
mH:ushort	MMI与之关联的内容处理会话的媒质句柄。
relUrlLen:uint	relUrl的长度，以字节为单位。
relUrl:char[]	相对URL，空字符终止。添加到会话基本URL将形成浏览器启动UI会话的URL。参见9.4.3.2.2。

响应参数定义：

uiSessionId：ushort	新UI会话的ID。

详细的语义：

- **ECI**客户端应能够一次处理多个UI会话。但是，只需要支持一个同时会话的UI会话类型EciUiSessionAppMenu或EciUiSessionAppMenu，并且每个打开的**媒质句柄**至多需要一个具有UI会话类型EciUiSessionMh的UI会话。
- **ECI客户端**应能够同时打开类型为EciUiSessionMh的多个UI会话。
- 如果**ECI客户端**支持父认证授权API，则**ECI客户端**应能够同时打开多个UI会话类型EciUiSessionParAuthDel的UI会话。此类UI会话应能够并行进行到**ECI客户端**的其他UI会话。
- 一个**ECI主机**可以支持一个或多个同时进行的UI会话，以适合其**CPE**应用程序模式。

先决条件请求：

1) 如果uiSessionType的值是EciUiSessionAppMenu或EciUiSessionDiaReq，则该消息之前应有一个具有相同uiSessionType参数的reqUiClientCommence消息。
2) 如果uiSessionType的值是EciUiSessionParAuthDel，则该消息应前置有一条从**ECI主机**到**ECI客户端**的媒质句柄mH的reqParAuthDel消息。
3) 如果uiSessionType的值是EciUiSessionMh，则Mh应该是一个开放的媒质句柄会话。

先决条件响应：

1) 如果uiSessionType的值是EciUiSessionAppMenu或EciUiSessionDiaReq或EciUiSessionParAuthDel，则**ECI主机**只能在它先前请求之的情况下接受UI会话请求，请求的原因尚未被缓和，并处于不会导致**屏幕冲突**的状态。
2) 如果uiSessionType的值是EciUiSessionMh，则**ECI主机**应授予UI会话请求，如果它可以与**用户**建立有意义的交互，而不造成屏幕优先级冲突。

3) 当第二个会话的uiSessionType等于EciUiSessionParAuthDel时，**ECI主机**不应拒绝来自**ECI客户端**的第二个会话。允许**ECI主机**取消第一个会话。

应用注释：

1) 如果**媒质句柄**会话用于记录，并且没有任何机会发起与用户的对话，原因是这将导致**屏幕冲突**或没有任何活动的屏幕，则**ECI主机**应拒绝会话。

2) 推荐**ECI主机**应用程序适应父认证UI会话，例如，当使用父认证API的reqParAuthCid消息编程未来可能需要进行父认证的记录时（参见第9.8.2.10节）。

3) **ECI主机**可以取消与**ECI客户端**的UI会话，以允许uiSessionType等于EciUiSessionParAuthDel或EciUiSessionMh的一个新会话。

有关reqUiSessionOpen消息的错误代码在表9.4.3.4.5-1中定义。

表9.4.3.4.5-1 – reqUiClientSessionStart错误代码

名称	描述
ErrUiScreenConflict	参见表9.4.3.4.9-1。
ErrUiNoScreen	

9.4.3.4.6 reqUiSessionClose消息

C→H reqUiSessionClose(ushort**uiSessionId) → H→C resUiSessionClose(**ushort**uiSessionId)**

- 该消息允许**ECI客户端**关闭现有的UI会话。

请求参数定义：

uiSessionId：ushort	要关闭的UI会话的ID。

响应参数定义：

uiSessionId：ushort	已关闭的UI会话的ID。

先决条件请求：

1) 带uiSessionId的UI会话将被打开。

2) 引用uiSessionId的其他消息不应发送到**ECI主机**。

先决条件响应：

1) 不得将引用uiSessionId的其他消息发送到**ECI客户端**。

9.4.3.4.7 reqUiSessionCancel消息

H→C reqUiSessionCancel(ushort**uiSessionId,** uint **reason) →**
C→H resUiSessionCancel(ushort**uiSessionId)**

- 该消息允许**ECI主机**关闭到**ECI客户端**的现有UI会话。该消息旨在在不再满足用于显示**ECI应用程序**的条件的情况下供**ECI主机**使用，例如，如果**用户**切换到一个不同的频道，它属于一个导致**屏幕冲突**的不同的**ECI客户端**。

请求参数定义：

uiSessionId：ushort	要取消的UI会话的ID。
reason：uint	取消会话的原因。这些值在表9.4.3.4.9-1中定义。

响应参数定义：

uiSessionId：ushort	被取消的UI会话的ID。

先决条件请求：

1) 带uiSessionId的会话将被打开。

2) 不得引用uiSessionId发送更多的消息。

先决条件响应：

1) 不得引用uiSessionId发送更多的消息。

9.4.3.4.8 reqUIClientQuery消息

H→C reqUIClientQuery(ushort**uiSessionId,** uint**queryLen,** KeyValPair **query[]) → C→H resUIClientQuery(**ushort**uiSessionId,** uint**statusCode,** uint**typeLen,** char **type[],** uint**bodyLen,** uchar **body[])**

- 该消息传送**ECI主机**浏览器中运行的**ECI应用程序**发出的HTTP请求，如第9.4.3.2.3.7节所述，并允许**ECI客户端**将HTTP响应发送回**ECI应用程序**。

请求参数定义：

uiSessionId：ushort	从中发出请求的UI会话的ID。
queryLen：uint	查询参数的长度（以字节为单位）。
query[]：KeyValPair	包含由浏览器发出之HTTP请求的查询参数关键值对。

KeyValPair的类型定义

```
#define MaxKeyLen 32
#define MaxValLen 256

typedef struct KeyValPair {
   char key[MaxKeyLen];   /* Key of the key value pair, null terminated*/
   char val[MaxValLen];   /* Value of the key value pair, null terminated */
} KeyValPair;
```

响应参数定义：

uiSessionId: ushort	UI会话的ID。
statusCode:uint	在[IETF RFC 7231]中定义的HTTP状态代码。
typeLen:uint	类型参数的长度以字节为单位。
type[]: char	响应类型为空而终止的ASCII字符串。
bodyLen:uint	主体参数的长度以字节为单位。
body[]: uchar	HTTP响应消息。

先决条件请求：

1) uiSessionId已打开。

详细的语义：

- 在来自**ECI应用程序**的格式不正确的查询字符串情况下，**ECI主机**可能会返回HTTP状态码400，而不会向**ECI客户端**提出请求。
- 第9.4.3.2.3.7节定义了与HTTP请求和浏览器响应的消息参数关系。

9.4.3.4.9 用户通信API的错误代码

表9.4.3.4.9-1列出了与**用户**接口通信有关的错误代码。

表9.4.3.4.9-1 – 用户通信API错误代码

名称	值	描述
ErrUiContainerFileNot	-256	找不到任何UI应用程序容器文件。
ErrUiContainerNot	-257	文件不是有效的UI应用程序容器文件。
ErrUiContainerSignature	-258	对应用程序容器文件的签名检查失败。
ErrUiContainerIndexTxtNot	-259	应用程序容器顶部目录中没有“EciIndex.txt”文件。
ErrUiResourceError	-260	**ECI客户端**无法安装UI应用程序容器资源。
ErrUiClientError	-261	**ECI客户端**未处于可呈现UI的操作状态。
ErrUiDiaNoMore	-262	来自**ECI客户端**的对话请求不再有效。
ErrUiScreenConflict	-263	**ECI主机**发生**屏幕冲突**，无法容纳或维持会话。
ErrUiNoScreen	-264	**ECI主机**没有或无法再访问用于UI会话演示的屏幕。
RFU	其他	保留以供未来使用。

9.4.4 用于访问ECI主机IP堆栈资源的API

9.4.4.1 引言

在配备有IP堆栈的**CPE**中，**ECI主机**代表**ECI客户端**提供互联网接入服务。**ECI客户端**可以使用UDP/IP来发送消息，并使用**ECI主机**在**ECI客户端**和服务器模式下向对等端打开TCP/IP连接。**ECI主机**名称可以使用**ECI主机**中的可用DNS服务解析为IP地址。

所提供的服务不能超越**CPE**本身的通用软件安全性。即，如果**ECI主机**外部的**CPE**软件受到威胁，则任何IP通信都可能被篡改。

用于IP连接的**ECI客户端**API基于许多当代操作系统中使用的BSD套接字范例。

API的定义分为四个部分：

1) 基本的**ECI** IP套接字和DNS功能（第9.4.4.3节）。
2) 使用**ECI** IP套接字的UDP/IP通信（第9.4.4.4节）。
3) 使用**ECI** IP套接字的TCP/IP通信（第9.4.4.5节）。
4) 使用**ECI主机**HTTP服务的HTTP(S)通信（第9.4.4.6节）。

9.4.4.2 基本规范

具有IP连接能力的**ECI主机**应实现包括IPv6 [IETF RFC 8200]在内的IP协议[IETF RFC 791]及其适用的更新。它应提供一种方法，根据[IETF RFC 1034]、[IETF RFC 1035]及其适用的更新，使用DNS完成从**ECI主机**名到IP地址的解析。

为了提供简单的不可靠短消息协议，**ECI主机**应依据[IETF RFC 768]支持经由IP的UDP，包括适用的更新。为了通过流量控制提供可靠的面向连接的消息交换，**ECI主机**应依据[IETF RFC 793]支持经由IP的TCP以及适用的更新。

ECI主机不必在发送或接收模式下为UDP多播提供支持。

9.4.4.3 ECI IP套接字

9.4.4.3.1 概述

为了使用TCP和IP发送和接收通信，**ECI客户端**可打开一个**ECI** IP套接字。

注 – 术语“套接字”表示与许多操作系统中使用的原始BSD套接字相似。作为一个概念，**ECI** IP套接字是相似的，但具有不同于BSD套接字的特定属性。具体而言，该行为是完全异步的。

ECI IP套接字是IP通信的端点。**ECI客户端**可以通过识别本地端口号和接受输入连接**请求**（作为TCP/IP服务器运行）的意愿来打开套接字。可以关闭套接字，在这种情况下，任何关联的连接或服务器行为都会关闭。对等主机名的IP地址可以使用**ECI主机**的DNS服务来解析。

可用的消息在表9.4.4.3.1-1中列出。

表9.4.4.3.1-1 – IP套接字消息

消息	类型	方向	标签	描述
reqIpSocket	A	C→H	0x0	打开**ECI** IP套接字
reqIpClose	A	C→H	0x1	关闭**ECI** IP套接字
reqIpAddrinfo	A	C→H	0x2	获取（远程）**ECI主机**的地址

这些API的结构类型定义在第9.3节中进行定义。

IP套接字API的类型定义：

```
typedef struct Addrinfo {
   ushortaddressType;        /* IPv4 or IPv6 address*/
   ucharipAddress[16];    /* the IP address itself */
   ushort port;              /* port number - if relevant */
} Addrinfo;
```

字段定义：

addressTyp: ushort	参见表9.4.4.3.4-1，只允许值ProtPrefIPv4或ProtPrefIPv6。该字段将hostAddress的长度定义为4或16字节（参见注释）。
ipAddress:uchar[16]	4或16字节，分别表示IPv4或IPv6地址的字节形式表示（按网络顺序）。IPv4地址应使用该参数的前4个字节。
port: ushort	要连接的套接字的端口号（字段可能未被使用）。
注： **ProtPrefIPv4**或**ProtPrefIPv6**在表9.4.4.3.4-1中定义。	

9.4.4.3.2 reqIpSocket消息

C→HreqIpSocket(uchar **source,** ushort**sourcePort,** ushort **protocol) →**
H→C resIpSocket(uchar**socketId)**

- 该**消息**在本地IP地址和端口上为基于TCP或UDP的通信打开一个套接字。

请求参数定义：

source: uchar	参见表9.4.4.3.2-1：指定要用于本地套接字的ECI主机IP地址（在分配多个IP地址的情况下为首选项）。如果特定IP地址不可识别，则ECI主机应选择一个合适的替代方案。
sourcePort:ushort	本地IP连接端点的端口地址。等于0x0000的值意味着ECI主机应为该套接字分配一个空闲的端口地址。其他低于1024的值是不允许的。
Protocol:ushort	参见表9.4.4.3.2-2：指定用于套接字的协议。对IPv4或IPv6的选择应具体分析。

表9.4.4.3.2-1 – IP源参数

名称	值	描述
IpSourceAny	0x00	**ECI主机**的默认IP地址。
IpSourceWan	0x01	用于WAN（互联网）通信的**ECI主机**IP地址。
IpSourcePriv	0x02	用于专有IP协议信道上专用IP通信的**ECI主机**IP地址。
IpSourceLan	0x03	用于本地网络通信的**ECI主机**IP地址。
RFU	其他	保留以供未来使用。

表9.4.4.3.2-2 – IP协议参数

名称	值	描述
SockProtUdpIPv4	0x0001	使用IPv4的UDP/IP。
SockProtUdpIPv6	0x0002	使用IPv6的UDP/IP。
SockProtUdpIPany	0x0003	使用IPv4或IPv6的UDP/IP。
SockProtTcpClientIpv4	0x0005	使用IPv4的TCP/IP，客户端模式（仅用于启动连接）。
SockProtTcpClientIpv6	0x0006	使用IPv6的TCP/IP，客户端模式（仅用于启动连接）。
SockProtTcpClientIpany	0x0007	使用IPv4或IPv6的TCP/IP，客户端模式（仅用于启动连接）。
SockProtTcpServerIpv4	0x0009	使用IPv4的TCP/IP，服务器模式（用于接受输入连接）。
SockProtTcpServerIpv6	0x000A	使用IPv6的TCP/IP，服务器模式（用于接受输入连接）。
SockProtTcpServerIpany	0x000B	使用IPv4或IPv6的TCP/IP，服务器模式（用于接受输入连接）。
RFU	其他	保留以供未来使用。

响应参数定义：

SocketId：uchar。	打开的套接字的套接字ID。

语义描述：

- 在初始化之后，允许立即停止**响应**，直到成功完成**ECI主机**IP地址初始化。性能数据在[b-ITU-T J Suppl. 7]中提出。

先决条件请求：

1) 不得超过**ECI客户端**允许请求的最大套接字数量。

2) 来源、源端口和协议是有效的参数配置。

后置条件响应：

1) 打开套接字或在**响应**中返回一个错误。

表9.4.4.3.2-3列出了关于打开套接字的错误代码。

表9.4.4.3.2-3 – resIpSocket错误代码

名称	描述
ErrIpSourceProt	参见表9.4.4.7-1。
ErrIpNoSockets	
ErrIpProtNotAvail	
ErrIpPortNotAvail	

9.4.4.3.3 reqIpClose消息

C→HreqIpClose(uchar**socketId)** →
H→C resIpClose(uchar**socketId)**

- 关闭IP套接字和任何关联的连接；所有至和自套接字的未决通信都可能被丢弃。

请求参数定义：

socketId：uchar	要关闭的套接字的ID。

响应参数定义：

socketId：uchar。	已关闭的套接字的ID。

语义描述：

- 该**请求**关闭套接字以及与其关联的任何IP连接。如果适用，它将让**ECI主机**来向任何通信对等端发送适当的断开连接消息。后者的成功完成并不是发送**响应**所必需的。没有关联连接的套接字也将被关闭。

先决条件请求：

1) 套接字存在并处于打开状态。

后置条件响应：

2) 套接字已关闭，不能再用于任何通信（除非在reqIpSocket上重新分配）。

表9.4.4.3.3-1列出了有关关闭套接字的错误代码。

表9.4.4.3.3-1 – resIpClose错误代码

名称	描述
ErrIpSocketNotOpen	参见表9.4.4.7-1。

9.4.4.3.4 reqIpAddrInfo消息

C→HreqIpAddrinfo(uint**hostnameLenth,** char **hostname[],** uchar**protPref)** →
H→CresIpAddrinfo(Addrinfo**ipaddress)**

- 该消息提供IP地址信息，以便使用首选的协议（protPref）寻址**ECI主机**、返回**ECI主机**地址。当需要解析**请求**时，协议将使用**ECI主机**的DNS服务。

请求参数定义：

hostNameLength:uint	名称字段的长度（以字节为单位）。
hostname: char[]	要解析的IP主机的名称；无论是IPv4点记法[IETF RFC 952]、IPv6冒号记法[IETF RFC 8200]还是实际主机名[IETF RFC 1123]。
protPref: uchar	指示表9.4.4.3.4-1中定义的IP协议首选项。

表9.4.4.3.4-1 – IP协议首选参数

名称	值	描述
ProtPrefIpv4	0x1	将返回一个IPv4地址。
ProtPrefIPv6	0x2	将返回一个IPv6地址。
ProtPrefAny	0x3	将返回一个IPv4或一个IPv6地址。
RFU	其他	保留以供未来使用。

响应参数定义：

Ipaddress：Addrinfo	**ECI主机**的IP地址。端口字段未定义。

语义描述：

- 该**请求**使用**ECI主机**DNS服务来把提供的主机名转换为二进制主机地址表示。由于临时缺少DNS服务访问（例如，在**CPE**启动期间），可能会发生延迟；**ECI主机**应确保观察到适当的超时（即**ECI客户端**始终收到**响应**）。

后置条件响应：

1) 解决主机地址或错误。

表9.4.4.3.4-2列出了有关关闭套接字的错误代码。

表9.4.4.3.4-2 – resIpAddrInfo错误代码

名称	描述
ErrIpHostUnknown	参见表9.4.4.7-1。
ErrIpHost	
ErrDnsOffline	

9.4.4.4 ECI UDP/IP

9.4.4.4.1 概述

ECI客户端应使用开放的**ECI UDP/IP**套接字来发送和接收UDP数据报。相关的消息在表9.4.4.4.1-1中定义。

表9.4.4.4.1-1 – UDP/IP套接字消息

消息	类型	方向	标签	描述
reqIpUdpSendMsg	A	C→H	0x3	将消息发送给对等端UDP端口。
reqIpUdpRecvMsg	A	C→H	0x4	从对等端UDP端口接收消息。

9.4.4.4.2 reqIpUdpSendMsg消息

C→H reqIpUdpSendMsg(uchar**socketId,** Addrinfo **peer,** uint**datagramLength,** byte **datagram[])** →
H→CresIpUpdSendMsg(uchar**socketId)**

- 该消息将UDP数据报发送给对等端（IP地址，IP端口）。

请求参数定义：

socketId:uchar	名称字段的长度（以字节为单位）。
peer: Addrinfo	数据报的对等端（IP地址，IP端口号）目的地。
datagramLength: uint	数据报的长度（以字节为单位）。
datagram:字节[]	数据报内容（按网络顺序的字节数）。

响应参数定义：

socketId:uchar	在其上发出匹配**请求**的套接字。

语义描述：

- 使用UDP协议以及套接字的IP主机地址和端口将数据报发送给对等端。

先决条件请求：

1) 已经使用与对等端地址相同的地址结构，为UDP打开套接字。

后置条件：

2) 数据报被发送（但可能会丢失）。

表9.4.4.4.2-1列出了有关发送UDP数据报的错误代码。

表9.4.4.4.2-1 – resIpUdpSendMsg错误代码

名称	描述
ErrIpUdpProtMismatch	参见表9.4.4.7-1。
ErrIpUdpSocketNot	
ErrIpUdpTooLong	
ErrIpUdpIpOffline	

9.4.4.4.3 reqIpUdpRecvMsg消息

C→HreqIpUdpRecvMsg(uchar**socketId)** →
H→C resIpUdpRecvMsg(uchar**socketId,** Addrinfo **peer,** uint**datagramLength,** byte **datagram[])**

- 该消息允许**ECI客户端**请求**ECI主机**从对等端接收UDP数据报（即主机名，端口），发送到具有**SocketId**的套接字。

请求参数定义：

socketId:uchar	预期将接收到UDP数据报的套接字（指的是端口号和主机地址）。

响应参数定义：

socketId:uchar	名称字段的长度（以字节为单位）。
peer: Addrinfo	数据报来源（对等端）的IP地址+端口号。
datagramLength: uint	数据报的长度（以字节为单位）。
datagram: 字节[]	数据报内容（按网络顺序的字节数）。

语义描述：

- 可以在套接字上接收一个数据报，在这种情况下返回一个响应。

注1 – 套接字关闭将终止任何未决的**reqIpUdpRecvMsg请求**。

注2 – 允许在同一个套接字上接收相应**响应**之前发出多个reqIpUdpRecvMsg，但**ECI主机**没有义务支持超过5个这样**请求**的排队。

先决条件请求：

- 套接字已经为UDP打开。

后置条件响应：

- 数据报被发送（但可能被丢弃）。

表9.4.4.4.3-1列出了有关接收UDP数据报的错误代码。

表9.4.4.4.3-1 – resIpUdpRecvMsg错误代码

名称	描述
ErrIpUdpSocketNot	参见表9.4.4.7-1。

9.4.4.5 ECI TCP/IP

9.4.4.5.1 概述

ECI客户端可以通过在创建套接字时打开的TCP/IP连接来发送和接收消息，从本地**ECI客户端**到远程对等端服务创建有效的无差错双向字节流序列，反之亦然。这允许**ECI客户端**充当一个服务器，以**请求**来自其他方的信道（通常用于LAN应用程序）。消息在表9.4.4.5.1-1中列出。

表9.4.4.5.1-1 – TCP/IP套接字消息

消息	类型	方向	标签	描述
reqIpTcpConnect	A	C→H	0x5	TCP客户端连接到TCP服务器对等端。
reqIpTcpSend	A	C→H	0x6	将数据发送到连接的对等端。
reqIpTcpRecv	A	C→H	0x7	从连接的对等端接收数据。
reqIpTcpAccept	A	C→H	0x8	TCP服务器对等端接受来自TPC客户端对等端的连接

9.4.4.5.2 reqIpTcpConnect消息

C→HreqIpTcpConnect(uchar**socketId,** Addrinfo **peer)** →
H→C resIpTcpConnect(uchar**socketId)**

- 该消息请求**ECI主机**使用套接字协议从打开的TCP套接字打开一个到对等端的连接。

请求参数定义：

socketId：uchar	自其要建立一个TCP连接的套接字（意味着端口号和主机地址）。
peer：Addrinfo	对等端IP地址，连接要打开的IP端口。

响应参数定义：

socketId：uchar	**请求**的套接字的套接字ID。

语义描述：

- 本地主机将尝试打开从本地套接字到对等端的TCP连接（IP地址，IP端口）。

先决条件：

- 已使用与peerAddressType相同的IP地址类型（IPv4或IPv6）为TCP打开套接字。

后置条件：

- 建立TCP连接或返回错误条件。

表9.4.4.5.2-1列出了有关通过TCP和IP的连接的错误代码。

表9.4.4.5.2-1 – resIpTcpConnect错误代码

名称	描述
ErrIpTcpProtMismatch	参见表9.4.4.7-1。
ErrIpTcpSockNot	
ErrIpTcpIpOffline	
ErrIpTcpConnRefused	
ErrIpTcpConnTimeout	

9.4.4.5.3 reqITCTCend消息

C→HreqIpTcpSend(ucha**rsocketId,** bool **more,** uint**dataLen,** byte **data[]) →**
H→CresIpTcpSend(ucha**rsocketId,** uint**actLen)**

- 该消息在TCP连接的套接字上使用TCP发送数据。

请求参数定义：

socketId：uchar	用于向对等方发送数据的套接字（意味着端口号和主机地址）。
更多：布尔	指示如果数据和前面的数据立即被转发到对等端（**更多=假**），或者后续的reqipTcpSend **请求**中有更多的数据（**更多=真**）。
dataLen：uint	要发送的数据量。
数据：字节[]	要发送的数据。

响应参数定义：

socketId：uchar	在其上签署发送的套接字的套接字ID。
actLen：uint	成功发送的实际字节数。

语义描述：

- 本地主机将通过一个带有socketID、至所连接对等设备的已连接TCP/IP套接字，将**数据**发送给对等设备。

先决条件请求：

1) 套接字处于已连接TCP/IP模式。

后置条件响应：

2) 在actLen不等于dataLen的情况下，一个错误条件将成立。

表9.4.4.5.3-1列出了有关发送TCP分组的错误代码。

表9.4.4.5.3-1 – resIpTcpSend错误代码

名称	描述
ErrIpTcpSockNot	参见表9.4.4.7-1。
ErrIpTcpIpOffline	
ErrIpTcpClosed	
ErrIpTcpConnTimeout	

9.4.4.5.4 reqIpTCPRecv消息

C→HreqIpTcpRecv(uchar**socketId,** uint**maxDataLen)** →
H→C resIpTcpRecv(uchar**socketId,** uint**dataLength,** byte **data[])**

- 该消息在TCP连接套接字上使用TCP来接收数据。

请求参数定义：

socketId：uchar	用于接收至对等端的数据的套接字（意味着端口号和主机地址）。
maxDataLen：uint	接收的最大数据量。

响应参数定义：

socketId：uchar	在其上发送接收消息的套接字的套接字ID。
dataLength：uint	从对等端收到的数据的字节数。
数据：字节[]	从对等端接收的数据。

语义描述：

本地主机通过与**socketID**连接的TCP/IP套接字从对等端接收**数据**。

先决条件请求：

1) 套接字是一个TCP套接字。

后置条件响应：

2) 最大长度的所有可用数据将返回到**请求**中的**maxDataLen**字段。如果没有任何数据可用，则**响应**将停止，直至连接关闭，TCP连接被认为暂时不可用或者到IP网络的本地连接丢失。

表9.4.4.5.4-1列出了有关接收TCP分组的错误代码。

表9.4.4.5.4-1 – resIpTcpRecv错误代码

名称	描述
ErrIpTcpSockNot	参见表9.4.4.7-1。
ErrIpTcpIpOffline	
ErrIpTcpClosed	
ErrIpTcpConnTimeout	

9.4.4.5.5 reqIpTCPAccept消息

C→HreqIpTcpAccept(uchar**socketId)** →
H→CresIpTcpAccept(uchar**socketId,** uchar**newSocketId,** Addrinfo **peer)**

- 该消息接受TCP服务器套接字上的输入连接**请求**。未决的连接**请求**将得到处置；具有**ECI主机**实施方案定义的最大值。TCP服务器的性能要求在[b-ITU-T J Suppl. 7]中提出。

请求参数定义：

socketId：uchar	用于接收连接**请求**的套接字（意味着端口号和主机地址）。

消息字段定义：

socketId：uchar	发出请求的套接字的套接字标识。
newSocketId：uchar	新近打开的、与对等端连接的套接字标识，它发出一个连接**请求**。主机地址和端口继承自**socketId**套接字。
peer：Addrinfo	连接对等端的IP地址+ IP端口。

语义描述：

- 本地**ECI主机**等待在套接字创建中指定的IP地址/端口上的输入TCP连接**请求**，并打开一个新连接的套接字，处理输入（或未决）的连接**请求**。如果没有任何输入**请求**，或者服务器套接字被关闭，则无**响应**。

先决条件请求：

1) 套接字是一个TCP服务器套接字。

后置条件响应：

2) 应任何可用的连接**请求**，返回一个具有打开的TCP/IP连接的新套接字给服务器套接字或者产生错误。

表9.4.4.5.5-1列出了有关接受TCP连接的错误代码。

表9.4.4.5.5-1 – resIpTcpAccept错误代码

名称	描述
ErrIpTcpListSockNot	参见表9.4.4.7-1。
ErrIpTcpNoMoreSockets	

9.4.4.6 用于HTTP(S)的API获取服务

9.4.4.6.1 概述

ECI主机应提供基本的HTTP(S)GET请求，以代表客户端从基于IP的HTTP服务器检索资源。这允许**ECI客户端**从互联网服务器检索基于万维网的资源（文件）。HTTPS可用于检索

第9.7.2节和第7.8.4.2节中定义的、基于万维网API的资源，如入口或出口数据。

安全性由底层CPE的TLS实施方案的HTTPS（TLS）来提供。

注 – 该安全性一般不应用于确保**ECI客户端**的内容保护完整性，但可用于确保操纵**ECI客户端**的DDOS和其他机会性尝试受到阻碍。

ECI主机应支持具有最少量资源的**ECI客户端**，以发布HTTP Get请求。值在[b-ITU-T J Suppl. 7]中提出。

表9.4.4.6.1-1列出了有关HTTP(S)获取API的API消息。

表9.4.4.6.1-1：HTTP Get API消息

消息	类型	方向	标签	描述
reqHttpGetFile	A	C→H	0x0	对URL执行一个HTTP获取请求，并将结果存储在文件中。
reqHttpGetData	A	C→H	0x1	对URL执行一个HTTP获取请求，并将结果作为数据发送给客户端。

9.4.4.6.2 适用规范

注 – 以下规范非第9.4.4.6.1节中所述的ECI安全的必要部分。

用于实现**ECI客户端**API的HTTP和HTTPS协议实施方案应符合HTTP1.1 [IETF RFC 7230]和[IETF RFC 7231]的要求。

用于向**ECI客户端**提供HTTP服务的传输层安全（TLS）实施方案应符合TLS 1.3 [IETF RFC 8446]。为了实现向后兼容性，应根据TLS1.3约束和以下规则支持TLS1.2：

1) TLS 1.2，参见[IETF RFC 5246]。
2) TLS AES-GCM，参见[IETF RFC 5288]。
3) TLS扩展，参见[IETF RFC 6066]。
4) PKIX/X.509 [IETF RFC 5280] +更新[IETF RFC 6818]。

所有TLS1.2实施方案都应支持[IETF RFC 5246]中定义的以下密码套件：

1) TLS_RSA_WITH_AES_128_CBC_SHA256。
2) TLS_DHE_RSA_WITH_AES_128_GCM_SHA256。

依据TLS1.3约束，可以支持TLS1.2的其他密码套件。

TLS1.2密码套件的选择具有以下规则：

1) TLS_DHE_RSA_WITH_AES_128_GCM_SHA256应该是默认的密码套件。
2) 应优先考虑AEAD密码套件。
3) 应该优先考虑基于DHE的密钥交换。
4) 不应优先考虑超过128位的密钥。
5) 不应使用3DES。
6) 不应使用RC4（如[W3C PNG]中所述）。
7) 不应使用MD5（如[IETF RFC 6151]中所述）。

以下处理规则适用：

1) TLS 1.2将是所有**ECI**实体要求的最低版本。

2) 不得使用SSL 2.0和3.0。

3) 不得使用重新谈判。

4) 不应使用压缩（GCM可接受）。

5) DH/DHE的素数应至少为1 024位，并在TLS握手期间进行验证。

6) **证书**和主机的验证应符合PKIX要求[IETF RFC 5280]和[IETF RFC 6125]。

用于验证TLS连接副本的根证书应基于最新列表，例如：https://cabforum.org/browser-os-info/。

CPE应支持**CPE制造商**在制造后可以删除或不信任根证书的方式。这可以通过固件升级机制或最好通过特定的根证书更新机制来处理，该机制可以允许更及时的更新。**CPE制造商**可以选择删除或不信任CPE中的强制性根证书，以应对安全威胁。**CPE**应该支持一种在制造后安全添加新的根证书的方法，以便随着时间的推移与服务器保持互操作性。

有关实施方案的其他导则可以在CA /浏览器论坛[b-CA浏览器]和[b-NIST SP 800-52r2]概述的处理规则中找到。

注 – 为确保互操作性，旨在利用基于HTTP的服务为**ECI客户端**提供支持的HTTP服务器，应该支持兼容的模式和选项以及此处为HTTP客户端定义的适用建议。

9.4.4.6.3 reqHttpGetFile和reqHttpGetData消息

C→H **reqHttpGetFile(**filename **fname** ;char **url[]**, char **userAgent[]**; uint**redirs**, uint**timeout**) →
H→C **resHttpGetFile**(uint**httpStatus**)

C→H **reqHttpGetData(**char **url[]**, **userAgent[]**; uint**redirs**, uint**timeout**) →
H→C **resHttpGetData**(uint**httpStatus,** byte **data[]**)

- 该消息请求**ECI主机**执行HTTP请求，以检索文件并在完成时返回HTTP状态。
- resHttpGetFile将资源作为客户端文件系统中的一个文件进行返回。
- resHttpGetData将资源作为大小有限的消息数据进行返回。

请求参数定义：

fname:fileName	文件的文件名，当中**ECI主机**存储请求的结果（发布数据）。任何现有的数据都会被覆盖。
url: char[]	UTF-8编码[IETF RFC 7230]的URL。非标准端口号可被指定为URL的一部分。TLS应用于符合[IETF RFC 7230]中“https URI方案”的URL。
userAgent: char[]	指定要用作HTTP报头的用户代理报头字段。**ECI客户端**可指定**url**之HTTP服务器预期的特定值（参见注释）。
redirs: unit	允许完成请求的最大重定向数量。在[b-ITU-T J Suppl. 7]中提出了**redirs**的最低性能数据。
timeout: unit	HTTP请求完成超时（以毫秒为单位）。如果发生超时，则请求将被中止，将在**响应**中返回一个超时错误。
注： 不建议将用户代理用作资源的访问控制或选择机制，并遵循[IETF RFC 7231]中定义的预期用途。	

响应参数定义：

httpStatus:uint	HTTP状态的值。
data: 字节[]	网络顺序中HTTP GET结果的数据。最大大小受消息缓冲区大小的限制。

详细的语义：

- **ECI主机**应确保HTTP请求支持广泛的通用文件和媒质类型。建议不要在HTTP请求报头中包含接受（Accept）报头字段。如果添加了接受（Accept）报头，则下列内容编码MIME类型应可用于检索资源：application/octet-stream、application/json、image/jpeg、image/png、image/gif、text/plain、text/html、text/css、text/xml和text/javascript。
- **ECI主机**应确保HTTP请求报头接受编码（Accept-Encoding）发出以下内容编码是可接受的信号：gzip。

后置条件响应：

1) 检索**URL**中的资源并将之存储在文件名**fname**中（用于**resHttpGetFile**）或作为数据返回（用于**rerHttpGetData**）或发生一个错误。

表9.4.4.6.3-1列出了resHttpGetFile和resHttpGetData相关的错误代码。

表9.4.4.6.3-1 – resHttpGetFile和resHttpGetData错误代码

名称	描述
ErrHttpGetNoSockets	参见表9.4.4.6.4-1。
ErrHttpGetProtNotAvail	
ErrHttpGetPortNotAvail	
ErrHttpHostUnknown	
ErrHttpDnsOffline	
ErrHttpIpOffline	
ErrHttpTimeout	
ErrHttpGetFSFailure	
ErrHttpGetFSExceeded	
ErrHttpGetTlsAuth	
ErrHttpGetRedir	
ErrHttpGetData	

9.4.4.6.4 HTTP Get API的错误代码

这些值指的是API特定的错误，它们可由该API的**响应**消息返回，列于表9.4.4.6.4-1中。

表9.4.4.6.4-1 – HTTP Get API的错误代码

名称	值	描述
ErrHttpGetNoSockets	-257	参见表9.4.4.7-1中关于IP套接字API的错误代码的相应值。
ErrHttpGetProtNotAvail	-258	
ErrHttpGetPortNotAvail	-259	
ErrHttpHostUnknown	-261	
ErrHttpDnsOffline	-263	
ErrHttpIpOffline	-267	
ErrHttpTimeout	-270	HTTP请求无法在请求中设置的超时内完成。
ErrHttpGetFSFailure	-512	值+ 256对应于表9.4.5.5-1中文件系统API的错误代码值。
ErrHttpGetFSExceeded	-514	
ErrHttpGetTlsAuth	-768	服务器或数据无法成功通过TLS协议的验证。
ErrHttpGetRedir	-784	超过重定向的数量。
ErrHttpError	-785	无法从服务器检索资源；HTTP错误代码指明原因。
ErrHttpGetData	-786	资源数据超出最大长度数据字段。

9.4.4.7 IP套接字API的错误代码

API特定错误的值可由该API的**响应**消息返回，列于表9.4.4.7-1中。

表9.4.4.7-1 – IP套接字API的错误代码

名称	值	描述
ErrIpSourceProt	-256	来源和协议的组合无效。
ErrIpNoSockets	-257	没有更多的套接字可用。
ErrIpProtNotAvail	-258	协议不可用。
ErrIpPortNotAvail	-259	请求的端口不可用。
ErrIpSocketNotOpen	-260	套接字未打开。
ErrIpHostUnknown	-261	**ECI主机**未知。
ErrIpHost	-262	**ECI主机**已知，但没有可用的地址（对于指定的IP地址类型）。
ErrDnsOffline	-263	DNS服务可能暂时处于离线状态。
ErrIpUdpProtMismatch	-264	对等端地址与套接字协议不匹配。
ErrIpUdpSockNot	-265	套接字不是UDP套接字。
ErrIpUdpTooLong	-266	数据报，以延长单个UDP消息。
ErrIpUdpIpOffline	-267	IP连接离线（无法访问对等端）。
ErrIpTcpProtMismatch	-268	对等端地址与套接字协议不匹配。
ErrIpTcpSockNot	-269	套接字不是TCP套接字。
ErrIpTcpIpOffline	-258	目前没有本地IP互联网连接。
ErrIpTcpConnRefused	-259	连接不被此端口上的对等端主机接受。
ErrIpTcpConnTimeout	-260	无法自对等端**ECI主机**获得**响应**。
ErrIpTcpClosed	-261	TCP连接不可用或不再可用。
ErrIpTcpListSockNot	-262	套接字不是TCP服务器套接字。
ErrIpTcpNoMoreSockets	-263	输入连接**请求**已收到，但主机不在套接字中。
RFU	其他	保留以供未来使用。

9.4.5 用于访问文件系统的API

9.4.5.1 引言

ECI客户端可以访问私有文件系统以存储有限数量的数据，这些数据在正常运行条件下可以承受**ECI客户端**生命周期、**CPE**电源周期、系统崩溃等。可靠性应至少与普通的**CPE**文

件系统相同；即在可能导致用户不适的某些特殊情况下可能发生故障。管理**ECI客户端**的安全系统确保不会对**用户**发生对内容访问权利的过度丢失。文件系统不安全。不可能由指定的**ECI客户端**及其支持的**ECI主机**之外的其他实体（例如不妥协的**CPE**和**ECI主机**）来操控。

文件系统抽象指的是单个平面目录的抽象。基本目录服务可用。文件系统访问函数类似于Unix/Linux/Posix文件系统调用，如open、close、write、read、lseek、opendir、readdir和lstat。

如果**用户**存储了最小数量的文件系统存储器，则每个**ECI客户端**都可以使用该文件系统存储器。该数量在[b-ITU-T J Suppl. 7]中提出。

文件系统API分为三个子部分：

1) 文件打开和关闭。

2) 读取和写入文件，随机访问并从文件中删除一个选定的数据。

3) 目录服务。

文件名应由8位ASCII字符序列组成，最少1位、最多8位以下字符（逗号分隔）：A-Z，a-z，0-9，_，并应以NULL字符终结。文件名定义参见表9.4.5.1-1。

表9.4.5.1-1 – FileName结构

```
typedef char fileName[9];
```

日志文件提供的功能允许**ECI客户端**以缓冲方式写入有限数量的数据；即没有停止执行。每个**ECI客户端**的日志文件数量在xxx中定义（每个客户端最少2个）。这使得这些文件适用于应用程序级别的日志记录、跟踪和事后分析。

9.4.5.2 文件打开和关闭

9.4.5.2.1 概述

ECI客户端可以打开一个文件进行读取与/或写入，它传送一个文件句柄，通过它可以执行后续的读写访问。如果文件不存在，则可以创建它。该文件有一个属性“文件位置”，它指向用于访问文件的当前位置。

文件句柄应由**ECI主机**进行管理。关闭的文件句柄不得在之后立即重用，以确保**ECI客户端**对文件的非同步访问不会导致访问错误的文件。

表9.4.5.2.1-1列出了文件打开和文件关闭消息。

表9.4.5.2.1-1：文件打开和关闭消息

消息	类型	方向	标签	描述
reqFileOpen	A	C→H	0x0	打开**ECI客户端**私有文件。
reqFileClose	A	C→H	0x1	关闭打开的文件。

9.4.5.2.2 reqFileOpen消息

C→H reqFileOpen(fileName**fname,** uint**fileOpenOptions**)→
H→CresFileOpen(uchar**fileHandle)**

- 该消息允许**ECI客户端**请求**ECI主机**打开一个具有特定访问权限的文件。

请求参数定义：

fname: filename	要打开的文件的名称。
fileOpenOptions: unit	打开文件的访问模式。在表9.4.5.2.2-1中定义了允许的值及其含义。

表9.4.5.2.2-1 – 文件打开选项

名称	位	值	描述
FileRead	0,1	0b00	为读取而打开文件。文件位置设置在文件的开头。
FileWriteAppend	0,1	0b01	为写入而打开文件；随后的写入会添加到现有的文件。文件位置设置在文件的末尾。
FileWriteOver	0,1	b11	为在任何位置写入而打开文件。文件位置设置在文件的末尾。
未用	0,1	0b10	不允许。
LogFileNo	2	0b0	普通文件。
LogFileYes	2	0b1	允许同步写入的特殊日志文件。
位32-2		其他	保留以供未来使用。

响应参数定义：

fileHandle：uchar	引用（句柄）打开的文件。

后置条件请求：

1) 以所需的访问模式打开文件或返回错误。错误代码列在表9.4.5.2.2-2中。

表9.4.5.2.2-2 – resfileOpen错误代码

名称	描述
ErrFileNameNotExist	参见表9.4.5.5-1。
ErrFileQuotaExceeded	
ErrFileSystemFailure	

9.4.5.2.3 reqFileClose消息

C→H reqFileClose(uchar**fileHandle) →**
H→CresFileClose()

- 该消息关闭对利用**fileHandle**打开之文件的访问。在表9.4.5.2.3-1中列出了有关关闭文件的错误代码。

请求参数定义：

fileHandle：uchar	要关闭文件的句柄。

先决条件请求：

1) fileHandle处于打开状态。

后置条件请求：

1) 对fileHandle的随后访问将失败，并显示ErrFileNotOpen。
2) 将提交任何未决的写操作（除非发生错误）。

表9.4.5.2.3-1 – resfileClose错误代码

名称	描述
ErrFileHandleNotExist	参见表9.4.5.5-1。
ErrFileSystemFailure	

9.4.5.3 文件访问

9.4.5.3.1 概述

文件访问消息允许通过一个文件句柄来读取和写入要访问的文件，并重新定位文件中的当前位置，以读取/写入。定义的原语与Linux/Unix约定有直接的对应关系。定义的消息在表9.4.5.3.1-1中列出。

注 – reqFileWrite和reqFileRead与reqTcpSend和reqTcpRecv具有很强的相似性。

表9.4.5.3.1-1 – 文件访问消息

消息	类型	方向	标签	描述
reqFileWrite	A	C→H	0x2	从当前文件位置开始连续写入字节。
reqFileRead	A	C→H	0x3	从当前文件位置开始读取连续的字节。
reqFileSeek	A	C→H	0x4	重新定位当前文件位置。
reqFileRemoveData	A	C→H	0x5	从当前位置的一个文件中删除数据。
callFileDataLog	S	C→H	0x6	在缓冲文件末尾添加数据。

9.4.5.3.2 reqFileWrite消息

C→H reqFileWrite(uchar**fileHandle,** bool **sync,** uint**dataLen,** byte **data[]) →**
H→C resFileWrite(uchar**fileHandle)**

- 该消息将dataLen字节写入从当前文件位置开始的文件中。

请求参数定义：

fileHandle:uchar	要写入的文件的句柄。
sync: bool	如果真，则写入**响应**确保文件系统的状态与此以及所有前面的写入保持同步。如果为假，则**ECI主机**可以缓冲写入**请求**（在系统故障时仍可能丢失）。
dataLen: uint	要写入文件的字节数。
data: byte[]	要写入文件的数据。

响应参数定义：

fileHandle：uchar	写入的文件的句柄。

先决条件请求：

1) 文件以写入模式打开（FileWriteOver或FileWriteAppend模式）。
2) 文件位置可以写入：如果文件以FileWriteAppend模式打开，则文件位置应位于末尾。
3) 要写入的数据量不会导致文件系统配额问题。

后置条件请求：

1) 除非发生错误，否则文件状态将被更新，并且文件位置将从当前（等待文件上的其他缓冲操作）前置到+ dataLen。

2) 在成功写入和同步的情况下，数据在**ECI主机**文件系统中以NV状态提交。

错误代码列在表9.4.5.3.2-1中。

表9.4.5.3.2-1 – resFileWrite错误代码

名称	描述
ErrFileHandleNotExist	参见表9.4.5.5-1。
ErrFileQuotaExceeded	
ErrFileSystemFailure	
ErrFileWriteNot	

9.4.5.3.3 reqFileRead消息

C→H reqFileRead(uchar**fileHandle**, uint**dataLen) →**
H→C resFileRead(uchar**fileHandle**, uint**dataRead,** byte **data[])**

- 该消息从当前文件位置开始读取文件中的最大dataLen字节。表9.4.5.3.3-1列出了有关从文件中读取数据的错误代码。

请求参数定义：

fileHandle：uchar	要读取的文件的句柄。
dataLen：uint	最大读取字节数。

响应参数定义：

fileHandle：uchar	从中读取的文件的句柄。
dataRead：uint	读取并存储在**数据**中的字节数。
数据：字节[]	读取的数据。

先决条件请求：

1) 打开文件。

后置条件请求：

1) 出现一个错误；或者

2) 从文件中读取从最后一个文件位置开始的文件中的**dataLen**最小或剩余字节数；以及

3) 文件位置已通过**dataRead**递增；

4) 除非发生错误，否则文件位置将由**dataLen**前置或者位于文件末尾。

表9.4.5.3.3-1 – resFileReade错误代码

名称	描述
ErrFileHandleNotExist	参见表9.4.5.5-1。
ErrFileSystemFailure	

9.4.5.3.4 reqFileSeek消息

C→H reqFileSeek(uchar**fileHandle,** int **offset,** uchar**seekPos) →**

H→C resFileSeek(uchar**fileHandle,** int**remOffset)**

- 该消息将指针定位在打开文件中的某个位置上，并返回部分文件内容。

请求参数定义：

fileHandle:uchar	文件位置将要更改的文件的句柄。
offset: int	由**seekPos**指定的查找参考位置的偏移量，文件位置应假定。
seekPos: uchar	参见表9.4.5.3.4-1。

表9.4.5.3.4-1 – 文件搜索参考位置

名称	值	描述
FileSeekSet	0x00	文件参考位置位于文件的开头。
FileSeekCur	0x01	文件参考位置在当前文件位置。
FileSeekEnd	0x02	文件参考位置位于文件末尾。
RFU	其他	保留以供未来使用。

响应参数定义：

fileHandle:uchar	文件位置已发生变化的文件的句柄。
remOffset: int	指定的偏移量与设定文件位置的偏移量之间的差异。

详细的语义：

- 文件位置被重新定位，并在**请求**的参数描述中予以定义。文件位置永远不会超出文件末尾或置于文件开始之前。请求的偏移量与文件引用位置的实际偏移量之差在**remOffset**结果参数中予以返回。错误代码列在表9.4.5.3.4-2中。

先决条件请求：

1) 文件被打开。

后置条件请求：

1) 出现一个错误；或者
2) 文件位置将按照上面的定义进行设置；以及
3) **remOffset**将反映偏移量与实际文件位置之间的差异，如上定义。

表9.4.5.3.4-2 – resFileReade错误代码

名称	描述
ErrFileHandleNotExist	参见表9.4.5.5-1。
ErrFileSystemFailure	

9.4.5.3.5 reqFileRemoveData消息

C→H reqFileRemoveData(uchar**fileHandle,** bool **sync,** uint**dataLen) →**
H→C resFileRemoveData(uchar**fileHandle)**

- 该消息从当前文件位置开始的文件中删除dataLen个字节。

请求参数定义：

fileHandle：uchar	文件句柄。
sync：布尔	如果真，则写入**响应**确保文件系统的状态与此以及所有前面的写入保持同步。如果为假，则**ECI主机**可以缓冲写入**请求**（在系统故障时它仍可能丢失）。
dataLen：uint	要从文件中删除的字节数。如果这超过了文件的末尾，则只有到文件结尾的字节被删除。

响应参数定义：

fileHandle：uchar	写入的文件的句柄。

先决条件请求：

1) 文件以写入模式打开（FileWriteOver模式）。

后置条件请求：

1) 文件状态将被更新。文件位置将保持不变。
2) 在成功删除和**同步**的情况下，数据将在**ECI主机**文件系统中以NV状态提交。

错误代码列在表9.4.5.3.5-1中。

表9.4.5.3.5-1 – resFileWrite错误代码

名称	描述
ErrFileHandleNotExist	参见表9.4.5.5-1。
ErrFileSystemFailure	
ErrFileWriteNot	

9.4.5.3.6 callFileDataLog消息

C→H callFileDataLog(uchar**fileHandle,** uint**dataLen**, byte **data[])**

- 该消息使用系统缓冲区将dataLen个字节（数据）附加到该文件的末尾。

调用参数定义：

fileHandle：uchar	文件句柄。
dataLen：uint	要附加到日志文件的字节数。
data[]：byte	要写入的数据。

先决条件调用：

1) 文件以写入模式打开（FileWriteOver或FileWriteAppend模式）。
2) 在文件的末尾设置文件位置。
3) 要写入的数据量不会导致文件系统配额问题。

后置条件调用：

1) 除非出现错误，否则文件状态会更新并且文件位置从当前值提前到当前值+dataLen。
2) 除非出现系统故障，否则结果将被提交给**ECI主机**文件系统。

详细的语义：

1) **ECI主机**应缓存数据，并尽快将之添加至文件末尾。
2) 在[b-ITU-T J Suppl. 7]中提出了用于此目的的日志的最大缓冲空间。

错误代码列在表9.4.5.3.6-1中。

表9.4.5.3.6-1 – resFileLog错误代码

名称	描述
ErrFileHandleNotExist	定义参见表9.4.5.5-1。
ErrFileQuotaExceeded	
ErrFileSystemFailure	
ErrFileWriteNot	
ErrFileLogNot	

9.4.5.4 目录服务

9.4.5.4.1 概述

目录服务提供扫描可用**ECI客户端**文件的功能。文件通过其唯一的名称来表征，有大小，有上次修改时间的属性。可用消息在表9.4.5.4.1-1中列出。

注 – 时间属性具有与文件系统和文件内容本身相同的完整程度。

表9.4.5.4.1-1 – 文件目录服务消息

消息	类型	方向	标签	描述
reqFileStat	A	C→H	0x07	返回文件的大小和修改时间。
reqFileCreate	A	C→H	0x08	创建一个新文件。
reqFileDelete	A	C→H	0x09	删除一个文件。
reqFileDir	A	C→H	0x0A	列出**ECI客户端**文件系统中可用文件的文件名。

9.4.5.4.2 reqFileStat消息

C→HreqFileStat(fileName **filename)** →
H→CresFileStat(uint **size;** long**mtime)**

- 该消息允许**ECI客户端**请求**ECI主机**检索存储文件的文件大小和上次修改时间。

请求参数定义：

filename: filename	将检索属性的文件的名称。

响应参数定义：

size: uint	文件大小（以字节为单位）。
mtime: long	上次同步文件修改的时钟时间。

先决条件请求：

1) 文件名是文件系统中的一个现有文件。

后置条件请求：

1) **size**和**mtime**反映名称为**filename**或出现错误的文件的属性。

错误代码列在表9.4.5.4.2-1中。

表9.4.5.4.2-1 – resFileStat错误代码

名称	描述
ErrFileNameNotExist	参见表9.4.5.5-1。
ErrFileSystemFailure	

9.4.5.4.3 reqFileCreate消息

C→HreqFileCreate(fileName **filename) →**
H→CresFileCreate()

- 该消息允许**ECI客户端**请求**ECI主机**创建一个新的空文件。任何具有相同名称的现有文件都将被删除。

请求参数定义：

文件名：文件名	将创建的新空文件的名称。

详细的语义：

- 除非文件系统已损坏，否则创建的文件将在系统故障后存在。

后置条件请求：

1) **ECI客户端**文件系统中存在名称为filename的空文件，修改时间戳设置为当前时间或错误出现时间。

错误代码列在表9.4.5.4.3-1中。

表9.4.5.4.3-1 – resFileCreate错误代码

名称	描述
ErrFileQuotaExceeded	参见表9.4.5.5-1。
ErrFileSystemFailure	

9.4.5.4.4 reqFileDelete消息

C→HreqFileDelete(fileName **filename) →**
H→CresFileDelete()

- 该消息删除名称为文件名的文件。

请求参数定义：

filename:fileName	将创建的新空文件的名称。

详细的语义：

- 除非文件系统已损坏，否则删除的文件在系统故障后不应存在。

后置条件请求：

1) 带**filename**的文件在文件系统中不存在。

错误代码列在表9.4.5.4.4-1中。

表9.4.5.4.4-1：resFileDelete错误代码

名称	描述
ErrFileNameNotExist	参见表9.4.5.5-1。
ErrFileSystemFailure	

9.4.5.4.5 reqFileDir消息

C→HreqFileDir(ushort**maxNr) →**
H→CresFileDir(uint**listLen;** fileName**dirList[])**

- 该消息提供文件名列表，最多有maxNr个项目。对列表顺序未做定义。

请求参数定义：

maxNr：ushort	将被检索的最大数量的文件名。

响应参数定义：

listLen：uint	列表的长度，以字节为单位。
dirList：filename []	**ECI客户端**可用文件的文件名列表。

错误代码列在表9.4.5.4.5-1中。

表9.4.5.4.5-1 – resFileDelete错误代码

名称	描述
ErrFileSystemFailure	参见表9.4.4.7-1。

9.4.5.5 文件系统API的错误代码

这些值指的是API特定的错误，它们可由该API的**响应**消息返回，列于表9.4.5.5-1中。

表9.4.5.5-1：文件系统API错误代码

名称	值	描述
ErrFileSystemFailuret	-256	损坏或卸除的文件系统。
ErrFileNameNotExist	-257	文件系统中不存在文件名。
ErrFileQuotaExceeded	-258	**ECI客户端**的文件系统资源已被超出。
ErrFileNameNotExists	-259	**ECI客户端**的文件系统中不存在文件名。
ErrFileHandleNotExists	-260	文件句柄不存在（可能之前已被关闭）。
ErrFileAppendNot	-261	尝试不在文件末尾写入文件。
RFU	其他	保留以供未来使用。

9.4.6 用于访问时间/时钟资源的API

9.4.6.1 引言

ECI客户端可以通过一个简单的API来访问计时器事件和一天的时间。

时钟的稳健性应通过适合**ECI生态系统**中所有应用的稳健性制度来定义。

- 如果要求**ECI生态系统**在脱机情况下支持文件存储系统的防回滚或与时间相关的权限表达，则时钟应具有稳健性，以便充分保护标有从该时钟衍生之时间戳的本地存储器上的操作被操纵。

计时器允许在未来某个（延迟）时间生成一条消息。计时器事件可以被取消。

注 – 可以创建定期发生计时器事件的、使用时钟和计时器API的组合。

计时器和时钟API分为两部分：

1) 计时器API。
2) 时钟API。

9.4.6.2 计时器API

9.4.6.2.1 概述

计时器API允许**ECI客户端**设置一个计时器，它将在设定的时间发送**响应**。如果需要，**ECI客户端**可以取消该事件。某个时刻未决计时器的数量可通过实施约束条件来限制。[b-ITU-T J Suppl. 7]中提出了**ECI主机**将对每个**ECI客户端**支持的最小未决计时器数量。计时器API的消息在表9.4.6.2.1-1中列出。

表9.4.6.2.1-1 – 计时器API消息

消息	类型	方向	标签	描述
reqTimerEvent	A	C→H	0x0	未来设置一个计时器事件。
reqTimerCancel	A	C→H	0x1	取消先前设置的计时器事件。

9.4.6.2.2 reqTimerEvent消息

C→HreqTimerEvent(uint**timeInterval) →**
H→CresTimerEvent()

- 该消息在未来设置计时器，并在计时器到期时接收**响应**。

请求参数定义：

timeInterval:uint	未来时间以毫秒为单位。

后置条件请求：

- 在timeInterval毫秒后，resTimerEvent将被发送到**ECI客户端**，除非首先收到reqTimerCancel。

先决条件响应：

- 计时器到期，没有收到计时器的reqTimerCancel。

错误代码列在表9.4.6.2.2-1中。

表9.4.6.2.2-1 – resTimerEvent错误代码

名称	描述
ErrTimerMaxExceeded	参见表9.4.6.4-1。

9.4.6.2.3 reqTimerCancel消息

C→HreqTimerCancel(msgId **id) →**
H→CresTimerCancel()

- 该消息取消原始**请求**的每个消息标识符的先前设置计时器。

请求参数定义：

id:msgId	取消使用带有消息ID的异步消息设置的计时器。

先决条件请求：

1) ID因reqTimerEvent而返回，计时器尚未过期。

后置条件响应：

2) 计时器被取消 - 不会发送resTimerCancel - 或者返回一个错误。

3) 如果计时器被取消，但在**resTimerCancel**之前收到**resTimerEvent**，则会出现TimerExpired错误。

9.4.6.3 时钟API

9.4.6.3.1 概述

时钟API允许**ECI客户端**以整数形式读取时钟，并将其转换为本地时间表示。时钟API消息在表9.4.6.3.1-1中列出。

表9.4.6.3.1-1 – 时钟API消息

消息	类型	方向	标签	描述
getTime	S	C→H	0x3	以整数值形式读取本地系统时钟。
callLocaltime	S	C→H	0x4	将时间整数值转换为本地时间。

9.4.6.3.2 getTime消息

C→H long**getTime()**

- 该消息返回从1970年1月1日格林尼治标准时间0:00开始的时间，以秒为单位。

9.4.6.3.3 callLocaltime消息

C→HcallLocaltime(long **time;** tm***tim)**

- 该消息将**时间**转换成人类表示方式，并在结构中定义**tim**。类似于<time.h>中的c-library函数localtime。

调用参数定义：

time: long	将从1970年1月1日格林威治标准时间0:00开始（以秒为单位）、以整数表示的时间，转换为当地时间。
tim: tm *	指向tm结构的指针，它将被设置为当地时间。**tm**在表9.4.6.3.3-1中定义。

表9.4.6.3.3-1 – 人类时间表示结构tm的类型定义

```
typedef struct tm {
   int tm_sec;  // 0 .. 59 (seconds) or 60 in case of a leap second
   int tm_min;  // 0 .. 59 (minutes)
   int tm_hour     // 0 .. 23 (hours)
   int tm_mday; // 1 .. 31 (day of month)
   int tm_mon;     // 1 .. 12 (month)
   int tm_year; // year - 1900
   int tm_wday; // 0 .. 6 (day of week; 0=Sunday)
   int tm_yday; // 0 .. 365 (day of year, 0= 1jan)
   int tm_isdst;   // 1=daylight saving in effect, 0=no daylight saving
   char tm_zone[15]; // string for time zone: e.g. GMT, CET
   int tm_gmtoff;  // local timeoffset from GMT
} tm ;
```

9.4.6.4 时间和时钟API的错误代码

API特定错误的值可由该API的**响应**消息返回，列于表9.4.6.4-1中。

表9.4.6.4-1 – 时间和时钟API错误代码

名称	值	描述
ErrTimerMaxExceeded	256	最大计时器时间超出。
RFU	其他	保留以供未来使用。

9.4.7 用于访问功率管理的API

9.4.7.1 引言

ECI客户端可以访问**ECI主机**的电源管理接口。该接口允许**ECI客户端**在系统待机事件中进行简单的断电或协商断电，并允许**ECI客户端**稍后从电源待机状态重新启动**CPE**和**ECI客户端**，以执行后台功能。**ECI主机**具有以下电源状态：

- **PwrOn**：**ECI主机**功能正常，无意关机。
- **PwrToStby**：**ECI主机**打算进入待机状态（但可以返回到PowerOn状态）。通常要求所有的**ECI客户端**关闭电源。
- 待机：**ECI主机**和**ECI客户端**状态不起作用。**CPE**（并因此而**ECI主机**和**ECI客户端**）可以在预先安排的事件（通常是计时器）中自该状态唤醒。
- 关机：**CPE**没有电源。**ECI主机**和**ECI客户端**未处于功能状态。

ECI客户端可以采用简单的电源管理模式，只要**ECI主机**认为适于关闭即可关闭。或者，**ECI客户端**可以通过发送**reqPwrInfo（PwrInfoOn）**消息来请求进入受控模式。在这种模式下，将通知之，**ECI主机**打算使用**reqPwrChange**消息进行关机，**ECI客户端**可以通过**resPwrChange（PwrDown）**来确认该消息，或者推迟一个适当的参数以**resPwrChange（PwrUp）**，直至完成并准备进入待机状态。**ECI主机**应定期重申**reqPwrChange**消息。

注 – 不能完全保证**ECI客户端**可以始终完成所有活动（例如，在不受控制的电源故障或延长进入待机状态的情况下）。

图9.4.7.1-1给出了带状态转换条件和动作/消息的**ECI主机**状态，它们将在向受控模式下的**ECI客户端**转换时被触发。

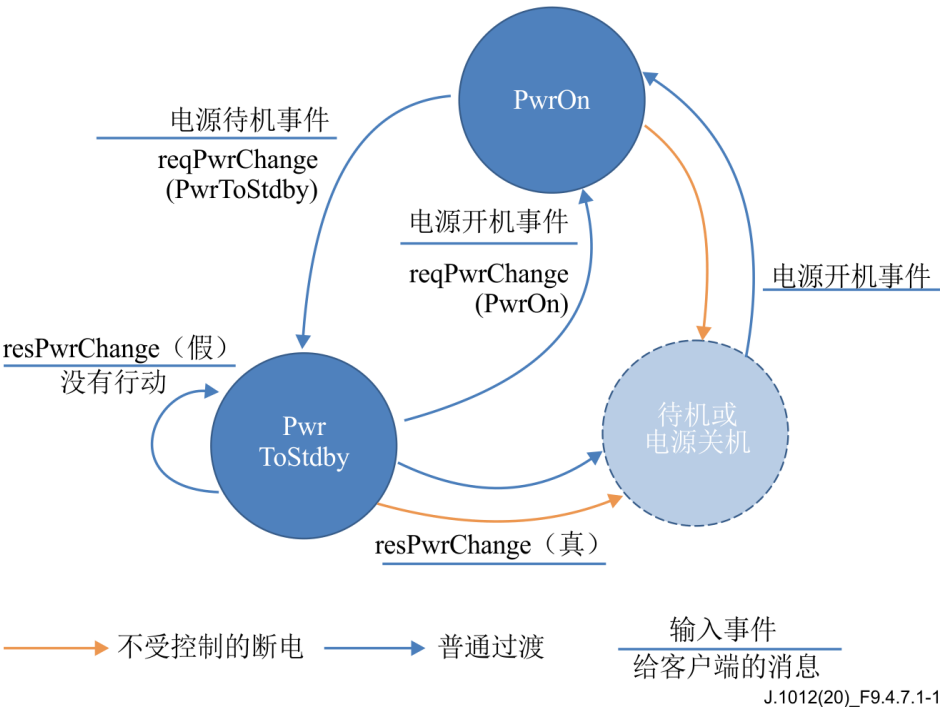


图9.4.7.1-1 – ECI主机电源状态以及与受管客户端的主要交互

ECI客户端和**ECI主机**应能管理从不受控制的断电事件中的恢复。在这种情况下，允许以非永久的方式来阻止通常的**ECI客户端**和**ECI主机**功能，以期将**用户**可能遇到的问题降至最低。

CPE可以选择在网络事件功能或其他低功耗模式下自低功耗状态唤醒。**ECI**没有为这种电源模式及其与**ECI主机**或**ECI客户端**的交互定义任何特定行为，但**ECI主机**和**ECI客户端**服务在**ECI主机**处于**PwrOn**或**PwrToStdby**状态时应继续有效。特别是：没有任何特定于暂停执行的状态。

ECI客户端应能请求**ECI主机**在将来的某个时间自待机状态唤醒，并向**ECI客户端**发送一条消息。

电源管理API分为以下几组消息：

1) 电源转换：管理**ECI客户端**的有序关闭。细节在第9.4.7.2节中定义。

2) 代表**ECI客户端**定时唤醒电源功能，细节在第9.4.7.3节中定义。

9.4.7.2 功率转换API消息定义

9.4.7.2.1 概述

有关电源管理API的这一节定义了允许**ECI客户端**在**ECI主机**中，在宣布的断电事件中执行告知的关闭功能，以便为**用户**提供最佳服务。定义的消息在表9.4.7.2.1-1中列出。

表9.4.7.2.1-1 – 电源转换消息

消息	类型	方向	标签	描述
getPwrStatus	S	C→H	0x0	获取当前值电源状态。
setPwrInfo	S	C→H	0x1	请求事件通知更改电源状态。
reqPwrChange	A	H→C	0x2	电源状态变化的通知。

发送**resPwrInfo（PwrDown）**后，**ECI客户端**不应终止，但在接收到**reqPwrChange（PwrOn）**消息时，准备恢复常规功能。

9.4.7.2.2 getPwrStatus消息

C→Huchar**getPwrStatus()**

- 该消息返回**ECI主机**的当前电源状态。

属性定义：参见表9.4.7.2.2-1。

表9.4.7.2.2-1 – 主机电源状态值

名称	值	描述
PwrOn	0x00	**ECI主机**的默认IP地址。
PwrToStdby	0x01	用于WAN（互联网）通信的**ECI主机**IP地址。
RFU	其他	保留以供未来使用。

9.4.7.2.3 setPwrInfo消息

C→HsetPwrInfo(bool**pwrInfo)**

- 该消息允许进入和离开受控断电模式，并控制**ECI主机**在电源状态改变事件时发送**ECI客户端resPwrChange**消息。

属性定义：

- **pwrInfo**为**真**，即为受控电源模式；**pwrInfo**为**假**，即为非受控电源模式。

语义描述：

- 当**pwrInfo**为真时，**ECI主机**将通知**ECI客户端**电源状态发生改变，并且在**ECI客户端**确认reqPwrChange（PwrToStby）之前不会关闭**ECI客户端**。一旦**pwrInfo**为**假**，**ECI主机**将不通知**ECI客户端**电源状态发生变化，并将“随意”关闭**ECI客户端**。
- 启动后，每个ECI客户端的PowerInfo状态都为假。

注 – 建议依赖受控断电的**ECI客户端**不要从断电周期敏感活动开始，直到向**ECI主机**发送**reqPwrInfo（真）**消息。

9.4.7.2.4 reqPwrChange消息

H → C reqPwrChange(uchar**hostPwrState) →**
C→H resPowerChange(bool **ready)**

- 该消息表示电源状态发生变化，如果参数为**PwrToStdbyRequest**，**ECI客户端**可确认并以受控方式进入待机状态，或者在它当前正在执行重要软件任务的情况下予以拒绝。

请求参数定义：

hostPwrState:uchar	新的**ECI主机**电源状态。表9.4.7.2.2-1中定义了可能的值。

响应参数定义：

ready: bool	指示**ECI客户端**进入待机状态的准备情况。

语义描述

- 如果**ECI客户端响应**为负（未准备好），则**ECI主机**应重发该消息。最小重复率和超时的数字在[b-ITU-T J Suppl. 7]中提出。

先决条件请求：

1) PwrInfo ==真。

2) **ECI主机**中有一个（最近的）电源状态发生变化，**ECI客户端**尚未确认已准备好进入待机状态。

后置条件响应：

1) 如果**就绪==真**，则**ECI客户端**准备进入待机状态，如果**就绪==假**，则不进入。

错误代码在表9.4.7.2.4-1中定义。

表9.4.7.2.4-1 – ansPwrChange错误代码

名称	描述
ErrPwrInfoNot	参见表9.4.7.4-1。

注 – ECI主机只接受ErrPwrInfoNot错误信息。

9.4.7.3 从待机唤醒消息定义

9.4.7.3.1 概述

有关电源管理API的这一节定义了允许**ECI客户端**在预编程时间内恢复执行的功能，如果需要，将**CPE**从待机电源状态唤醒。定义的消息在表9.4.7.3-1中列出。

表9.4.7.3-1 – 从待机唤醒消息

消息	类型	方向	标签	描述
setPwrWakeup	设置	C→H	0x3	设置**ECI客户端**的唤醒时间。
reqPwrWakeupEvent	A	H→C	0x4	提示唤醒计时器到期。

9.4.7.3.2 setPwrWakeup消息

C→HsetPwrWakeup(uint **time)**

- 该消息设置一个计时器：**time**时间后，如果需要，**ECI主机**应将从待机状态唤醒**ECI客户端**，并发送**reqPwrWakeupEvent()**。

属性定义：

time:uint	以秒为单位的时间，直到**ECI主机**为**ECI客户端**生成一个唤醒事件。值0意味着**ECI客户端**不需要任何唤醒事件。

详细的语义：

- 如果**ECI主机**不受阻碍，则它将从待机中唤醒并**立即启动ECI客户端**。如果遇到阻碍，则可在此后尽快发送唤醒事件。时间精度要求在[b-ITU-T J Suppl. 7]中提出。

9.4.7.3.3 reqPwrWakeupEvent消息

H→CreqPwrWakeupEvent() →
C→HresWakeupEvent()

- 该消息通知**ECI客户端**其唤醒计时器到期。当完成关键的唤醒事件处理时，**ECI客户端**应通过一个**响应**来确认该**请求**。

详细的语义：

- **ECI主机**应尝试在连续的**ECI客户端**初始化事件中重新发送该消息，直到**ECI客户端**使用一条**resPwrWakeupEvent()**消息予以确认为止。事件在**PwrOn**电源状态期间发送，但在**PwrToStdby**期间延迟。

先决条件请求：

1) **ECI客户端**的电源唤醒计时器先前已做设置并已过期。
2) 该事件尚未利用一个**响应**予以确认。
3) **ECI主机**处于**PwrOn**电源状态。

后置条件响应：

4) **ECI主机**应根据匹配请求的电源变化事件来停止发送**reqPwrWakeupEvent()**消息；参考先决条件2）。

9.4.7.4 功率转换API的错误代码

API特定错误的值可由该API的**响应**消息返回，列于表9.4.7.4-1中。

表9.4.7.4-1 – 功率转换API的错误代码

名称	值	描述
ErrPwrInfoNot	-256	**ECI客户端**指明它不请求了解电源状态变化事件。

9.4.8 用于访问国家/语言设置资源的API

9.4.8.1 引言

用于国家和语言设置的API允许**ECI客户端**或**ECI主机**分别向**ECI主机**或**ECI客户端**请求用户的实际国家和语言设置。表9.4.8.1-1列出了有关国家/语言设置API的消息。

表9.4.8.1-1 – 国家/语言设置API消息

消息	类型	方向	标签	描述
reqHCountry	A	C→H	0x0	请求实际**ECI主机**首选的国家/地区设置。
reqCCountry	A	H→C	0x1	请求实际**ECI客户端**首选的国家/地区设置。
reqHLanguage	A	C→H	0x2	请求实际**ECI主机**首选的语言设置。
reqCLanguage	A	H→C	0x3	请求实际**ECI客户端**首选的语言设置。

9.4.8.2 国家/语言API消息定义

9.4.8.2.1 reqHCountry设置消息

C→HreqHCountry() →
H→CresHCountry setting (uint**iso_3166_country_code)**

- 该消息允许**ECI客户端**请求**用户**当前所在国家的实际设置，并从**ECI主机**接收存储的国家设置的**响应**。

响应参数定义：

iso_3166_country_code:uint	该字段包含当前的**ECI主机**国家设置。国家代码是一个24位字段，它使用ISO 3166-1 alpha 3 [ISO 3166-1]规定的3个大写字符来标识主机国家。根据[ISO/IEC 8859-1]，每个字符都被编码为8位。

错误代码列在表9.4.8.2.1-1中。

表9.4.8.2.1-1 – reqHCountry错误代码

名称	描述
ErrCountryNotExists	参见表9.4.8.2.5-1。

9.4.8.2.2 reqCCountry设置消息

H→CreqCCountry() →
C→HresCCountry setting (uint**iso_3166_country_code)**

- 该消息允许**ECI主机**请求用户当前所在国家的实际设置，并从**ECI客户端**接收存储的国家设置的**响应**。

响应参数定义：

iso_3166_country_code:uint	该字段包含当前的**ECI主机**国家设置。国家代码是一个24位字段，它使用ISO 3166-1 alpha 3 [ISO 3166-1]规定的3个大写字符来标识主机国家。根据[ISO/IEC 8859-1]，每个字符都被编码为8位。

错误代码列在表9.4.8.2.2-1中。

表9.4.8.2.2-1 – reqCCountry错误代码

名称	描述
ErrCountryNotExists	**参见表**9.4.8.2.5-1。

9.4.8.2.3 reqHLanguage设置消息

H→CreqHLanguage(uint**iso_3166_language_code) →**
C→HresHLanguage setting()

- 该消息允许**ECI客户端**请求**用户**当前偏好的语言的实际设置，并从**ECI主机**接收存储的语言设置的**响应**。

响应参数定义：

iso_3166_language_code:uint	该字段包含当前的**ECI主机**语言首选设置。这是一个24位字段，使用[ISO 639-2]指定的3个小写字符来标识语言。可以使用ISO 639-2/B和ISO 639-2/T。根据[ISO/IEC 8859-1]，每个字符都被编码为8位。

错误代码列在表9.4.8.2.3-1中。

表9.4.8.2.3-1 – reqHLanguage错误代码

名称	描述
ErrLanguageNotExists	参见表9.4.8.2.5-1。

9.4.8.2.4 reqCLanguage设置消息

H→CreqCLanguage(uint**iso_3166_language_code) →**
C→HresCLanguage setting()

- 该消息允许**ECI主机**请求**用户**当前偏好的语言的实际设置，并从**ECI客户端**接收存储的语言设置的**响应**。

响应参数定义：

iso_3166_language_code:uint	该字段包含当前的**ECI主机**语言首选设置。这是一个24位字段，使用[ISO 639-2]指定的3个小写字符来标识语言。可以使用ISO 639-2/B和ISO 639-2/T。根据[ISO/IEC 8859-1]，每个字符都被编码为8位。

错误代码列在表9.4.8.2.4-1中。

表9.4.8.2.4-1 – reqCLanguage错误代码

名称	描述
ErrLangageNotExists	参见表9.4.8.2.5-1。

9.4.8.2.5 国家/语言设置API的错误代码

表9.4.8.2.5-1列出了此API的**响应**消息可以返回的API特定错误的值。

表9.4.8.2.5-1：国家/语言设置API的错误代码

名称	值	描述
ErrCountryNotExists	-256	**ECI主机**指明**用户**尚未声明其目前居住的国家。
ErrLangageNotExists	-257	**ECI主机**指明**用户**尚未为任何**用户**接口通信声明其首选的语言。

9.5 用于ECI特定ECI主机资源的API

9.5.1 用于ECI特定ECI主机资源的API列表

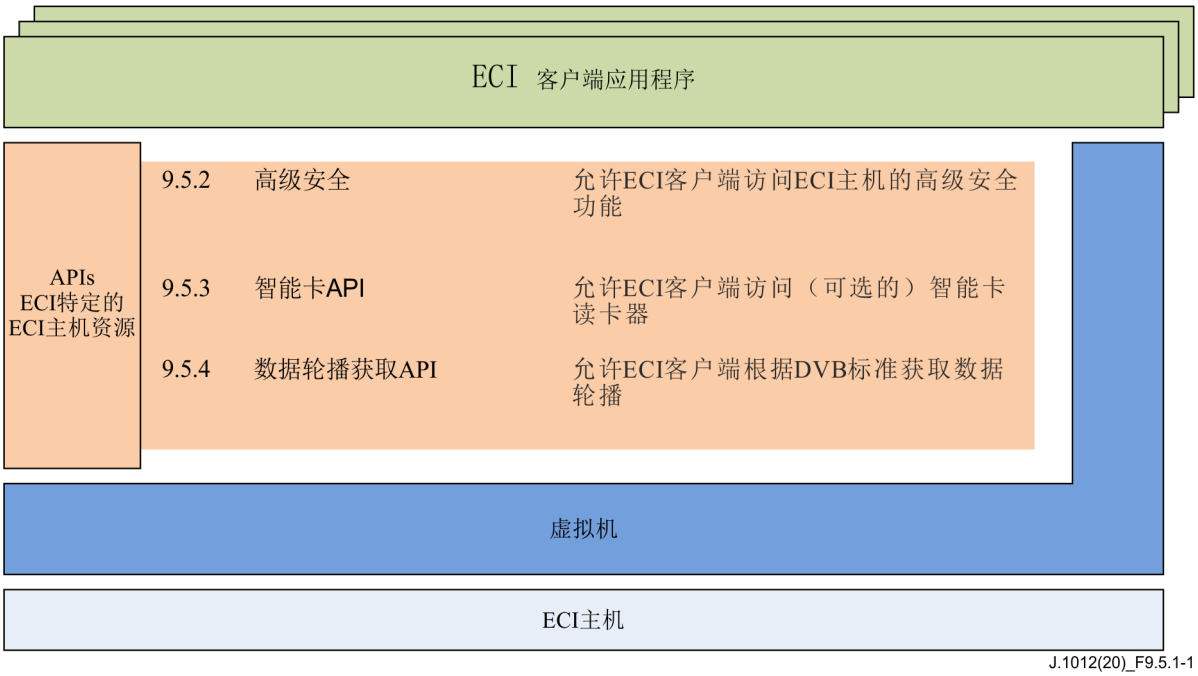


图9.5.1-1 – 第9.5节中定义的API的框图

表9.5.1-1列出了第9.8节中涵盖的各API，表9.5.1-1使用**ECI体系结构**说明了第9.5节中定义的API的位置。

表9.5.1-1 – 第9.5节中定义的API列表

条款	API名称	描述
9.5.2	高级安全API	允许**ECI客户端**访问**ECI主机**的高级安全功能。
9.5.3	**智能卡**API	允许**ECI客户端**访问（可选的）**智能卡**读卡器。
9.5.4	数据轮播获取API	允许**ECI客户端**根据DVB标准获取数据轮播。

9.5.2 高级安全API

9.5.2.1 引言

在加载**ECI客户端**时，**ECI主机**会分配一个适当的高级安全时隙（**ECI客户端**类型或**微服务器**类型）。该时隙将在该**ECI客户端**的生命周期中可用。**ECI主机**应通过加载包含**平台操作**公钥的**平台操作证书**链来初始化该时隙。这将任何有意义的进一步交换与AS时隙绑定到**平台操作**密钥的持有者。

高级安全API允许**ECI客户端**与CPE中的高级安全功能进行交互。**ECI客户端**与AS功能之间有几种类型的交换，通常由**ECI客户端**来启动。**ECI客户端**在完成较长的AS操作后会收到一个信号。

AS时隙支持多个会话，允许在**AS时隙**中重用存储的信息（状态和配置）进行多媒质解密和**重加密会话**。**AS时隙**为每个会话存储一个称为顶级“链路密钥”（LK_1）的中间密钥。

会话的新控制字可以根据其LK_1来快速计算。

AS时隙还可以计算可用于**ECI客户端**应用程序的秘密“身份验证密钥”，允许将高度安全的秘密信息传递给**ECI客户端**。

AS时隙具有由**ECI客户端**初始化并定义其操作模式的配置。**AS时隙**允许客户端对其配置进行验证：有两个基本的验证模式：

1) **密钥阶梯模式**。认证作为控制字计算的一部分：在计算中使用时隙的配置来生成加密内容的控制字，并且需要相同的信息来计算用于解密内容的正确控制字，从而隐式地认证配置。

2) **验证密钥模式**。使用验证数据，通过显式验证功能来执行验证，验证数据只能由**ECI客户端**的提供商来生成。该功能对于配置为重加密的**AS时隙**实际上是必需的，因为这不能基于正确的解密作为验证手段。

除了上述模式之外，**ECI客户端**可以要求通过要求“在线认证”对每个时隙初始化执行重新验证。或者，可以执行“离线认证”。为了成功认证，所选的认证模式必须与用于生成供应商提供之认证的数据相匹配。

整个AS API被分成若干独立的API，允许使用它来反映**ECI主机**和**ECI客户端**的功能：

1) AS通用API：该API定义通用AS功能。所有**ECI主机**和**ECI客户端**都应支持它。

2) AS解密API：该API定义解密特定的AS功能。所有能够解密的**ECI主机**和**ECI客户端**都应支持它。

3) AS出口API：该API定义出口特定的AS功能。所有能够支持解密和出口的**ECI主机**和**ECI客户端**都应支持它。支持出口的**ECI主机**也应支持加密。

4) AS加密API：该API定义加密特定的AS功能。所有能够加密的**ECI主机**和**ECI客户端**都应支持它。

以下约束条件适用：

- **ECI客户端**应支持解密或加密，并且不需要同时支持两者。

ECI主机和**ECI客户端**应使用**ECI主机**接口发现资源来为彼此提供有关彼此功能的信息。**ECI主机**应根据发现结果分配适当的时隙：需要加密的**ECI客户端**的加密时隙和需要解密的**ECI客户端**的解密AS时隙。

注 – 提供补充功能的函数可存在于不同的API中：AS通用API和更特定的AS API。

为了体现**ECI主机**功能（解密、出口和加密支持），AS通用API中的消息只需要**ECI主机**支持。

AS API的消息根据[ITU-T J.1014]第8.2.4节和第9.9节中定义的AS功能来定义。[ITU-T J.1014]第8.2.4.1节概述了AS功能。[ITU-T J.1014]中的定义省略了第一个参数，即slotId参数：这由**ECI主机**来提供。

本API定义中使用的许多参数类型定义和值定义在[ITU-T J.1014]中进行定义。该API的错误代码在[ITU-T J.1014]中进行定义，在本建议书中并没有按消息在消息上特别列出。参数值的错误代码对应参数序列，按[ITU-T J.1014]中引用函数定义的那样进行计数，通常有一个附加的参数（slotId）。

9.5.2.2 高级安全通用API消息定义

9.5.2.2.1 概述

高级安全通用API提供如表9.5.2.2.1-1所列的消息。

表9.5.2.2.1-1 – 高级安全通用消息

消息	类型	方向	标签	描述
reqAsInitSlot	A	C→H	0x0	初始化**AS时隙**。
callAsNextKeySession	S	C→H	0x1	更改为会话的下一个随机密钥。
reqAsStopSession	A	C→H	0x2	停止会话。
reqAsLoadSlotLk	A	C→H	0x3	计算顶级链路密钥（LK1）。
reqAsComputeAkClient	A	C→H	0x4	计算**ECI客户端**应用程序的验证密钥。
reqAsClientChalResp	A	C→H	0x5	在数据上应用**ECI客户端**验证密钥并返回结果。
getAsSlotRk	S	C→H	0x6	获取**AS时隙**的随机密钥值。
getAsSessionRk	S	C→H	0x7	获取会话的随机密钥值。
getAsSessionLimitCounter	S	C→H	0x8	获取会话的当前限制计数器值。
setAsSessionLimitEvent	S	C→H	0x9	设置发送reqAsEventSessionLimit消息到**ECI客户端**的限制值。
reqAsEventSessionLimit	A	H→C	0xA	达到剩余单位的限制值时，将事件发送给**ECI客户端**。
getAsClientRnd	S	C→H	0xB	获取**ECI客户端**应用程序的新随机数。
getAsSC	S	C→H	0xC	获取会话中内容的当前加扰控制字段状态。
reqAsEventSC	A	H→C	0xD	会话中加扰控制字段变化的事件消息。
getChipsetId	S	C→H	0xE	获取**密钥阶梯块**的ChipsetID值
getImageTargetId	S	C→H	0xF	获取CPE的ECI_Image_Target_Id值

9.5.2.2.2 reqAsInitSlot消息

C→H reqAsInitSlot(uint**slotVersion,** uint**slotMode→**
H→C resAsInitSlot()

- 该消息用各种常规参数来初始化时隙。

请求参数定义：

slotVersion:uint	[ITU-T J.1014]中定义的时隙功能版本。
slotMode:uint	时隙工作的主模式；参见[ITU-T J.1014]。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的AS函数reqAsInitSlot；**ECI主机**提供slotId和POPKchain参数的值。

9.5.2.2.3 callAsNextKeySession消息

C→H callAsNextKeySession(uint**sessionId**)

- 该消息会导致会话的下一个随机密钥发生更改。

请求参数定义：

sessionId:uint	要更改其下一个随机密钥的会话。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的AS消息callAsNextKeySession；**ECI主机**提供slotId参数的值。

9.5.2.2.4 reqAsStopSession消息

C→H reqAsStopSession(uint**sessionId**) →
H→C resAsStopSession()

- 该消息停止**AS时隙**会话。

请求参数定义：

sessionId:uint	停止会话的ID。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的AS函数reqAsStopSession；**ECI主机**提供slotId参数的值。

9.5.2.2.5 reqAsLoadSlotLk消息

C→H reqAsLoadSlotLk(uint**sessId**, InputV**inputV**, ulong**spkUri,**uchar**spkIndx**) →
H→C resAsLoadSlotLk()

- 该消息计算后续可用于计算控制字的顶级链路密钥LK_1。

请求参数定义：

sessId: uint	要初始化的会话的Id。
inputV: InputV	包含芯片组公钥加密和发送方密钥签名保护的链路密钥LK_1的消息。
spkUri: ulong	随后用于计算控制字的SPK向量的使用规则，参见[ITU-T J.1014]。
spkIndx: uchar	定义SPK向量中AS时隙SPK位置的索引，随后用于计算控制字，参见[ITU-T J.1014]第7节。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的**AS时隙**函数reqAsLoadLk1；**ECI主机**提供slotId参数的值。
- 如果停止之前与另一个**AS时隙**解密会话耦合的**AS时隙**解密会话，则**ECI主机**还应发布一个reqAsDecoupleDecryptSession函数[ITU-T J.1014]（参见第9.5.2.3.1节）。

9.5.2.2.6 reqAsComputeAkClient消息

C→H reqAsComputeAkClient(InputV**inputV,** uint**nSpk**uchar**spkIndx,** PubKey**spk[16],** PubKey**popk[16],** SessionConfig**akCnf[16],** ulong**spkUri;** uchar **XT[32],** bool **online)→ H→C resAsComputeAkClient** ()

- 该消息计算用于**ECI客户端**的认证密钥。

请求参数定义：

inputV: InputV	包含芯片集公钥加密的消息以及用于计算AK的发送方密钥签名保护的r值。
nSpk: uint	SPK向量中的值的数量，参见[ITU-T J.1014]。
spkIndx: uchar	定义SPK向量中AS时隙SPK位置、POPK向量中AS时隙POPK值以及clCnf向量中AS时隙slotConfig的索引，用于计算客户端验证密钥，参见[ITU-T J.1014] 。
spk[16]: PubKey	用于计算客户端身份验证密钥的发件人**公钥关键字**; 见[ITU-T J.1014]。
popk[16]: PubKey	用于计算客户端验证密钥的平台运营商公钥关键字；参见[ITU-T J.1014]。
akCnf[16]: SessionConfig	用于计算客户端验证密钥的客户端会话配置向量，参见[ITU-T J.1014]。
spkUri: ulong	随后用于计算控制字的SPK向量的使用规则，参见[ITU-T J.1014]。
XT[32]: uchar	用于计算客户端验证密钥的扩展字段值；参见[ITU-T J.1014]。默认值为{0x00}。
online: bool	如果为真，则时隙随机密钥被用于验证密钥计算，以迫使供应商进行新的验证密钥计算。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的AS函数reqAsComputeAkClient；**ECI主机**提供slotId参数的值。

9.5.2.2.7 reqAsClientChalResp消息

C→H reqAsClientChalResp(uchar**challenge**[16]);→
H→C reqAsClientChalResp(uchar**response**[16])

- 该消息使用由reqAsComputeAkClient消息（在[ITU-T J.1014]中定义）计算得到的客户端验证密钥来解密128位挑战参数输入，以产生128位响应参数输出。

请求参数定义：

challenge[16]: uchar	将由客户端验证密钥解密的128位输入。

响应参数定义：

response[16]: uchar	128位解密输出。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的AS函数reqAsClientChalResp；**ECI主机**提供slotId参数的值以及承载“响应”参数结果的**响应**消息。

9.5.2.2.8 getAsSlotRk消息

C→H SymKey**getAsSlotRk**()

- 该消息读取**ECI客户端AS时隙**会话的随机密钥。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的**AS**函数getAsSlotRk；**ECI主机**提供slotId参数的值。

9.5.2.2.9 getAsSessionRk消息

C→H SymKey**getAsSessionRk**(uint**sessionId,** uint**rkIndx)**

- 该消息读取**ECI客户端**会话的当前（rkIndx == 0）或下一个（rkIndx == 1）随机密钥，其标识符为sessionId。

请求参数定义：

sessionId: uint	为其检索随机会话密钥的会话的ID。
rkIndx: uint	确定是否要检索当前（rkIndx == 0）或下一个（rkIndx == 1）随机会话密钥。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的AS消息getAsSessionRk；**ECI主机**提供slotId参数的值。

9.5.2.2.10 getAsSessionLimitCounter消息

C→H ulong**getAsSessionLimitCounter**(uint**sessionId**)

- 该消息返回**ECI客户端**sessionId的限制计数器值。

语义描述：

- 该函数相当于[ITU-T J.1014]中定义的AS函数getAsSessionLimitCounter；**ECI主机**提供slotId参数的值。

请求参数定义：

sessionId: uint	要检索会话限制计数器的会话的Id。

9.5.2.2.11 setAsSessionLimitEvent消息

C→H ulong**setAsSessionLimitEvent** (uint**sessionId,** ulong**eventLimit**)

- 该消息为**ECI客户端**会话的limitCounter设置限制值eventLimit，其标识符为sessionId，以将reqAsEventSessionLimit消息返回给**ECI客户端**。

请求参数定义：

sessionId: uint	要为其设置会话eventLimit的会话的Id。
eventLimit: ulong	要设置的事件限制的值。

语义描述：

- 该函数相当于[ITU-T J.1014]中定义的AS函数setAsSessionLimitEvent；**ECI主机**提供slotId参数的值。

9.5.2.2.12 reqAsEventSessionLimit消息

H→CreqAsEventSessionLimit (uint**sessionId**)
C→Hres**AsEventSessionLimit ()**

- 该消息返回**ECI客户端**sessionId的限制计数器值。

响应参数定义：

sessionId: uint	生成一个eventLimit事件的会话的Id。

语义描述：

- 该函数相当于[ITU-T J.1014]中定义的AS函数reqAsEventSessionLimit；**ECI主机**删除slotId参数。

9.5.2.2.13 getAsClientRnd消息

C→H SymKey**getAsClientRnd()**

- 该消息返回一个128位的随机数。

语义描述：

- 该函数相当于[ITU-T J.1014]中定义的AS消息getAsClientRnd。

9.5.2.2.14 getAsSC消息

C→H uint**getAsSC**(uint**sessionId**)

- 该消息返回会话中内容的当前加扰控制字段状态。

请求参数定义：

sessionId: uint	要为其检索当前加扰控制字段的会话的Id。

语义描述：

- 该函数相当于[ITU-T J.1014]中定义的AS函数getAsSC；**ECI主机**提供slotId参数的值。

9.5.2.2.15 reqAsEventSC消息

H→CreqAsEventSC(uint**sessionId**; uint**scramblingControlField**)
C→Hres**AsEventSC()**

- 该消息指示带标识符sessionId的会话中加扰控制字段出现的改变。

响应参数定义：

sessionId: uint	发生扰乱状态字段更改的会话的Id。
scramblingControlField: uint	加扰状态字段的新值。有关这些值及其语义的定义，参见[ITU-T J.1014]第9.9节。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的AS函数reqAsEventSC；**ECI主机**将删除slotId参数的值。

9.5.2.2.16 getChipsetId message

C→H ulong**getChipsetId**()

- 该消息返回[ITU-T J.1014]中定义的**密钥阶梯块**的ChipsetID值。

9.5.2.2.17 getImageTargetId message

C→H ECI_Image_Target_Id**getImageTargetId**()

- 该消息返回表6.2.2.2-1中定义的CPE的ECI_Image_Target_Id值。

9.5.2.3 高级安全解密API消息定义

9.5.2.3.1 概述

高级安全解密API提供了如表9.5.2.3.1-1中所列的消息。

可以耦合两个解密会话，允许使用不同的控制字来解密两个内容流，在解密后它们被视为单个内容项。

示例： 体育频道可以用多个声音频道进行广播，如果特定的订阅可用于解密，则只有特定语言的声音频道才可用。只有一个会话可以耦合到另一个会话。

表9.5.2.3.1-1 – 高级安全解密消息

消息	类型	方向	标签	描述
reqAsAStartDecryptSession	A	H→C	0x0	在**ECI客户端**的**AS时隙**中启动一个解密会话。
reqAsComputeDecrCw	A	H→C	0x1	计算一个解密控制字。
reqAsAuthDecrSlotConfig	A	H→C	0x2	使用认证机制（解密模式）认证时隙配置。

9.5.2.3.2 reqAsStartDecryptSession消息

C→HreqAsAStartDecryptSession(ushort**mh**, PubKey**spk**, SessionConfig**config,** ScrambleMode**sm**) →
H→CresAsASartDecryptSession(uint**sessionId)**

- 该消息在**ECI客户端**的**AS时隙**中启动一个解密会话。

请求参数定义：

mh: ushort	其内容被解密的**媒质句柄**（由**ECI主机**用来将要解密的内容与分配给该会话的解密资源相关联）。
spk:PubKey	该会话的发送方公钥。
config:SessionConfig	会话配置。
sm: ScrambleMode	使用的解扰模式。定义参见表9.5.2.3.2-1。参阅注释。
注： sm参数中的信息不应与随后的reqAsComputeDecrCw消息的cwUri参数相矛盾。	

表9.5.2.3.2-1：ScrambleMode定义

```
typedef  ScrambleMode  {
uchar     modeRef;
uchar     mode[16] ;
}  ScrambleMode;
```

modeRef的定义参见表9.9.2.11-1。

表9.5.2.3.2-2 – modeRef定义

名称	值	描述
ScrambleModeHost	0x01	主机应根据标准化或专有信息选择（解扰）扰码模式。
ScrambleModeDvb	0x02	使用加扰模式的DVB定义。模式字段的字节0包含一个值，它与[IEC 62766-5-2]中定义的Scrambling_descriptor的scrambling_mode字段具有相同的含义。对字节0取值0x02、0x03或0x10（即，DVB CSA1/2，用于解扰和DVB-CISSA版本1模式的DVB CSA3模式），字节1具有以下含义： 值== 0x01：TS模式（解扰）加扰。 值== 0x02：PES模式（解扰）加扰。 保留所有其他值；保留模式字段的所有未用字节。参见注1。
ScrambleModeCencEnum	0x03	扰码模式在[ITU-T T.871]中定义或模式字段的字节0定义为： 值== 0x01：CENC CTR模式。 值== 0x02：CENC CBC模式。 保留字节0的其他值。 对以上定义的字节0，字节1指示子方案： 值== 0x01：定义的主机，用于从以下定义的值之一中选择加密。 值== 0x02：在[W3C GIF V89a]中定义的全段加密。 值== 0x03：[W3C PNG]中定义的子样本加密。 保留字节1的其他值。 对字节0的其他值，保留字节1。 保留字节2-15。 参见注2。
RFU	其他	保留以供未来使用。
注1 – **ECI主机**应至少支持用于解扰的DVB CSA1/2、DVB CSA3模式以及用于加扰和解扰的DVBCISSA版本1模式。 注2 – **ECI客户端**或（如果允许的话）**ECI主机**可以选择一种适合应用程序加密的加扰模式；特别考虑到流式应用程序通常使用CBC全段加密和存储应用程序通常使用CTR模式，并可受益于子样本加密。		

响应参数定义：

sessionId: uint	创建的会话的Id。

语义描述：

- 该消息相当于[ITU-T J.1014] 中的AS函数reqAsASartDecryptSession；**ECI主机**提供slotId参数的值，并在**响应**消息中返回sessionId结果。

当为同一个**媒质句柄**启动第二个解密会话时，**ECI主机**也发布一个reqAsCoupleDecryptSession函数[ITU-T J.1014]，以便将这些会话耦合为解密会话，并将第二个会话耦合到第一个会话。

9.5.2.3.3 reqAsComputeDecrCw消息

C→HreqAsComputeDecrCw(int **sessionId**, ulong**cwUri,**uint**nSpk**, uint**nElk**, SymKey**elk**[24], PubKey**spk**[16], PubKey**popk**[16], SessionConfig**config**[16], uchar**XT**[32], uint**rkIndx,** Field2**field2,** uint**cwIndx**) →
H→CresAsComputeDecrCw ()

- 该消息计算解密控制字。

请求参数定义：

sessionId: int	要为其计算控制字的会话Id。
cwUri: ulong	cwUri定义控制字的应用程序。cwUri值在[ITU-T J.1014]第7.5节中进行定义。
nSpk: uint	SPK向量中的SPK值的数量。
nElk: uint	ELK向量中Elk值的数量。
elk[24]: SymKey	对称加密密钥值的向量，将通过密钥阶梯机制连续解密。值Elk [nElk-2]为内容属性验证的字段1输入，如[ITU-T J.1014]第8.2.3节定义，使用如[ITU-T J.1014]第8.2.4.7节定义的函数。
spk[16]: PubKey	如[ITU-T J.1014]第.5节中定义的发送方公钥的向量。
popk[16]: PubKey	平台运营商公钥的向量，如[ITU-T J.1014]第7.5节定义。
config[16]: SessionConfig	客户端会话配置的向量，如[ITU-T J.1014]第7.5节定义。
XT[32]: uchar	控制字机制的备用输入，如[ITU-T J.1014]第7.5节定义。
rkIndx: uint	确定是否在控制字计算中使用当前（rkIndx == 0）或下一个（rkIndx == 1）随机会话密钥。
field2: Field2	更大的内容属性，字段1中未经验证的内容，如[ITU-T J.1014]第8.2.3节定义。
cwIndx: uint	要计算的控制字索引：偶数为0，奇数控制字为1；对基于文件的解密没有意义。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的AS函数reqAsComputeDecrCw；**ECI主机**提供slotId参数的值。

9.5.2.3.4 reqAsAuthDecrSlotConfig消息

C→HreqAsAuthDecrSlotConfig(uint**sessionId**, InputV**inputV**; uchar**nSpk,** uint**spkIndx**, PubKey**spk**[16], PubKey**popk**[16], SessionConfig**cnf**[16], ulong**spkUri**, uchar**XT**[32], bool **online,** uchar **verifier[16]**) →
H→CresAsAuthDecrSlotConfig ()

- 该消息使用认证机制（解密模式）认证时隙配置。

请求参数定义：

sessionId: uint	要验证时隙配置的会话的Id。
inputV: InputV	包含芯片集公钥加密的消息以及用于计算AK（用于验证**AS时隙**配置）的发送方密钥签名保护的r值。
nSpk: uchar	SPK向量中的SPK值的数量。
spkIndx: uint	定义SPK向量中AS时隙SPK位置、POPK向量中AS时隙POPK值以及clCnf向量中AS时隙slotConfig的索引，用于计算客户端验证密钥，参见[ITU-T J.1014] 。
spk[16]: PubKey	发送方公钥的向量，如[ITU-T J.1014]第7.5节定义。
popk[16]: PubKey	平台运营商公钥的向量，如[ITU-T J.1014]第7.5节定义。
cnf[16]: SessionConfig	客户端配置的向量，如[ITU-T J.1014]第7.5节定义。
spkUri: ulong	随后用于计算验证密钥AK的SPK向量的使用规则，参见[ITU-T J.1014]。
XT[32]: uchar	用于计算客户端验证密钥的扩展字段值；参见[ITU-T J.1014]。默认值为{0x00}。
online: bool	如果为真，则时隙随机密钥被用于验证密钥计算，以迫使供应商进行新的验证密钥计算。
verifier[16]: uchar	reqAsAuthDecrSlotConfig用于验证时隙配置的值。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的AS函数reqAsAuthDecrSlotConfig；**ECI主机**提供slotId参数的值。

9.5.2.4 高级安全出口API

9.5.2.4.1 概述

高级安全出口API提供如表9.5.2.4.1-1中列出的消息。

表9.5.2.4.1-1 – 高级安全出口消息

消息	类型	方向	标签	描述
reqAsExportConnSetup	A	C→H	0x0	设置从解密到加密会话的**出口连接**。
reqAsExportConnEnd	A	C→H	0x1	终止现有的出口会话。

9.5.2.4.2 reqAsExportConnSetup消息

C→H reqAsExportConnSetup(uint**sessId,** ushort**expMh,** uint**grpIndx**; CertSerialChain**expCh,** CertSerialChain**impCh,** CertSerialChain **auth**[]) →
H→CresAsExportConnSetup()

- 该消息设置从解密会话到出口**媒质句柄**会话的高级安全连接。

请求参数定义：

sessId: uint	**ECI客户端AS时隙**的出口会话的ID。
expMh: ushort	出口**媒质句柄**的ID，用于加密AS会话中的解密内容。
grpIndx: uint	存储出口会话连接的索引；允许的值为0或1。该参数可用于将**出口连接**验证移交**微服务器**（例如，用于预测流中**出口组**ID的即将转换）。
expCh: CertSerialChain	**ECI客户端**的**出口链**。
impCh: CertSerialChain	用于加密/输入**ECI客户端**的**入口链**。
auth[]: CertSerialChain	**入口链**的授权**证书**。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的**AS**函数reqAsExportConnSetup；**ECI主机**提供slotId、impSlotId和ImpSessId参数的值。**ECI主机**应使用出口会话的**媒质句柄**来将AS解密会话连接到相应的AS加密会话，即在[ITU-T J.1014]的reqAsExportConnSetupAS函数中提供impSlotId和impSessId参数。

9.5.2.4.3 reqAsExportConnEnd消息

C→H reqAsExportConnEnd(ushort**expMh)** →
H→C resAsExportConnEnd()

- 该消息终止现有的出口会话。

请求参数定义：

expMh: ushort	其内容交换将被终止的AS会话的出口**媒质句柄**。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的AS函数reqAsExportConnEnd；**ECI主机**提供与expMh关联的slotId和sessionId参数的值。

9.5.2.5 高级安全加密API

9.5.2.5.1 概述

高级安全加密API提供如表9.5.2.5.1-1所列的消息。

表9.5.2.5.1-1 – 高级安全加密消息

消息	类型	方向	标签	描述
reqAsStartEncryptSession	A	C→H	0x0	开始一个加密会话。
reqAsComputeEncrCw	A	C→H	0x1	计算加密控制字。
reqAsAuthEncrSlotConfig	A	C→H	0x2	使用验证机制（加密模式）来验证时隙配置和加密参数。
reqAsLdUssk	A	C→H	0x3	加载**微服务器**密钥。
reqAsMInikLk1	A	C→H	0x4	计算不对称**微客户端**初始化消息。
reqAsEventCpChange	A	H→C	0x5	加密会话中入口内容的内容属性更改事件消息。
setAsPermitCPChange	S	C→H	0x6	启用/禁用入口的内容属性CP更改将在加密会话中对加密控制字选择产生作用。
setAsSC	S	C→H	0x7	设置加密会话的加密内容的加扰控制字段。

9.5.2.5.2 目标客户端链定义

微服务器可以使用**证书处理系统**来提供非对称客户端验证的强大实施方案。**ECI**定义证书链，以允许此类**微客户端**验证。这样的目标链被用作reqAsMInikLk1消息的输入。

证书链应符合第5.4.1节的规定。涉及两种类型的**证书**：

- 微客户端证书验证单个微客户端；如果微客户端是ECI客户端，则证书的公钥应与微客户端CPE的芯片组公钥相同。
- 目标组证书验证一个或多个目标组或微客户端证书。

微DRM系统运营商可使用**ECI撤销列表**机制来为服务器安全地管理认证之**微客户端**的演变。

注 – 维护**撤销列表**是**微DRM系统**运营商的私事。

目标组**证书**的**证书**ID在表9.5.2.5.2-1中定义。

表9.5.2.5.2-1 – 目标组ID定义

语法	位数	助记符
ECI_Target_Group_Id {		
padding(4)		
type	4	uimsbf
target_group_id	20	uimsbf
target_group_version	8	uimsbf
}		

语义：

type:整数	按照表5.1.3-1的值。
target_group_id:整数	**目标组**号码，在**父**的情形下是唯一的。
target_group_version:整数	在微组更改其**证书**情况下递增。

表9.5.2.5.2-2中定义了**微客户端证书**的**证书**ID。

表9.5.2.5.2-2 – 微客户端ID定义

语法	位数	助记符
ECI_Micro_Client_Id {		
padding(4)		
type	4	uimsbf
micro_client_id	20	uimsbf
micro_client_version	8	uimsbf
}		

语义：

type: 整数	按照表5.1.3-1的值。
micro_client_id: 整数	**微客户端**号，在**父**的情形下是唯一的。
micro_client_version: 整数	在微组更改其**证书**情况下递增。

9.5.2.5.3 reqAsStartEncryptSession消息

C→H reqAsStartEncryptSession(ushort**mh**, PubKey**spk**, SessionConfig**config**, uint**nEncr**, PubKey**encrSpk[MaxSpkEncr]**, PubKey**encrPopk[MaxSpkEncr]**, ulong**encrCwUri**)→
H→C resAsStartEncryptSession()

- 该消息启动加密会话。

请求参数定义：

mh: ushort	加密内容的**媒质句柄**的标识符，为其创建一个加密会话。
spk: PubKey	发送方的公钥用于由AS系统来对发送方和LK1加密的消息进行验证。
config: SessionConfig	会话的配置。
nEncr: uint	为加密和可能的后续解密定义的附加SPK（和POPK）值的数量。最大值是MaxEncr（参见[ITU-T J.1014]）。
encrSpk: PubKey[]	带有用于加密的附加SPK值的向量。
encrPopk: PubKey[]	带有用于加密的附加POPK值的向量。
encrCwUri: ulong	CWURI值用于加密；参见[ITU-T J.1014]第8.2.2节。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的AS函数reqAsStartEncryptSession；**ECI主机**提供slotId参数的值。**ECI主机**应从mh值派生出importSlotId和importSessionId参数。

注 – 如果没有出现任何错误，则响应消息返回创建的新会话ID。

9.5.2.5.4 reqAsComputeEncrCw消息

C→H reqAsComputeEncrCw(int **sessId**, ulong**cwUri**, uint**nElk**, SymKey**elk**[24], uchar**XT**[32], uint**rkIndx.** Field2**field2.** uint**cwIndx)→**
H→C resAsComputeEncrCw()

- 该消息计算加密控制字。

请求参数定义：

sessId: int	要为其计算控制字的会话的ID。
cwUri: ulong	cwUri定义了控制字的应用。cwUri值在[ITU-T J.1014]第7.5节中定义。
nElk: uint	ELK向量中Elk值的数量。
elk[24]: SymKey	对称加密密钥值的向量将密钥阶梯机制连续解密。值Elk [nElk-2]是[ITU-T J.1014]第8.2.3节中定义的内容属性验证的字段1输入，使用[ITU-T J.1014]第8.2.4.6节中定义的函数。
XT[32]: uchar	控制字机制的备用输入，如[ITU-T J.1014]第7.5节中的定义。
rkIndx: uint	确定是否在控制字计算中使用当前（rkIndx == 0）或下一个（rkIndx == 1）随机会话密钥。
field2: Field2	字段1中未经验证的更大的内容属性内容，如[ITU-T J.1014]第8.2.3节中的定义。
cwIndx: uint	要计算的控制字索引：对偶数控制字为0，对奇数控制字为1；对基于文件的加密没有意义。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的AS函数reqAsComputeEncrCw；**ECI主机**提供slotId参数的值。

9.5.2.5.5 reqAsAuthEncrSlotConfig消息

C→HreqAsAuthEncrSlotConfig(uint**sessId**, InputV**inputV**, uchar**XT**[32], bool **online,** uchar **verifier[16]**) →
H→CresAsAuthEncrSlotConfig()

- 该消息使用认证机制（加密模式）认证时隙配置。

请求参数定义：

sessId: uint	其配置将被验证的会话的ID。
inputV: InputV	包含芯片集公钥加密的消息以及用于计算AK（用于验证**AS时隙**配置）的发送方密钥签名保护的r值。
XT[32]: uchar	控制字机制的备用输入，如[ITU-T J.1014]第7.5条中的定义。
online: bool	如果为真，则时隙随机密钥被用于认证密钥计算，以迫使供应商进行新的验证密钥计算。
verifier[16]: uchar	reqAsAuthDecrSlotConfig使用该值来验证时隙配置。

语义描述：

- 该消息相当于[ITU-T J.1014]中定义的AS函数reqAsAuthEncrConfig；**ECI主机**提供slotId参数的值。

9.5.2.5.6 eqAsLdUssk消息

C→H reqAsLdUssk(uint**sessId**, InputV**inputV,**ucharXT[32], bool **online,**
uchar**mUssk**[NUSSK])→
H→C resAsLdUssk()

- 在对能够解码内容的**ECI客户端**进行不对称验证时，该消息加载**微服务器**密钥。

请求参数定义：

sessId: uint	将要加载的**微服务器**密钥的会话ID。
inputV: InputV	包含芯片集公钥加密的消息以及用于计算AK（用于解密要加载的**微服务器**密钥）的发送方密钥签名保护的r值。
XT[32]: uchar	控制字机制的备用输入，如[ITU-T J.1014]第7.5节中的定义。
online: bool	如果为真，则时隙随机密钥被用于验证密钥计算，以迫使供应商进行新的验证密钥计算。
mUssk[NUSSK]: uchar	加密的**微服务器**密钥。

语义描述：

- 该函数相当于[ITU-T J.1014]中定义的AS函数reqAsLdUssk；**ECI主机**提供slotId参数的值。

9.5.2.5.7 reqAsMInikLk1消息

C→H reqAsMInikLk1(uint**sessId**, ECI_Certificate_Chain**ClCPK**) →
H→C resAsMInikLk1(InputV**inputV)**

- 该消息计算不对称的**微客户端**初始化消息。

请求参数定义：

sessId: uint	将要加载的**微服务器**密钥的会话ID。
ClCPK: ECI_Certificate_Chain	目标**证书链**，如第9.5.2.5.2节中的定义，用于加载**微客户端**芯片组公钥，将用于加密**微服务器**与**微客户端**之间的秘密会话密钥。

响应参数定义：

inputV: InputV	使用微客户端芯片组公钥加密并通过**微服务器**密钥签名的微DRM会话密钥。可由微客户端用作加载公共会话LK_1的消息。

语义描述：

- 该函数相当于[ITU-T J.1014]中的 AS函数reqAsMInikLk1；**ECI主机**提供slotId参数的值。

9.5.2.5.8 reqAsEventCpChange消息

H→C reqAsEventCpChange(int **sessionId**)

- 该消息请求在加密会话中对入口内容进行内容属性更改。

请求参数定义：

sessionId: int	加密会话，在其上，在输入的内容上发生内容属性更改事件。

语义描述：

- 该消息相当于[ITU-T J.1014]中的 AS函数reqAsEventCpChange；**ECI主机**删除slotId参数。

9.5.2.5.9 setAsPermitCPChange消息

C→H setAsPermitCPChange(int **sessionId**; bool **permit**)

- 该消息在加密会话中启动入口内容的内容属性更改。

请求参数定义：

sessionId: int	加密会话，对之，当内容属性将出现更改或未决时，允许自动转换控制字。
permit: bool	值为真表示授予权限，值为假表示不授予权限。

语义描述：

- 该函数相当于[ITU-T J.1014]中的AS函数setAsPermitCPChange；**ECI主机**提供slotId参数的值。

9.5.2.5.10 setAsSC消息

C→H setAsSC(int **sessionId**, uint**scramblingControlField**)

- 该消息设置加密会话中下一个加扰控制字段的值。

请求参数定义：

sessionId: int	加密会话，其加扰控制字段将设置为将在流中第一个可能的变化点上使用。
scramblingControlField: uint	加扰控制字段的值；有关允许的值及其含义，参见[ITU-T J.1014]第9.9节。

语义描述：

- 该功能相当于[ITU-T J.1014] AS功能setAsSC；**ECI主机**提供slotId参数的值。

9.5.2.5.11 高级安全（AS）API的错误代码

AS API的所有错误代码在[ITU-T J.1014]第8.2.4.15节中定义。

9.5.3 智能卡API

9.5.3.1 引言

ECI允许**ECI客户端**与一个可拆卸的本地安全模块（**智能卡**）进行接口。**ECI客户端**可以创建一条从**ECI客户端**到**智能卡**的安全信道，或者（安全智能）直接从**智能卡**到安全模块的安全信道，从而为保护控制字提供最大的稳健性。为密钥管理进行交换的实际协议细节不是由**ECI**定义的，而是全部由CA/DRM系统根据[ITU-T J.1014]中定义的**高级安全**块API进行定义的。

符合**ECI**标准的**CPE**可以有一个或多个读卡器时隙。**ECI主机**完全透明地为**ECI客户端**管理读卡器。**ECI主机**将任何插入的**智能卡**与可用的**ECI客户端**相匹配。为此，**ECI客户端**将一个卡说明符列表发布到**ECI主机**。**ECI主机**管理希望访问相同**智能卡**的**ECI客户端**之间的任何潜在冲突。**ECI主机**还为读卡器提供争用管理。

9.5.3.2 基本规范

本条款规定了**CPE**读卡器硬件及相关驱动程序和**ECI主机**软件应符合的基本标准和规范。

CPE的读卡器物理特性可以基于相关的市场需求。条件访问卡的主要格式是ID-1（信用卡大小），但也使用ID-000格式（SIM）的卡。参见[ISO/IEC 7816-1]、[ISO/IEC 7816-2]和[ISO/IEC 14496-12]。

常规的**CPE**读卡器应符合[ISO/IEC 7816-3]第5节，至少支持A类（5V）和B类（3V）操作。应支持以下引脚：C1（VCC）、C2（RST）、C3（CLK）、C5（GND）和C7（I/O）。

ECI主机可支持不符合上述要求的读卡器。对此类读卡器应做清晰标记，以便不被**用户**误认为是常规的**ECI**读卡器。

ECI主机和**CPE**读卡器硬件应支持[ISO/IEC 7816-2]第6节至第12节中定义的**ECI**相关特性。**ECI主机**应使用[ISO/IEC 7816-2]中定义的程序来初始化任何插入的卡。

ECI主机应根据实施本建议书中规范的要求来执行[ISO/IEC 7816-3]的功能。**ECI主机**应尽可能支持[ISO/IEC 7816-5]，以支持下面第9.5.3.3节中定义的AID检索功能。

9.5.3.3 智能卡访问管理

在初始化与**ECI客户端**的连接之前，**ECI主机**应根据[ISO/IEC 7816-3]第6节至第11节对协议和读卡器进行初始化。它应为协议、通信时序参数和**智能卡**的操作等级选择适当的设置。

ECI主机应能够按[ISO/IEC 7816-4]第8.2.1.2节中的定义以及[ISO/IEC 7816-4]第8.2.1节中的定义，来检索AID（应用程序标识符），以及从历史字节或初始数据字符串，按[ISO/IEC 7816-4]第8.2.2.1节中的定义，来从卡上检索。对多应用**智能卡**，**ECI主机**应能够按[ISO/IEC 7816-4]第8.2节、特别是第8.2.1.1节、第8.2.2节以及第8.2.2.3节中的定义，来检索AID列表。

ECI主机应为卡使用以下卡标识符列表：

1) 如果卡是符合[ISO/IEC 7816-3]的多用途卡，则应使用从EF.DIR的应用程序模板和EF.DIR中直接表示的AID中检索到的AID列表作为卡标识符列表。
2) 如果卡不是上述1）中的多用途卡，则应将从[ISO/IEC 7816-4]第8.1.1节或第8.1.2节定义的“历史字节”中检索的AID用作单卡标识符。
3) 如果不能按照上述1）或2）的规定检索AID，则应将[ISO/IEC 7816-4]第8.2条中定义的ATR用作单个卡标识符。用于匹配目的的ATR定义为从T0到Tk，不包括TCK（如果存在的话）。

根据上述卡标识符列表，**CPE**应相应地匹配**ECI客户端**。

如果它已准备好连接到卡，则**ECI客户端**应提供符合条件的卡标识符说明符列表。依据卡标识符说明符提供独有的卡属性，并指出，**ECI主机**应向**用户**发送**智能卡**访问解析冲突信号。在多个**ECI客户端**请求访问与卡标识符说明符相匹配的**智能卡**的情况下，此类**智能卡**插入或存在于其中的一个**CPE智能卡**读卡器中。

ECI主机应根据以下规则检测，并在可能的情况下解决卡标识与匹配的**ECI客户端**之间的任何冲突：

- 如果其卡标识符列表中的一个卡标识符与**ECI客户端**的卡标识符说明符之一相匹配，则**智能卡**被认为匹配**ECI客户端**。
- 如果一个**智能卡**匹配多个**ECI客户端**，且没有一个**ECI客户端**希望排他地访问，则会按以下顺序授予卡会话：
  - 应首先为最近与卡进行了会话的**ECI客户端**建立卡会话。
  - 如果没有这样的**ECI客户端**存在或者卡未被识别为已被插入到**CPE**的读卡器中，则可以通过由**ECI主机**选择的算法建立卡会话。

- 如果**ECI客户端**无法使用**智能卡**操作，则它应断开**智能卡**会话，以便**ECI主机**可以将之与其他尝试使用它的**ECI客户端**相匹配。

ECI客户端应能够在**智能卡**会话中处理**ECI主机**生成的“连接”和“断开连接”事件。

9.5.3.4 智能卡读卡器争用管理

本条款定义了**ECI主机**的应用程序冲突解决功能，用于管理客户端与用于访问**智能卡**的可用读卡器之间的争用。

当通过读卡器（**智能卡**会话）访问**智能卡**时，**ECI客户端**应提供**智能卡**会话优先级。这些值是：

- **活跃的**：用于主要功能，如果中断会给**用户**造成不适。一个例子是由**用户**请求的观看会话或者由**用户**事先编程的记录会话。
- **后台的**：用于后台处理，必要时可以中断 - 这是默认状态。一个例子是处理EMM消息，以获得未来的访问权限。

如果特定的**媒质句柄**不需要之，则**ECI客户端**应能够请求一张要插入的**智能卡**-指的是活跃的使用 - 引用一个或多个**媒质句柄**或者字符串（指明需要该卡的应用程序）。

如果**ECI客户端**使用以下指南请求卡，则**ECI主机**应将**用户**指向适当的读卡器：

- 如果可用，则它将尝试指向一个免费的读卡器。
- 如果没有免费的阅读器可用，则它将尝试指向后台模式阅读器。
- 如果没有后台模式或免费阅读器可用，则它应尝试指向活跃的模式阅读器，通过在这些阅读器的当前活跃会话中使用来自应用程序/**ECI客户端**的信息，可导致**用户**最轻微的恶化。

上述过程可以涉及**ECI主机**使用附加信息来将该卡与适当的阅读器类型（如物理尺寸）相匹配，例如，通过将阅读器类型与**ECI客户端**（符合**ECI客户端**成功连接的要求）相关联 – 假设未来将重新插入相同的卡类型。**ECI主机**可以为此使用自己的策略。

9.5.3.5 智能卡会话管理API

9.5.3.5.1 概述

智能卡会话管理API应提供客户对**智能卡**访问的管理权限，如第9.5.3.3节和第9.5.3.4节所定义。

对于**智能卡**会话管理，表9.5.3.5.1-1列出了可用的API消息。

表9.5.3.5.1-1 – 智能卡会话管理API消息

消息	类型	方向	标签	描述
setCardMatch	set	C→H	0x0	设置**ECI客户端**的卡识别说明符列表。
callCardSessionPrio	call	C→H	0x1	设置**智能卡**会话优先级。
getCardConnStatus	get	H→C	0x2	提供卡连接状态的状态。
reqCardConOpen	A	H→C	0x3	通知**ECI客户端**已打开卡会话。
reqCardConClose	A	H→C	0x4	通知**ECI客户端**已关闭卡会话。
reqCardConClose	A	C→H	0x5	通知**ECI主机ECI客户端**希望终止与所连接卡的会话。

9.5.3.5.2 setCardMatch消息

C→H setCardMatch(uint**matchListLenth,** CardSpecifier**matchList[])**

- 该消息允许**ECI客户端**指明**ECI客户端**希望连接到哪个卡标识符。

CardMatch属性定义

matchListLength: uint	依据说明符的matchList的长度。
matchList: CardSpecifier[]	参见表9.5.3.6.1-1：**智能卡**通信消息。根据第9.5.3.3节，**ECI主机**应使用此列表将连接的**智能卡**与**ECI客户端**进行匹配。类型定义在表9.5.3.5.2-1中给出，说明符类型字段的值在表9.5.3.5.2-2中定义。

表9.5.3.5.2-1 – 智能卡说明符的类型定义

```
#define MaxAtr 32
#define MaxAid 16

typedef struct CardSpecifier {
    bool                exclusiveFlag;
    ucharspecifierType;
    union specifier {
        struct {
            ucharatrLen;
            byte atr[MaxAtr];
        } atrSpec;
        struct {
            ucharaidLen;
            byte aid[MaxAid];
        } aidSpec;
    }
} CardSpecifier;
```

表9.5.3.5.2-2 – 智能卡说明符类型

名称	值	描述
CardSpecifierATR	0x01	卡说明符为ATR类型。如果**atrLen**字段与卡的ATR长度相同且卡的ATR字节与**atr**字段的第一个**atrLen**字节相匹配，则卡与说明符相匹配。卡的ATR在第9.5.3.5.3节T0..TCK中定义。
CardSpecifierAID	0x02	卡说明符为AID类型。如果**aidLen**字段与卡的AID长度相同且卡的AID字节与**aid**字段的第一个**aidLen**字节相匹配，则卡与说明符相匹配。卡的AID在第9.5.3.3节中定义。
RFU	其他	保留以供未来使用。

先决条件：

1) 如果**matchListLength**> 0，则**ECI客户端**准备响应invCardConOpen和invCardConClose消息。

后置条件

2) **ECI主机**将把读卡器中插入的任何卡都匹配到**ECI客户端**，若第9.5.3.3节中的定义。在匹配的情况下，它应向**ECI客户端**打开一个卡会话，如第9.5.3.5.5节中的定义。

3) 在新的matchList不再提供与当前所连接**智能卡**的匹配时，**ECI主机**不会丢弃正在运行的卡会话。出于此目的，**ECI客户端**应使用reqCardConnClose消息。

9.5.3.5.3 callCardSessionPrio消息

C→H callCardSessionPrio(uchar **priority,** uint**nrMh,** ushort**mH[],** char ***clientApplication)**

- 该消息更新卡会话优先级，并向**ECI主机**提供**媒质句柄**mH的列表以及**ECI客户端**请求或进行活动卡会话的内部原因。

调用参数定义

priority:uchar	**ECI客户端**所需卡会话的优先级。值在表9.5.3.5.3-1中定义。
nrMh: uint	**媒质句柄**的数量取决于卡上的活跃会话。
mH: ushort	需要与**智能卡**进行活跃会话的**媒质句柄**列表。
clientApplication: char *	空字符终结的字符串，其原因是**ECI客户端**需要与无关**媒质句柄**活动的**智能卡**进行活跃会话。如果该指针等于空（NULL），则不存在这样的要求。如果指针不等于空，则字符串值应对**用户**有一个有意义的值。可显示字符的最大数量为40。

表9.5.3.5.3-1 – 智能卡会话优先级值

名称	值	描述
CardPriorityBackground	0x01	**ECI客户端**卡优先级要求为后台，并在第9.5.3.4节中定义。
CardPriorityActive	0x02	**ECI客户端**卡优先级要求是活跃的，并在第9.5.3.4节中定义。
RFU	其他	保留以供未来使用。

后置条件：

1) **ECI主机**应根据**优先级**管理第9.5.3.4节中定义的卡会话，并根据需要，使用**mH**和**客户端应用程序**来解决经**用户**接口对读卡器的访问冲突问题。

9.5.3.5.4 getCardConnStatus消息

C→H uchar**getCardConStatus()**

- 该消息将当前会话连接状态返回到**智能卡**。

属性定义：参见表9.5.3.5.4-1。

表9.5.3.5.4-1 – 卡连接状态值

名称	值	描述
CardConNo	0x00	**ECI客户端**没有与**智能卡**进行会话。
CardConYes	0x01	**ECI客户端**与**智能卡**进行会话。
RFU	其他	保留以供未来使用。

9.5.3.5.5 reqCCardConOpen消息

H→C reqCCardConOpen() →
C→H resCardConOpen()

- **该消息允许ECI主机**通知**ECI客户端**关于与卡的新会话连接事件；**ECI客户端**响应确认事件正在处理中。

先决条件请求：

1) 根据第9.5.3.3节，将要建立与**ECI客户端**的卡会话。

后置条件响应：

2) **ECI客户端**将根据第9.5.3.4节中的要求来管理会话优先级。

3) 如果第9.5.3.3节中定义的卡没有用途，则**ECI客户端**应关闭会话。

9.5.3.5.6 reqCCardConClose消息

H→C reqCCardConClose () →
C→H resCardConClose ()

- **该消息允许ECI主机**通知**ECI客户端**已关闭与卡的会话。**ECI客户端**响应确认已对事件进行处理。

先决条件请求：

1) 卡从读卡器中取出或读卡器子系统中的主要故障导致连接丢失。

后置条件响应：

1) **ECI客户端**的**响应**确认**ECI客户端**已对事件进行处理，并准备接受由CardMatch属性定义的新卡连接。

9.5.3.5.7 reqHCardConClose消息

C→H reqHCardConClose() →
H→C reqHCardConClose ()

- 该消息允许**ECI客户端**向**ECI主机**指示它没有更多的、与所连接**智能卡**进行交互的目的。

后置条件响应

1) **ECI主机**将**智能卡**连接到第9.5.3.3中定义的另一个匹配的**ECI客户端**，并且不得尝试将此卡连接到**ECI客户端**（等待重新启动和关闭电源）。

2) 在可能将另一个匹配的**智能卡**重新连接到**ECI客户端**之前，**ECI主机**应等待接收**响应**。

9.5.3.6 智能卡通信API消息定义

9.5.3.6.1 概述

在由**ECI主机**管理的打开的**智能卡**会话情形下，**智能卡**命令**响应**API将提供**ECI客户端**与**智能卡**之间的通信会话原语。**ECI客户端**可以在[ISO/IEC 7816-3]第12节中定义的APDU级别（见注释）上执行与**ECI主机**的[ISO/IEC 7816-3]命令/响应交换。**ECI客户端**可以访问所有**智能卡**管理功能，并可根据需要执行自定义参数设置的重新启动和重新初始化，并检索通信设置。**ECI** API消息在表9.5.3.6.1-1中定义。

注 – 这也允许通过在APDU级别接口使用短命令和响应交换，在TPDU级别上进行T=0协议交换。

表9.5.3.6.1-1 – 智能卡通信API消息

消息	类型	方向	标签	描述
reqCardCmdRes	A	C→H	0x6	发送卡命令，获取卡回应。
reqCardReInit	A	C→H	0x7	重启卡（热启动或冷启动），并用最新的初始化首选项设置重新执行初始化序列。
callCardSetProp	设置	H→C	0x8	设置卡通信参数。
callCardGetProp	获取	H→C	0x9	获取卡通信属性/参数。

9.5.3.6.2 reqCardCmdRes消息

C→H reqCardCmdRes(byte**nodeAddrByte,** uint**cmdApduLen,** byte**cmdApdu[]) → H→C resCardCmdRes(**uint**resApduLen,** byte**resApdu[])**

- 如[ISO/IEC 7816-3]第12节定义，该消息通过**ECI主机**向**智能卡**发送一个命令APDU，并获取返回的响应APDU。在表9.5.3.6.2-1中定义了相关的错误代码。

请求参数定义：

nodeAddrByte:字节	[ISO/IEC 7810]第11.3.2.1节中定义的、已建立**智能卡**协议的T = 1协议设置的节点地址字节。如果**智能卡**协议设置为T = 0，则忽略此参数。
cmdApduLen: unit	cmd APDU的长度（以字节为单位）。请注意，**cmdApdu**的内部长度编码不得超过**cmdAduLen**。
cmdAdpu: 字节[]	要发送给卡的命令APDU。**ECI主机**会忽略cmdApdu字段中的多余字节。

响应参数定义：

resApduLen:uint	**响应**APDU的长度（以字节为单位）。
resAdpu: 字节[]	**响应**APDU从卡中收到。

先决条件请求：

1) ECI客户端有一个开放的**智能卡**会话。

2) 先前的reqCardCmdRes导致了resCardCmdRes或连接已（重新）初始化。

表9.5.3.6.2-1 – resCardCmdRes错误代码

名称	描述
ErrCardConnOpenNot	参见表9.5.3.7-1。
ErrCardConnFail	

9.5.3.6.3 reqCardReInit消息

C→H reqCardReInit(uchar**resetMode)→**
H→C resCardReInit()

- 该消息请求**ECI主机**使用resetMode重置**智能卡**，并使用最新的卡连接首选项设置重新初始化它。当处理完成（或失败）后，返回**响应**。表9.5.3.6.3-2中定义了相关的错误代码。

请求参数定义：

resetMode: uchar	参见表9.5.3.6.3-1。

表9.5.3.6.3-1 – 卡resetMode值

名称	值	描述
CardResetCold	0x01	应执行冷重启，并应重新初始化该卡，就好像它刚第一次通电一样（参见[ISO/IEC 7816-1]第6.2.3节）。
CardResetWarm	0x02	如果适用的话，应执行热重启，应重新初始化卡通信时序参数（参见[ISO/IEC 7816-3]第6.2.3节），并应再次执行[ISO/IEC 7816-3]第9节中定义的“协议和参数选择”。这可以专门用于尝试将接口时序参数切换到**ECI客户端**首选值。
RFU	其他	保留以供未来使用。

先决条件请求：

1) **ECI客户端**有一个开放的**智能卡**会话。

后置条件响应：

2) **响应**表示成功建立了接口协议和参数设置。

表9.5.3.6.3-2 – resCardCmdRes错误代码

名称	描述
ErrCardConnOpenNot	参见表9.5.3.7-1。
ErrCardConnFail	

9.5.3.6.4 callCardSetProp消息

C→H callCardSetProp (ushort**propTag,** uint**tvalueLen,** byte ***propValue)**

- 该消息将智能卡接口的propTag指示的可写属性设置为propValue。

请求参数定义：

propTag:ushort	卡通信协议属性的标签被改变。这些值在表9.5.3.6.5-2中定义。
valueLen: uint	paramValue字段的长度（以字节为单位）。
propValue: 字节 *	指向要写入paramTag指示之参数的属性值的指针。

表9.5.3.6.4-1 – callCardSetProp错误代码

名称	描述
ErrCardConnOpenNot	参见表9.5.3.7-1。

9.5.3.6.5 callCardGetProp消息

C→H callCardGetPropf(ushort**propTag,** uint**valueLen,** byte ***propValue)**

- 该消息将**智能卡**接口**propTag**所指示的可访问属性读入**propValue**。表9.5.3.6.5-1中定义了相关的错误代码。

请求参数定义：

propTag:ushort	卡通信协议属性的标签被改变。这些值在表9.5.3.6.5-2中定义。
valueLen: uint	paramValue字段的最大长度（以字节为单位）。任何多余的属性字节都不会被复制到propValue。
propValue: 字节 *	指向所请求属性值的指针。

表9.5.3.6.5-1 – callCardSetProp错误代码

名称	描述
ErrCardConnOpenNot	参见表9.5.3.7-1。

表9.5.3.6.5-2 – 卡协议属性的卡API标签值和语义

名称	标签值	描述
CardPropClass	0x0001	一个字节。值类A = 0x01，类B = 0x02，类= 0x03。保留其他值以供未来使用。只读。
CardPropAtrLen	0x0002	一个字节。**CardPropAtr**中卡的ATR的字节长度。只读。
CardPropAtr	0x0003	字节字符串，最大值为 16个字节。卡ATR冷重启。只读。
CardPropPpsExch	0x0004	如果不等于0x00，则卡和接口成功完成一次PPS交换。只读。
CardPropPpsVal	0x0004	一个字节。PSS1卡PPS交换结果的值。本建议书不支持其他值。只读。
CardPropTAEff	0x0005	一个字节。用在接口上时钟时序的TA有效值。只读。
CardPropTCEff	0x0006	一个字节。用在接口上时钟时序的TC有效值。只读。
CardPropProt	0x0007	一个字节。这表示接口设备选择用于与卡进行通信的协议。这些值在[ISO/IEC 7816-3]第8.2.3节“T”字段中定义。值0x00表示T = 0协议，值0x01表示T = 1协议。其他值可以出现（最高为0x0E）。只读。
CardPropT1IFSC	0x0008	一个字节。在[ISO/IEC 7816-3]第11.4.2节中定义了编码T = 1协议中IFSC（卡信息字段大小）的当前协议值。只读。
CardPropT1IFSD	0x0009	一个字节。在[ISO/IEC 7816-3]第11.4.2节中定义了编码T = 1协议中IFSD（设备信息字段大小=卡读卡器）的当前协议值。只读。
CardPropAidListLen	0x000A	一个字节：初始化期间从卡中检索到的卡AID的列表长度。只读。
CardPropAidList	0x000B	*（字节 [MaxAid]）：初始化期间从卡中检索到的AID列表。只读。
CardPropClassPref	0x0011	三个字节。首选类值的序列。应尝试按序建立优选的值（不违反安全=ty）。3个字节的值在**CardPropClass**中，值0x00表示“没有更多优先选择”。读和写。
CardPropImplClock	0x0012	在ATR中的TA_2位5指示时钟频率的隐含值时，应采用一个字节**TA**值。读和写。
CardPropPps1SegLen	0x0013	一个字节。值表示一个无符号的二进制数。最小值是0，最大值是0x08。代表在**CardPropPps1Seq**的PPS交换协商中尝试使用的PPS1值的数量在[ISO/IEC 7816-3]第9节中定义。请参阅注释。
CardPropPps1Seq	0x0014	一个字节。序列的最大长度为8，从PPS1的最理想值开始，以尝试在PPS交换中建立之。各值在[ISO/IEC 7816-3]第9.2节中定义。读和写。
CardPropInfdPref	0x0015	一个字节。值表示接口设备要为T1协议建立的首选IFSD值。读和写。
RFU	其他	保留以供未来使用。
注：此API不支持PPS2和PPS3的值，并且不需要**ECI主机**支持。读和写。		

9.5.3.7 智能卡API的错误代码

API特定错误的值可由该API的**响应**消息返回，列于表9.5.3.7-1中。

表9.5.3.7-1 – 智能卡API的错误代码

名称	值	描述
ErrCardOpenNot	-256	没有建立任何卡会话。
ErrCardConnFail	-257	卡会话已建立但未建立连接（重新设置后）。
RFU	**其他**	保留以供未来使用。

9.5.4 数据轮播获取API

9.5.4.1 概述

数据轮播获取API允许**ECI客户端**从第7.7.2节中定义的**ECI**格式的广播轮播中检索信息。**ECI客户端**可以使用此功能来检索可能更新的入口/出口信息。

注 – 数据轮播旨在承载准静态数据，而不是针对临时数据的优先传输协议。

ECI客户端可以直接从轮播数据中读取数据，或者请求**ECI主机**对其感兴趣的轮播项目模块或组的更新情况进行监控。对于监控，可能是在PwrOn电源状态期间或者在待机状态期间的某个指定间隔周期上进行监控。鼓励（出于功耗管理原因）让这些周期与**ECI主机**监控周期相一致。

ECI主机将尝试获取请求的数据，并将之存储在某个文件中，供**ECI客户端**通过文件系统API稍后进行访问。按照[b-ITU-T J Suppl. 7]中的提议，**ECI主机**为每个**ECI客户端**提供最少数量的并行采集信道。

表9.5.4.1-1列出了数据轮播获取API的消息。

表9.5.4.1-1 – ECI数据轮播获取API消息

消息	类型	方向	标签	描述
reqDCAcqGroupInfo	A	C→H	0x0	**ECI客户端**请求**ECI主机**读取指定ECI数据轮播的DSI消息中的GroupInfoIndication结构。
reqDCAcqModule	A	C→H	0x1	**ECI客户端**请求**ECI主机**使用模块过滤参数和各种模式将特定的**ECI**数据轮播模块获取到某个文件中。

9.5.4.2 reqDCAcqGroupInfo消息

C→H reqDCAcqGroupInfo (uint**operatorId,** uint**platformId)→**
H→C resDCAcqGroupInfo (byte**gii[])**

- **ECI客户端**请求**ECI主机**读取指定**ECI**数据轮播的DSI消息中的GroupInfoIndication结构。在表9.5.4.2-1中定义了相关的错误代码。

请求参数定义：

operatorId:uint	在PSI的data_broadcast_id_descriptor()中承载的ECI_carousel_id结构中发现的20位的**运营商**ID（参见第7.7.2.4节）。
platformId:uint	在PSI的data_broadcast_id_descriptor()中承载的ECI_carousel_id结构中发现的20位的**平台操作**ID（参见第7.7.2.4节）。

响应参数定义：

gii:字节[]	按照DVB DSM-CC [ETSI EN 301 192]的定义，承载GousepInfoIndication结构的字节数组承载于轮播的DSI中。

详细的语义：

- **ECI主机**仅提供对加载的客户端轮播的访问。

表9.5.4.2-1：reqDCGroupInfo错误代码

名称	描述
ErrDCAcqNetwAccessResource	参见表9.5.4.4-1.
ErrDCAcqNetwAccessFail	
ErrDCAcqNoCarousel	

9.5.4.3 reqDCAcqModule消息

C→H reqDCAcqModule(uchar **aid,** fileName**fname**, uint**oId,** uint**pId,** byte**dType,** uint **model,** uint **version,** uint **index,** uint **mode)** →
H→C resDCAcqModule()

- 该消息允许**ECI客户端**使用模块过滤器参数和各种模式，来请求**ECI主机**将特定的**ECI**数据轮播模块获取到文件中。

请求参数定义：

aid:uchar	获取过滤器的编号。**ECI客户端**最多可以有三个活跃的获取过滤器（值为0 .. 2）。
fname: fileName	应复制将要获取的、数据来自轮播模块的文件的名称。任何现有的数据都将被覆盖。
oId: uint	在PSI的data_broadcast_id_descriptor()中承载的ECI_carousel_id结构中发现的20位的**运营商**ID（参见第7.7.2.4节）。
pId:uint	在PSI的data_broadcast_id_descriptor()中承载的ECI_carousel_id结构中发现的20位的**平台操作**ID（参见第7.7.2.4节）。
dType: 字节	该字段应与表7.7.2-3中定义的模块组的描述符类型字段相匹配。
model: uint	承载16位无符号值，该值应与要获取组的兼容性描述符中的模型字段相匹配。参见表7.7.2.4-1。
version: uint	承载16位无符号值，根据**模式**参数位0和位1，该值应该匹配（正过滤器）或不匹配（负过滤器）或者在匹配时忽略要获取组的兼容性描述符中的版本字段。参见表7.7.2.4-1 。
index: uint	组中要访问的模块的索引。该参数应根据**模式**参数位1来解释。

mode: uint	参数由若干字段组成： **位0**：指示**版本**上的正或负滤波：0b0是正滤波，0b1是负滤波， **位1**：指示是否忽略**版本**上的过滤（值0b1表示忽略。值0b0表示不忽略）， **位2**：指示是否忽略索引（值1表示忽略）以及是否获取任何模块（对于单模块轮播），或者是否需要使用索引（模数numberOfModules，参阅表7.7.2.6-1）， **位29**：指示是否设置**ECI主机**应根据自己对该轮播的获取要求，通过检查轮播，在待机期间执行获取；以及继续进行此类获取，直至在待机和开机模式中进一步通知，直至获得所请求的数据为止， **位30**：指示获取是否应假定数据轮播正在运行，并且要在正常轮播计划时间（值0b0）内完成获取，或者获取是否应当进行，以及何时可以获取轮播以及何时获取过滤器匹配（0b1）（即等到数据出现）， **位31**：使用该过滤器**aid**启用（值0b1）或禁用（值0b0）获取。

先决条件响应：

1) 获取请求的轮播模块，遇到文件系统错误，或者如果设置了**模式**位30，则遇到采集问题。

2) **ECI主机**处于PwerOn状态，即**ECI客户端**在待机期间不会在获取时被唤醒。

后置条件响应：

1) 文件包含指定的模块或出现一个错误。

2) 当**模式**参数位30被设置时，不会出现任何采集错误。

详细的语义：

- **ECI主机**仅提供对加载之**ECI客户端**轮播的访问，并且出于**ECI主机**的目的，**ECI客户端**正在执行对广播数据轮播的监控。
- 如果没有设置，则不会执行这种待机采集。希望创建自己获取计划的**ECI客户端**可以使用第9.4.7.3节中的唤醒API来这么做。
- 在模式位31清零的情况下，**ECI主机**应提供一个“微不足道的”**响应**。

表9.5.4.3-1列出了相关的错误代码。

表9.5.4.3-1 – reqDCAcqModule错误代码

名称	描述
ErrDCAcqNetwAccessResource	参见表9.5.4.4-1。
ErrDCAcqNetwAccessFail	
ErrDCAcqNoCarousel	
ErrDCAcqCarNoGroup	
ErrDCAcqCarNoModule	
ErrDCAcqCarTimeout	
ErrDCAcqFileSystemFailure	
ErrDCAcqFileQuotaExceeded	

9.5.4.4 数据轮播获取API的错误代码

API特定错误的值可由该API的**响应**消息返回，列于表9.5.4.4-1中。

表9.5.4.4-1：TS媒质的错误代码媒质会话API

名称	值	描述
ErrDCAcqNetwAccessResource	-256	参见表9.6.2.3.7-1。
ErrDCAcqNetwAccessFail	-257	参见表9.6.2.3.7-1。
ErrDCAcqNoCarousel	-258	在可访问**ECI主机**的广播网络中找不到任何具有匹配**运营商**和**平台操作**ID的轮播。
ErrDCAcqCarNoGroup	-260	找到轮播DSI中的groupInfoIndication结构，但没有找到匹配的组。
ErrDCAcqCarNoModule	-261	找到转盘组（DII），但没有找到任何匹配的模块。
ErrDCAcqCarTimeout	-262	访问轮播DSI、DII或DDB时发生超时。
ErrDCAcqFileSystemFailure	-263	参见表9.4.5.5-1。
ErrDCAcqFileQuotaExceeded	-264	参见表9.4.5.5-1。

9.6 用于访问ECI主机解密资源的API

9.6.1 ECI主机解密API

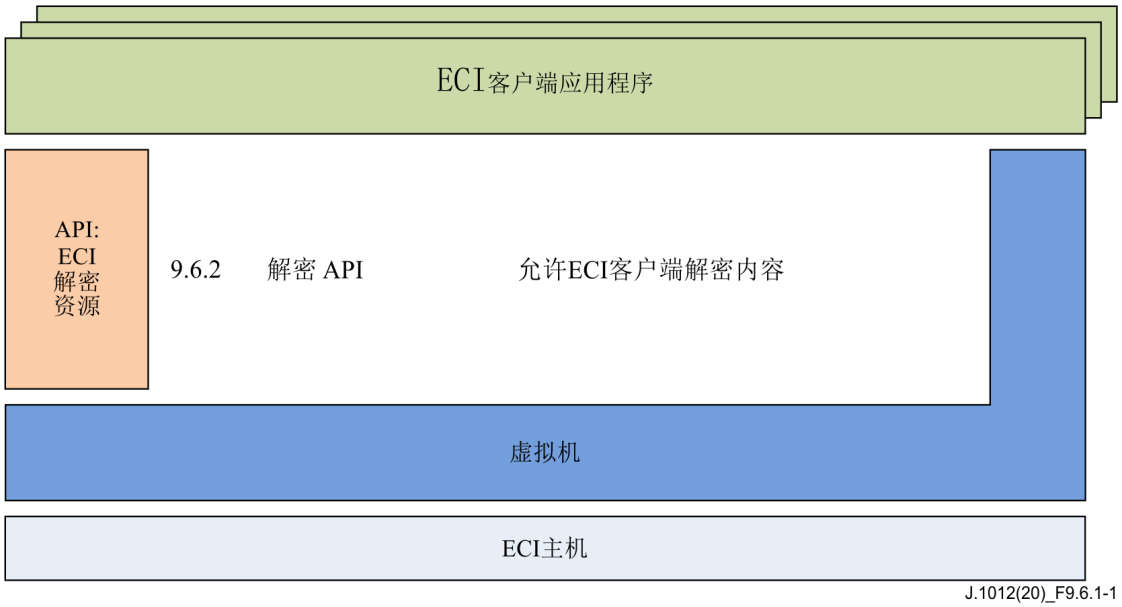


图9.6.1-1 – 第9.6节中定义的API的框图

表9.6.1-1列出了第9.6节中涵盖的API，图9.7.1使用**ECI**体系结构说明了第9.6节中定义的API的位置。

表9.6.1-1 – 第9.6节中定义的API列表

条款	API名称	描述
9.6.2	**ECI主机**解密API	允许**ECI客户端**将与特定内容元素相关的标准URI信息传送给**ECI主机**。

9.6.2 定义ECI主机解密API

9.6.2.1 引言

解密API允许**ECI主机**（如根据驻留或下载的应用程序的请求）选择与内容解密要求相匹配的**ECI客户端**，并请求解密之。在**ECI客户端**与**ECI主机**之间的所有解密消息在**媒质句**

柄情形下进行交换，它表示内容、任何相关的传送网络以及解码之所需的资源。

以下API构成解密API：

1) 适用所有媒质类型的通用媒质会话API，包括内容与**ECI客户端**之间的匹配功能。

2) 传输流解密API。

3) 文件和流解密API。

9.6.2.2 媒质会话API

9.6.2.2.1 概述

ECI客户端可以公布匹配说明符列表，通过之，**ECI主机**可以将之与内容匹配。

ECI主机可以请求一个匹配的**ECI客户端**来为媒质句柄打开解扰会话。打开会话并不意味着要开始解码。它仅确保访问其中的内容与/或元数据以及执行解扰会话所需的任何资源在**ECI主机**侧和**ECI客户端**侧都可用。**ECI客户端**应该确保在确认会话之前访问实际解扰内容所需的**智能卡**或其他资源。表9.6.2.2.1-1列出了API函数。

表9.6.2.2.1-1 – 媒质句柄解密会话API消息

消息	类型	方向	标签	描述
setDcrMhMatch	Set	C→H	0x0	向**ECI主机**发送信号，在该ID下可认可**ECI客户端**用于解密内容。
reqDcrMhOpen	A	H→C	0x1	**ECI主机请求ECI客户端**使用**媒质句柄**来打开指定类型的一个媒质会话。
reqDcrMhClose	A	H→C	0x2	**ECI主机**使用一个**ECI客户端**来关闭媒质会话。
reqDcrMhBcAlloc	A	C→H	0x3	**ECI客户端**为自己的广播网络访问目的而**请求媒质句柄**会话。
reqDcrMhCancel	A	C→H	0x4	**ECI客户端**取消与**ECI主机**的媒质会话。

9.6.2.2.2 setDcrMhMatch API消息

C→HsetDcrMhMatch(uint**matchListLength,** MatchSpecifier**matchList[])**

- 该消息允许**ECI客户端**向**ECI主机**指示其能够提供传输流解密服务的解密系统Id。

注 – 解密内容的实际能力可能取决于订阅情况、付款状态或其他条件。

SetDcrMhMatch属性定义

matchListLength:uint	依据说明符的matchList长度。
matchList: MatchSpecifier[]	表9.6.2.2.2-1。根据第9.5.3.3节，**ECI主机**应使用此列表来将内容与潜在的**ECI客户端**解密能力进行匹配。匹配说明符通过MatchSpecifier类型进行定义。MatchSpecifier的所有字段都应与内容匹配，以便生成一个匹配。

表9.6.2.2.2-1 – MatchSpecifier的类型定义

```
#define MaxMhSubFormat 16;
typedef       struct       MatchSpecifier       {
   uchar decryptIdType; /*see table 9.6.2.2.2-2*/
   union decryptId {
      bool      ECI Client ID;
      ushort dvbCaId;
      byte      uuid[16];
   }
   byte   mhType;
   byte   subFormat[MaxMhSubFormat];
} MatchSpecifier;
```

表9.6.2.2.2-2 – setDcrMhMatchdecryptIdType定义

名称	值	描述
无	0x00	与发布的**请求**中的任何内容都不匹配；在打开一个会话的情况下表示“不匹配”。
ClientEciId	0x01	**ECI客户端**的标识可以基于**ECI客户端**Id来完成，它由本建议书第7节中规定的20位值组成（不包括类型和版本字段）<<operator_id，platform_operation_id>，<vendor_id，client_id>。
ClientDvbCaId	0x02	decryptId是[CEN EN 50221]和[ETSI EN 301 192]中定义的条件访问系统标识符。该值表示dvbCaId是specifierType联合的已使用变体。dvbCaId的实际值在[CEN EN 50221]中定义。
ClientUUID	0x03	解密的是由CENC/Dash定义的DRM ID，被指定为UUID [IETF RFC 4122]。
RFU	其他	保留以供未来使用。

mhType: unit	**ECI客户端**为此ClientEciID支持的**媒质句柄**类型（主要解密模式）。
subFormat: 字节[]	该参数允许为**ECI客户端**定义其他类型的规范。对这些字节的解释取决于表9.6.2.2.2-3中定义的**mhType**。

表9.6.2.2.2-3 – subFormat类型定义

mhType值	subFormat字段的语义
ISOBMFF	subFormat字段包含适合由**ECI客户端**解码的、ISOBMFF ftyp或styp box品牌值的零个或多个顺序4CC定义。这些4CC值中的一个（或多个）值应与ISOBMFF容器的ftyp或styp box的major_brand或compatible_brands[]值相匹配。 subFormat中的值0x0000应表示没有值（总是不匹配），值0xFFFF作为第一个条目，应指的是任何品牌价值（而不管以下字节）。
其他	保留以供未来使用。

详细的语义：

当试图呈现基于传输流的内容时，**ECI主机**应尝试按优先顺序使用以下规则将内容与可用的**ECI客户端**进行匹配：

1) **ECI主机**应尝试使用第7.2.2节中定义的内容的**ECI客户端**ID来建立一组适用的匹配说明符。如果任何适用的**ECI客户端**ID和相关的匹配属性与一个**ECI客户端**的**MatchSpecifier**相匹配，则它应向该**ECI客户端**提供解密内容。如果多个**ECI客户端**

与**ECI主机**相匹配，则应使用以下过程：

a) **ECI主机**应提供通过**ECI客户端**解密的内容，它最近成功交付用于解密来自相同“内容来源”的内容的CW。

b) 如果第一个**ECI客户端**未能解密内容，则它将尝试使用匹配的备选**ECI客户端**，从而它应该应用与“内容来源”有关的、最近成功解密历史记录的**ECI客户端**顺序。

2) 如果**ECI主机**不能为内容建立任何**ECI客户端**ID，或者如果上述1）中的**ECI客户端**都不能解码内容，则**ECI主机**应尝试为内容建立一组其他ID，如第9.5.4.3节所定义。如果只有一个标识符和相关的匹配属性与一个**ECI客户端**相匹配，则**ECI主机**应向该**ECI客户端**提供用于解密的内容。如果多个**ECI客户端**与**ECI主机**相匹配，则应使用以下过程：

a) ECI**主机**应提供用于解密的内容，**ECI客户端**最近从相同的“内容来源”解密成功。

b) 如果第一个**ECI客户端**未能解密内容，则它将尝试使用匹配的备选**ECI客户端**，它因此而应该应用与“内容来源”有关的、最近成功解密历史记录的**ECI客户端**顺序。

上述“内容来源”一词至少应包括：

3) 一个DVB广播网络或其中发送TS的业务群。

4) 用于浏览器浏览的网站，它提供对内容的引用。

9.6.2.2.3 reqDcrMhOpen消息

H→C reqDcrMhOpen(ushort**mH,** MatchSpecifier **match)** →
C→H resDcrMhOpen(ushort**mH)**

- 该消息允许**ECI主机**向**ECI客户端**请求一个解密会话。**ECI客户端**应该保留通常所需的所有资源，以执行由**mh**和**match**确定的解密。在表9.6.2.2.3-1中定义了相关的错误代码。

请求参数定义

mH: ushort	要解密的内容的**媒质句柄**。
match: MatchSpecifier	匹配说明符的副本（也包含会话的媒质句柄类型）。

响应参数定义

mH: ushort	要解密的内容的**媒质句柄**。

先决条件请求：

- **ECI主机**保留了解密内容所需的全部资源。对于TS内容，这包括任何调谐或其他网络访问资源及其适用的控制，为至少一个cw对应用程序解复用资源和解扰资源。

后置条件响应：

- 在成功取得结果的情况下，**ECI客户端**保留了解码请求之会话内容通常需要的所有资源。这应该包括访问解密操作通常需要的任何外部资源（DRM服务器、**智能卡**等）。

注 – 异常或资源需要的资源，当排除需要的时，通常可获得之。

- 在返回ErrDcrUserDelay的情况下，**ECI客户端**正在等待**用户**输入以打开会话（例如，访问**智能卡**）。**ECI主机**应该重复发送reqDcrMhOpen请求（使用相同的参数），直到返回一个肯定的结果或返回一个确定的错误，或者作为选择，可以发送reqDcrMhClose来终结未决的会话。在无法获得所需的**用户**输入时，**ECI客户端**可用reqDcrMhCancel来取消。

表9.6.2.2.3-1 – reqDcrMhOpen错误代码

名称	描述
ErrDcrUserDelay	参见表9.6.2.2.7-1
ErrDcrCardMissing	
ErrDcrServiceMissing	
ErrDcrResourceMissing	
ErrDcrMmiMissing	

9.6.2.2.4 reqDcrMhClose消息

H→C reqDcrMhClose(ushort**mH)** →
C→H resDcrMhClose(ushort**mH)**

- 该消息使**ECI主机**能够关闭与**ECI客户端**的解密会话。**ECI客户端**可释放该会话的资源。

请求参数定义

mH: ushort	要关闭的会话的**媒质句柄**。

请求参数定义

mH: ushort	已关闭的会话的**媒质句柄**。

后置条件请求：

- **ECI客户端**释放专门用于会话的任何资源。

后置条件响应：

- **ECI主机**可以释放与**媒质句柄**有关的任何资源。

9.6.2.2.5 reqDcrMhBcAlloc消息

C→H reqDcrMhBcAlloc(byte**networkType[2],** uchar **priority,** char **reason[80])** →
H→C resDcrMhBcAlloc(ushort**mH)**

- 该消息允许**ECI客户端**为了安全数据获取之目的而请求连接到广播网络。

请求参数定义

networkType:字节[2]	由ECI客户端访问的广播网络类型；值依据表9.6.2.3.6.2-3。
priority:uchar	表9.6.2.2.5-1定义了访问网络的优先级。
reason:char[80]	空终结的字符串，最多80个字符，可呈现给**用户**，以解决ECI主机中用于解决本请求的资源冲突。

表9.6.2.2.5-1 – 广播网络访问优先级定义

名称	值	描述
DcrAllocPrioBackground	0x01	后台处理需要访问，当具有较高优先级的任务需要访问资源时，后台处理可能不会被准许或者可能会被中断。一个例子是访问中央多路复用的EMM或安全可更新数据。
DcrAllocPrioActivec	0x02	主解扰功能需要访问，如果不被准许（或者当中断时），会给**用户**造成不适。一个例子是由**用户**请求的观看会话或由**用户**事先编程的记录会话。
RFU	其他	保留以供未来使用。

请求参数定义

mH: ushort	打开会话的**媒质句柄**。

详细的语义：

- 在另一个任务需要具有更高优先级的网络访问资源的情况下，**ECI主机**可以使用reqDcrMhClose消息来取消会话。
- 在其不再需要访问网络的情况下，**ECI客户端**应使用reqDcrMhCancel消息来关闭会话。

后置条件请求：

1) **ECI主机**已分配所有资源，以访问所请求的网络类型。

后置条件响应：

2) **ECI客户端**应做调整，以便在开始区段采集之前，使用reqDcrTsRelocate消息来获取传输流。

表9.6.2.2.5-2 – reqDcrMhBcAlloc错误代码

名称	描述
ErrDcrNetworkAccessCapability	参见表9.6.2.2.7-1。
ErrDcrNetworkAccessResource	
ErrDcrPrioOverride	
ErrDcrResourceMissing	

9.6.2.2.6 reqDcrMhCancel消息

C→H reqDcrMhCancel(ushort**mH,** uchar **reason) →**
H→C resDcrMhCancel(ushort**mH)**

- 该消息允许**ECI客户端**关闭与**ECI主机**的解密会话。**ECI客户端**已释放会话特别需要的所有资源。

请求参数定义：

mH: ushort	要关闭的会话的**媒质句柄**。
reason: uchar	取消解密会话的原因。这些值在表9.6.2.2.6-1中定义。

表9.6.2.2.6-1 – reqDcrMhCancel原因值

名称	值	描述
DrcMhUndefined	0x00	**ECI客户端**中出现一个未定义的错误，要求它取消会话。
DcrMhCardMissing	0x01	**智能卡**需要解码，但无法成功（重新）连接，并协助在合理的时间内解密内容。
DcrMhServiceMissing	0x02	支持**ECI客户端**提供维护解密会话所需之解密服务的服务（在**CPE**外部）在合理的时间内不可用。
DcrMhResourceMissing	0x03	提供解密服务所需的资源（在**CPE**内部）在合理的时间内对**ECI客户端**不可用（不包括**DcrMhMmiMissing**）。
DcrMhMmiMissing	0x04	**ECI客户端**未成功获得在合理的时间内维护解密会话所需的、用于用户交互的MMI会话资源。
DcrMhAllocTerminate	0x05	**媒质句柄**通过reqDcrMhBcAlloc代表**ECI客户端**来分配，**ECI客户端**不再需要之。
RFU	其他	保留以供未来使用。

ECI主机取消**媒质句柄**会话的合理时间在[b-ITU-T J Suppl. 7]中提出。

响应参数定义：

mH: ushort	已取消的会话的**媒质句柄**。

先决条件请求：

- **ECI客户端**已释放了专门用于会话所需的资源。

后置条件请求：

- **ECI主机**可以释放与**媒质句柄**相关的任何资源。

后置条件响应：

- 媒质句柄会话由ECI主机关闭。

9.6.2.2.7 媒质会话API的错误代码

这些值指的是API特定的错误，它们可由该API的**响应**消息返回，列于表9.6.2.2.7-1中。

表9.6.2.2.7-1 – TS媒质的错误代码媒质会话API

名称	值	描述
ErrDcrUserDelay	-256	需要很长的延时来等待完成操作所需的、来自**用户**的输入。操作未完成。
ErrDcrCardMissing	-257	会话所需的**智能卡**不可访问/可用。
ErrDcrServiceMissing	-258	在解密操作中支持**ECI客户端**所需的**CPE**外部服务不可用。
ErrDcrResourceMissing	-259	访问或解密内容所需的**CPE**内部未定义资源不可用。
ErrDcrMmiMissing	-260	**ECI客户端**无法访问MMI。
ErrDcrDescrContinue	-261	**ECI主机**继续尝试对此TS中的内容进行解扰。
ErrDcrNetworkAccessCapability	-262	**ECI主机**没有用于定位所请求TS的网络访问资源。
ErrDcrNetworkAccessResource	-263	**ECI主机**无法获取访问所请求TS的网络访问资源。
ErrDcrPrioOverride	-264	**CPE**中较高优先级的任务需要**媒质句柄**导致**媒质句柄**会话被终止的资源。
RFU	其他	保留以供未来使用。

9.6.2.3 解扰传输流数据

9.6.2.3.1 引言

ECI主机可以通过向**ECI客户端**提供**媒质句柄**（参见第9.1.2节），来请求**ECI客户端**执行解扰会话（特定类型：在本例中为mpeg广播类型）。**ECI主机**将提供**ECI客户端**指定的、用于解扰数据的安全数据。

对于大多数传输流格式的内容进行解扰，**ECI**使用隐式时序模型来同步控制字与提供给解扰器的内容。在这个模型中，**ECI主机**为**ECI客户端**提供来自传输流的安全控制数据，因为它正在被解复用和解扰。**ECI客户端**在适当的时间提供所需的控制字（典型地，每个基本流两个，通常对所有基本流都相同）。**ECI客户端**通常将ECM解码为CW，并立即将CW加载到解扰器中。使用TS分组级或PES分组级的加扰控制位，通过内容流中的信令，这些控制字的应用实现与流的同步。

API分为以下部分：

1) 启动、重新启动和停止传输流解密（第9.6.2.3节）。
2) 安全数据获取（第9.6.2.3.5条）。
3) 广播调谐功能（第9.6.2.3.6节）。

9.6.2.3.2 传输流格式和会话版本

使用媒质会话类型**MhDvbTsBroadcast**通过**媒质句柄**解扰的传输流应符合以下规范：[ISO/IEC 13818-1-1]（特别是对TS分组应用加扰控制位）和[ETSI ETR 289]。

9.6.2.3.3 ECI主机处理要求

9.6.2.3.3.1 加扰密码检测

ECI主机应根据以下规则向**ECI客户端**发出适用的密码模式信号：

1) 对DVB流，它应使用利用PMT中加扰描述符的信令，如[ETSI TS 103 127]和[ETSI TS 100 289]中的定义。
2) 如果在1)下没有找到描述符并且来源是一个DVB广播网络，则**ECI主机**应假设按照加扰描述符定义中的规定来使用CSA1。

9.6.2.3.3.2 CA识别检测

为了建立用于加扰业务的可用DVB CA ID列表，在传输流（源自广播网络或其他）中通过TS或PES分组加扰位检测到加扰，**ECI主机**应使用以下获取规则序列：

1) 它应尝试检索服务PMT中承载的CA_descriptor。

2) 在不成功的情况下，对内容进行加扰。它应尝试检索CA标识符描述符中承载的CA_system_id集，以及适用于内容的任何DVB业务群、SDT或EIT表中所承载的。

注 – 对于某些基于传输流的内容来源，可以通过其他方式来了解适用的CA或DRM ID。

9.6.2.3.4 开始和停止传输流解密

9.6.2.3.4.1 概述

ECI主机可以使用保留的**ECI客户端**资源来在开放的**媒质句柄**上开始解密内容。**ECI主机**应提供一个“CA-PMT”表，它包含要解密的基本流的规范。表9.6.2.3.4.1-1列出了可用的解密API消息。

表9.6.2.3.4.1-1 – 媒质句柄TS内容解密API

消息	类型	方向	标签	描述
reqDcrTsDescrStart	A	H→C	0x08	**请求ECI客户端**解扰或返回TS中节目的解扰状态。
reqDcrTsDescrStop	A	H→C	0x09	**ECI主机**请求**ECI客户端**解扰**媒质句柄**。
reqDcrTsDescrQuit	A	C→H	0x0A	**ECI客户端**终止与**ECI主机**的解扰会话。

9.6.2.3.4.2 reqDcrTsDescrStart消息

H→C reqDcrTsDescrStart(ushort**mH,** uint**caPmtLen,** byte **caPmt[]) →**
C→H resDcrTsDescrStart(ushort**mH,** unit**sizeofEsStat,** descrStat**esStat[])**

- 该消息允许**ECI客户端**开始解密由**mH**标识之流上的**caPmt**所定义的节目，或者询问这样做所需的能力或条件。

请求参数定义：

mH: ushort	TS流的**媒质句柄**。
caPmtLen: uint	**caPmt**参数的长度（以字节为单位）。
caPmt: 字节[]	ca_pmt对象按照网络字节顺序在[ETSI TR 101 202]第8.4.3节中进行定义，具有表9.6.2.3.4.2-1中定义的ca_pmt_list_management和ca_pmt_cmd_id参数的修改解释。

ca_pmt_list_management参数值和语义应符合表9.6.2.3.4.2-1中的定义。

表9.6.2.3.4.2-1 – ca_pmt_list_management值

名称	值	描述
DcrTsDescrStartOnly	0x03	一个节目应该在服务中进行解扰。这可以是一个新的或经更新的值。
DcrTsDescrStartUpdate	0x05	与**DcrTsDescrStartOnly**意思相同。
RFU	其他	保留以供未来使用。

ca_pmt_cmd_id参数值应与[CEN EN 50221]第8.4.3节相同，但有以下限制：

1) 值0x02（ok_mmi）是不允许的。

2) 值0x01（ok_scrambling）和0x03（query）不得出现在相同的ca_pmt结构中，即**请求**应是纯粹的查询或纯粹的解扰请求。

响应参数定义：

mH: ushort	TS流的**媒质句柄**。
sizeofEsStat: uint	esStat参数的字节数。
esStat: descrStat	基本流的解扰状态在**请求**的**caPmt**参数中指定。descrStat在表9.6.2.3.4.2-2中进行定义。一个**descrStat.pid**值只能在**esStat**中出现一次。[CEN EN 50221]的ca_pmt结构的每个elementary_PID参数都应只出现一次，除非它对应的ca_pmd_cmd_id是0x04（not_selected），在这种情况下它不得出现在esStat中。

表9.6.2.3.4.2-2 – descrStat结构的类型定义

```
typedef struct descrStat {
   ushort pid;
   uchar     caStatus
} descrStat;
```

pid:ushort	要被解扰的流的PID值。
caStatus: uchar	值应对应于[CEN EN 50221]第8.4.3节中ca_pmt_reply对象的CA_enable参数的定义。

详细的语义：

1) 在要解码的基本流集必须改变的情况下，**ECI主机**应发出此命令。

2) 在媒质会话停止的情况下，**ECI主机**应发出一个**reqDcrTsDescrEnd请求**。如果不这样做，可能会误导**ECI客户端**注册**用户**正在进行的内容消费和相关费用。

3) 表9.6.2.3.4.2-3中定义了相关的错误代码。

先决条件请求：

1) **mH**是打开的，并具有TS格式。

后置条件请求：

2) **ECI客户端**可以开始解扰操作，并使用其他与mHTS相关的功能。

表9.6.2.3.4.2-3 – reqDcrTsStart错误代码

名称	描述
ErrDcrUserDelay	参见表9.6.2.3.7-1。
ErrDcrCardMissing	
ErrDcrServiceMissing	
ErrDcrResourceMissing	
ErrDcrMmiMissing	

9.6.2.3.4.3 reqDcrTsDescrStop消息

H→C reqDcrTsDescrStop(ushort**mH) →**
C→H resDcrDescrStop(ushort**mH)**

- 该消息允许**ECI主机**指示**ECI客户端**应停止与当前**mH**相关的TS解扰操作。

请求参数定义：

mH: ushort	TS流的**媒质句柄**。

响应参数定义：

mH: ushort	TS流的**媒质句柄**。

先决条件响应：

1） 终止与解扰mH相关的任何**ECI客户端**操作。

9.6.2.3.4.4 reqDcrTsDescrQuit消息

C→H reqDcrTsDescrQuit(ushort**mH, ushort**reason**) →**
H→C resDcrDescrQuit(ushort**mH)**

- 该消息允许**ECI客户端**通知**ECI主机**它已停止处理与当前**mH**有关的TS解扰操作的密钥。

请求参数定义：

mH: ushort	TS流的**媒质句柄**。
reason: ushort	表9.7.2.5.9-1中定义了**ECI客户端**终止解扰操作密钥处理的原因。

响应参数定义：

mH: ushort	TS流的**媒质句柄**。

先决条件响应：

1) 与解扰**mH**相关的所有**ECI主机**活动已终止或返回一个错误。

后置条件响应：

2) 与**mH**相关的所有**ECI客户端**活动应立即终止或返回一个错误。

表9.6.2.3.4.4-1 – reqDcrTsDescrQuit错误代码

名称	描述
ErrDcrDescrContinue	参见表9.6.2.3.7-1。

9.6.2.3.5 TS中ECI客户端解密数据获取

9.6.2.3.5.1 概述

ECI客户端可以以与**媒质句柄**相关联的传输流区段的形式，来获取解密所需的带内TS数据。最直接的形式是设置区段过滤器。为了加速获取信道变动，它可以设置一个包含PMT和ECM流的默认区段过滤器。它还可以从**ECI主机**读取其他标准MPEG和DVB表。MPEG区段是[ISO/IEC 13818-1-1]第2.4.4.1节private_section()结构中定义的数据结构。这部分MPEG TS API的功能在表9.6.2.3.5.1-1中列出。

表9.6.2.3.5.1-1 – ECI主机TS解扰控制消息

消息	类型	方向	标签	描述
setDcrTsSectionAcqDefault	设置	C→H	0x10	设置区段获取的默认过滤器。
setDcrTsSectionAcq	设置	C→H	0x11	设置区段获取的过滤器。
reqDcrTsSection	A	H→C	0x12	将获取的区段转交给**ECI客户端**。
reqDcrTsTable	A	C→H	0x13	**ECI客户端**获取流中的表。

9.6.2.3.5.2 区段过滤器规范

在[ISO/IEC 13818-1-1]第2.4.4.11节中定义MPEG段，可以从**ECI客户端**到传输流中的**ECI主机**提取规范。**ECI主机**应支持针对**ECI客户端**的八区段过滤器。区段过滤器设置允许**ECI客户端**用有限数量的间接说明符（例如，用于PMT），从TS流中的一个PID进行过滤。它允许**ECI客户端**设置正过滤器（选定的区段字段符合**ECI客户端**的规范）和负过滤器（区段数据与**ECI客户端**的过滤器规格不同）。过滤后的区段可以聚集，并在达到最大缓冲区大小时予以发送，或者在获取后立即予以转发。

段字节的过滤应跳过段的第两个和第三个字节。

表9.6.2.3.5.2-1给出了区段过滤器的规范。

表9.6.2.3.5.2-1 – DcrSectionFilterSpecstructure#define DcrSectionFilterMaxlen 16的类型定义

```
#define DcrSectionFilterMaxlen 16
typedef struct dcrSectionFilterSpec {
   ushort    pid;
   ushort caId;
   ushort bufferSize;
   uint      timeout;
   uint      modeFlags;
   byte      filter[DcrSectionFilterMaxlen];
   byte      mask[DcrSectionFilterMaxlen];
   byte      neg[DcrSectionFilterMaxlen];
} dcrSectionFilterSpec;
```

语义如下：

pid:ushort	要过滤的TS分组的PID。PID值应由其无符号的13位值来表示，即在0x0000与0x1FFF之间。要获取的流PMT的PID由0x8000来表示。0x8001表示要获取的关联ECM流的PID。
caId: ushort	该字段仅在**pid**字段的值为0x8001时才有用。在这种情况下，该字段的值是条件接入系统的MPEG/DVB CA ID，对此，应获取ECM流。**ECI主机**应解析要解扰的业务的PMT，并将**caId**字段与适用于视频PID（如果存在的话）的CA_descriptor（如[ISO/IEC 13818-1-1]中所定义）或PMT中的第一个ES相匹配，并使用匹配描述符中的CA-PID字段来标识要获取和过滤的ECM流。
bufferSize: ushort	缓冲区的最大尺寸。至少应能缓冲一个区段。通过将该字段设置为零，每个区段都将被单独转发。
timeout: uint	用于过滤单个区段的超时时间（以毫秒为单位）。在每个成功得到过滤的区段重新开始。值为0意味着没有超时。
modeFlags: uint	当位0被设置时，**ECI主机**应阻止两次向**ECI客户端**发送相同的启动。**ECI主机**应使用之前获取的最大区段作为缓冲区。64 kB用于此目的。所有其他位都被保留，并由**ECI客户端**设置为0。
filter: 字节[]	匹配相应区段字节的值。
mask: 字节[]	如果一个位被设置为0，则忽略对应该区段值的匹配。
neg:字节[]	如果一个位被设置为1，则对应该区段位的匹配是否定的。

如果所有正过滤的屏蔽区段位均匹配其相应的过滤器值并且没有负过滤的屏蔽区段位匹配其相应的过滤器值（假设存在至少一个负过滤位），则区段匹配过滤器。区段匹配（由区区段字节1和字节3-18的**数据**表示）通过**sectionFilterMatch**函数进行定义。

```
bool sectionFilterMatch(byte *data, *filter, *mask, *neg) {
    int i;
    bool posMatch, negMatch;

    posMatch = True;
    negMatch = True;

    /* if all neg bytes are 0; the negative filter is always fulfilled */
    for (i=0; i<DcrSectionFilterMaxlen; i++)
        negMatch&&= neg[i] == 0;

    /* match section data to positive and negative filtering criteria*/
    for (i=0; i<DcrSectionFilterMaxlen; i++) {
        posMatch&&= (data[i] & mask[i] & ~neg[i]) == (filter[i] & mask[i] & ~neg[i]);
        negMatch ||= (data[i] & mask[i] & neg[i]) != (filter[i] & mask[i] & neg[i]);
    }
    return posMatch&&negMatch;
}
```

9.6.2.3.5.3 reqDcrTsSectionAcqDefault消息

C→H setDcrTsSectionAcqDefault(ushort**mH,** uchar**filterNr,** dcrSectionFilterSpec**sectionFilter)**

- 该消息设置默认的区段过滤器，在接收到resDcrTsDescrStart消息后，**ECI主机**将使用该过滤器从**ECI客户端**的流中获取信息。例如，**ECI客户端**可以使用此功能来加快**ECI主机**在信道更换期间ECM获取区段的速度。

请求参数定义：

mH: ushort	要在其上设置默认区段过滤器的TS流的**媒质句柄**。
filterNr: uchar	要编程的过滤器的编号。该值应介于0与7之间。
sectionFilter:dcrSectionFilterSpec	根据第9.6.2.3.5.2节dcrSectionFilterSpec的区段过滤器规范。

后置条件：

- 在接收到一个成功的resDcrTsDescrStart后，该区段过滤器应由**ECI主机**来使之立即生效。如果合理的可能，则**ECI主机**应该预见一个成功的**resDcrTsDescrStart**。

9.6.2.3.5.4 reqDcrTsSectionAcq消息

C→H setDcrTsSectionAcq(ushort**mH,** uchar**filterNr,** dcrSectionFilterSpec**sectionFilter**

- 该消息设置区段过滤器，**ECI主机**将用之自EH客户端的**mH**流处获取信息。

请求参数定义：

mH: ushort	要在其上设置默认区段过滤器的TS流的**媒质句柄**。
filterNr: uchar	要编程的过滤器编号。该值应介于0和7之间。
sectionFilter:dcrSectionFilterSpec	根据第9.6.2.3.5.2节dcrSectionFilterSpec的区段过滤器规范。

详细的语义：

- 设置默认区段过滤器后使用该消息将修改区段过滤器，直到下一个resDcrTsDescrStart在同一个**媒质句柄**上发出，该**媒质句柄**将把它重置为默认区段过滤器（如果设置了默认值的话）。

后置条件设置：

- 该区段过滤器应由**ECI主机**来使之立即生效。

9.6.2.3.5.5 reqDcrTsSection消息

H→C reqDcrTsSection(ushort**mH,** uchar**filterNr,** uint**sectionDataLen**, byte **sectionData[]) →**
C→H resDcrTsSectionAcq (ushort**mH,** uchar**filterNr)**

- 在由**mH**标识的TS流和由**filterNr**标识的过滤器的情形下，该消息向**ECI客户端**发送一个或多个由**ECI主机**获取的段。
- 表9.6.2.3.5.5-1中定义了相关的错误代码。

请求参数定义：

mH: ushort	要在其上设置默认区段过滤器的TS流的**媒质句柄**。
filterNr: uchar	要编程的过滤器编号。该值应介于0和7之间。
sectionDataLen: uint	**sectionData**中的字节数。
sectionData: 字节[]	private_section序列（按网络顺序的字节）在[ISO/IEC 13818-1-1]第2.4.4.11节中定义。任何有CRC错误的区段都不会被传递给**ECI客户端**。

响应参数定义：

mH: ushort	TS流的**媒质句柄**。
filterNr: uchar	编程过滤器的编号。

先决条件请求：

1) **ECI主机**应根据区段过滤器规格或过滤器超时期限来获取区段。
2) 先前的reqDcrTsSection消息已通过resDcrTsSection进行确认。

后置条件响应：

1) 来自相同过滤器的下一个**reqDcrTsSection**消息可以通过**ECI主机**进行发送。

表9.6.2.3.5.5-1：reqDcrTsSection错误代码

名称	描述
ErrDcrTsSectionTimeout	参见表9.6.2.3.7-1。
ErrDcrTsSectionCrcErr	

9.6.2.3.5.6 reqDcrTsTable消息

C→H reqDcrTsTable(ushort**mH,** uchar**tableId,** uint **timeout,** uint**maxLen)**
H→C resDcrTsTable(ushort**mH,** uint**tableDataLen,** byte**tableData[])**

- 该消息请求**ECI主机**发送构成适用于在**mH**上解扰之节目的标准表或子表的区段。

请求参数定义：

mH: ushort	要在其上设置默认区段过滤器的TS流的**媒质句柄**。
tableId: uchar	要编程的过滤器的编号。表9.6.2.3.5.6-1列出了有效值。
timeout: uint	超时时间，以毫秒为单位。值0意味着没有超时。
maxLen: uint	要返回的最大区段数据字节数。**ECI主机**应在此限制范围内向下舍入到最大数量的区段。

表9.6.2.3.5.6-1：ca_pmt_list_management值

名称	值	描述
DcrTsTableMpegPat	0x0000	符合[ISO/IEC 13818-1-1]的PAT表。
DcrTsTableMpegCat	0x0001	符合[ISO/IEC 13818-1-1]的CAT表。
DcrTsTableMpegPmt	0x0002	符合[ISO/IEC 13818-1-1]的、所选节目的PMT表格。在应用程序使用复合PMT的情况下，结果为空。
DcrTsTableDvbNit	0x0140	实际交付网络的NIT表，如[ETSI EN 300 468]和[ETSI TS 101 211]中所规定的那样。在使用$NIT_{其他}$承载与这种网络区域相关联的表的电缆网络上，应指定适用于**CPE**区域的、适用的$NIT_{其他}$表。
DcrTsTableDvbSdt	0x0142	SDTactual_current表，如[ETSI EN 300 468]和[ETSI TS 101 211]中所规定的那样。
DcrTsTableDvbBat	0x014A	为由**ECI主机**与/或其应用程序积极使用的业务群，在[ETSI EN 300 468]中规定的BATactual表。
DcrTsTableDvbEitPf	0x014E	EITactual当前和后续的表，如[EITI EN 300 468]和[ETSI TS 101 211]所规定的那样。
DcrTsDescrStartUpdate	0x05	与**DcrTsDescrStartOnly**意思相同。

响应参数定义：

mH: ushort	TS流的**媒质句柄**。
tableDataLen: uint	tableData中的字节数。
tableData: 字节[]	代表（子）表的private_sections序列（网络顺序中的字节）在[ISO/IEC 13818-1-1]第2.4.4.11节中定义。

详细的语义：

- ECI主机应使用区段过滤器来获取ECI客户端可能请求的、所有表的新数据（以及用于其他目的）。表区段应由ECI主机发送一次。如果ECI主机仍需获取所请求的表，则它将停止**响应**。该表应为“最新”，并使用**ECI主机**可用的最新完整数据。错误代码在表9.6.2.3.5.6-2中定义。

注 – 将来随时可以通过流中的下一版本来取代一个表。

– [b-ITU-T J Suppl. 7]中提出了更新相关DVB SI表的最小重复率。
– PAT、CAT和PMT：数据超过20秒。

表9.6.2.3.5.6-2 – reqDcrTsTable错误代码

名称	描述
ErrDcrTsSectionTimeout	**参见表9.6.2.3.7-1。**
ErrDcrTsSectionCrcErr	

9.6.2.3.6 ECI客户端源控制

9.6.2.3.6.1 概述

ECI客户端能够读取传输流来源的类型，控制（重定向）传输流来源，并重定向由**ECI主机**解码的节目与/或组件。这些消息列在表9.6.2.3.6.1-1中。

表9.6.2.3.6.1-1 – TS客户端源控制API消息

消息	类型	方向	标签	描述
getDcrTsSource	get	C→H	0x18	**ECI客户端**获得TS的来源。
reqDcrTsRelocate	A	C→H	0x19	**ECI客户端**重新安置TS的来源。
reqDcrTsSelectPrg	A	C→H	0x1A	**ECI客户端**通过节目号在TS中选择节目。
reqDcrTsSelectPmt	A	C→H	0x1B	**ECI客户端**通过PMT在TS中选择节目。
reqDcrTsSelectCancel	A	C→H	0x1C	**ECI客户端**取消其之前的节目选择。

9.6.2.3.6.2 getDcrTsSource消息

C→H tsSourceType**getDcrTsSource(**ushort**mH**)

- 该消息根据网络类型和网络中的定位符来返回**媒质句柄**的来源类型。

参数定义：

mH: ushort	TS流的**媒质句柄**，以获取调谐流的类型和位置。

属性定义：

属性定义参见表9.6.2.3.6.2-1。

表9.6.2.3.6.2-1 – tsSourceType结构的类型定义

```
#define MaxTsSourceDescr 254

typedef struct tsSourceType{
    ushort tsSourceTag ;
    byte   tsSourceDescr[MaxTsSourceDescr] ;
} tsSourceType ;
```

tsSourceTag: ushort	TS来源的类型。定义的值在下面列出，包括**tsSourceDescr**的相应含义。
tsSourceDescr:字节[MaxTsSourceDescr]	含义取决于表9.6.2.3.6.2-2中列出的tsSourceTag。

表9.6.2.3.6.2-2 – tsSource标签的含义

名称	值	描述
tsSourceDvbTuner	0x0001	TS来源是DVB调谐器。tsSourceDescr以网络字节顺序包含来自表9.6.2.3.6.2-3中的单个描述符。
tsSourceDvbFile	0x0002	TS来源是一个文件或其他不可调整的资产，如IP网络（参见[b-ETSI TS 102 034]）。未定义tsSourceDescr字段。
tsDvbDuplet	0x8003	可以使用当前网络中的原始网络ID和传输流ID来找到TS来源。tsSourceDescr应包含struct dvbDuplet {ushortonid; ushorttsid}的网络字节顺序； 该值不会由getDcrTsSource消息来返回（它将返回tsSourceDvbTuner），但可以用在reqDcrTsRelocate消息中。
RFU	其他	保留以供未来使用。

高于0x7FFF的值不是绝对定位符，不应由**getDcrTsSource**返回。

表9.6.2.3.6.2-3 – DVB调谐器源描述符

DVB交付描述符名称	DVB描述符标签值
terrestrial_delivery_system_descriptor	0x5A
T2_delivery_system_descriptor	0x7F, 0x04
satellite_delivery_system_descriptor	0x43
S2_delivery_system_descriptor	0x79
cable_delivery_system_descriptor	0x44
C2_delivery_system_descriptor	0x7F, 0x0D

描述符应按[ETSI EN 300 468]中的定义使用，并应包含一个目的地频率。

9.6.2.3.6.3 reqDcrTsRelocate消息

C→H reqDcrTsRelocate(ushort**mH**, tsSourceType**tsLoc**) →
H→C resDcrTsRelocate(ushort**mH)**

- 该消息请求**ECI主机**将TS来源重新定位到**tsLoc**。表9.6.2.3.6.3-1中定义了相关的错误代码。

请求参数定义：

mH: ushort	要重新定位/重新调整的TS流**媒质句柄**。
tsLoc: tsSourceType	表9.6.2.3.6.2-1定义了重新定位流的位置。

响应参数定义：

mH: ushort	已重新定位的TS流**媒质句柄**。

详细的语义：

- 如果需要另一个网络访问资源（例如用于广播的调谐器/解调器）而不是当前分配给**媒质句柄**的，则由于资源限制的原因，该**请求**可能不会被**ECI主机**授予。
- 在成功进行重新调谐后，任何现有的滤波与/或解扰都将终止。一旦获得TS，默认获取将开始。

表9.6.2.3.6.3-1 – reqDcrTsRelocate错误代码

名称	描述
ErrDcrTsNetworkAccessCapability	参见表9.6.2.3.7-1。
ErrDcrTsNetworkAccessResource	
ErrDcrTsNetworkAccessFail	

9.6.2.3.6.4 reqDcrTsSelectPrg消息

C→H reqDcrTsSelectPrg(ushort**mH,** ushort**prgNumber) →**
H→C resDcrTsSelectPrg(ushort**mH)**

- 该消息将当前TS中的**ECI主机**的解扰节目选择设置为**prgNumber**。

请求参数定义：

mH: ushort	TS流的**媒质句柄**。
prgNumber: ushort	定义**ECI主机**待选服务的TS中的MPEG PAT和PMT表中的节目编号（参见[ISO/IEC 13818-1-1]）。

响应参数定义：

mH: ushort	TS流的**媒质句柄**。

详细的语义：

- **ECI主机**应将PAT定位在由**mH**指示的TS中。它应通过匹配**prgNumber**和program_number来定位PMT的PID。它应从定位的PID中获取PMT，并使用常规**ECI主机**功能来选择要渲染的节目组件。如果成功完成，则**ECI主机**应发出一个**reqDcrTsDescrStart请求**，以开始节目解扰。

后置条件请求：

1) 如果ECI主机解扰未由reqDcrTsSelectPrg或reqDcrTsSelectPmt请求选择的节目，则应存储节目选择参数，以便稍后可以在reqDcrTsSelectCancel上返回给节目。

后置条件响应：

2) 如果没有返回任何错误，则ECI主机随后将发送reqDcrTsDescrStart。

表9.6.2.3.6.4-1-1给出了该API消息的错误代码。

表9.6.2.3.6.4-1：reqDcrTsSelectPrg错误代码

名称	描述
ErrDcrTsPrgNumberNotInPsi	参见表9.6.2.3.7-1。
ErrDcrTsComponentSelectError	

9.6.2.3.6.5 reqDcrTsSelectPmt消息

C→H reqDcrTsSelectPmt(ushort**mH,** uint**pmtLen,** byte**pmt[]) →**
H→C resDcrTsSelectPmt(ushort**mH)**

- 该消息通过在由**mH**标识的传输流中发送定义节目组件的MPEG PMT表，来选择将由**ECI主机**解扰的新节目。

请求参数定义：

mH: ushort	TS流的**媒质句柄**。
pmtLen: uint	**pmt**参数的字节数。
pmt: 字节	根据[ISO/IEC 13818-1-1]，包含一个PMT表的private_section。

响应参数定义：

mH: ushort	TS流的**媒质句柄**。

详细的语义：

- 此命令允许**ECI客户端**选择没有适当PAT和PMT表的、TS中的组件。**ECI主机**应使用**pmt**来选择要渲染的节目组件。如果成功完成，则**ECI主机**应发出一个**reqDcrTsDescrStart请求**，以**开始**解扰节目。

后置条件请求：

1) 如果ECI主机正在解扰一个未由reqDcrTsSelectPrg或reqDcrTsSelectPmt请求选择的节目，则应存储节目选择参数，以便稍后可以在reqDcrTsSelectCancel上返回节目。

后置条件响应：

2) 如果没有返回任何错误，则ECI主机随后将发送reqDcrTsDescrStart。

表9.6.2.3.6.5-1给出了该API消息的错误代码。

表9.6.2.3.6.5-1 – reqDcrTsSelectPmt错误代码

名称	描述
ErrDcrTsComponentSelectError	参见表9.6.2.3.7-1。

9.6.2.3.6.6 reqDcrTsSelectCancel消息

C→H reqDcrTsSelectCancel(ushort**mH) →**
H→C resDcrTsSelectCancel(ushort**mH)**

- 该消息通过ECI客户端来取消先前的reqDcrTsSelectPrg或reqDcrTsSelectPmt，返回给由mH标识的TS中通过ECI主机选择的原始节目。

请求参数定义：

mH: ushort	TS流的**媒质句柄**。

响应参数定义：

mH: ushort	TS流的**媒质句柄**。

后置条件响应：

1) ECI主机随后可发送reqDcrTsDescrStart，以恢复对原始节目的解扰。

9.6.2.3.7 用于TS媒质的媒质会话API的错误代码

这些值指的是API特定的错误，它们可由该API的**响应**消息返回，列于表9.6.2.3.7-1中。

所有TS特定的**媒质句柄**请求都会返回一个有关**媒质句柄**参数的错误代码，以防将其用于非TS**媒质句柄**。

表9.6.2.3.7-1 – TS媒质的媒质会话API的错误代码

名称	值	描述
ErrDcrTsUserDelay	-256	需要很长的延时来等待完成操作所需的、来自**用户**的输入。操作未完成。
ErrDcrTsCardMissing	-257	会话所需的**智能卡**不可访问/可用。
ErrDcrTsServiceMissing	-258	在解密操作中支持**ECI客户端**所需的**CPE**外部服务不可用。
ErrDcrTsResourceMissing	-259	访问或解密内容所需的**CPE**内部未定义资源不可用。
ErrDcrTsMmiMissing	-260	**ECI客户端**无法访问MMI。
ErrDcrDescrContinue	-261	**ECI主机**继续尝试对此TS中的内容进行解扰。
ErrDcrTsSectionTimeout	-262	获取区段出现超时。
ErrDcrTsSectionCrcErr	-263	在超时期限内，检索到区段，但CRC错误。通常这意味着流被严重损坏。
ErrDcrTsNetworkAccessCapability	-264	ECI主机没有用于定位所请求TS的网络访问资源。
ErrDcrTsNetworkAccessResource	-265	ECI主机无法获取访问所请求TS的网络访问资源。
ErrDcrTsNetworkAccessFail	-266	网络访问资源未能（可靠地）获取请求的TS。
ErrDcrTsPrgNumberNotInPsi	-267	具有相应节目编号的PMT无法从PAT来定位。
ErrDcrTsComponentSelectError	-268	无法选择PMT中的组件进行解复用/解扰。
ErrDcrTsPidNotDescrambled	-269	**ECI主机**未选择Pid进行解扰。
ErrDcrTsCwIdNotValid	-270	引用了一个无效的控制字ID。
RFU	其他	保留以供未来使用。

9.6.2.4 解密基于文件和流的内容

9.6.2.4.1 引言

本节定义了一个**ECI客户端/ECI主机**API，允许**CPE**和下载的应用程序通过**ECI主机**与安全**ECI客户端**进行交互，以解密格式化为ISOBMFF [ISO/IEC 23001-9]或其他任何文件或流的内容，当中，**ECI主机**（或底层**CPE**或通过它下载的应用程序）：

- 可以从文件或流中提取所需的安全控制数据，并将之传递给**ECI客户端**；
- 允许由**ECI客户端**生成的解扰密钥通过密钥ID而正确地应用（同步）于内容。

ISOBMFF文件[ISO/IEC 23001-9]是许多非实时和自适应下载方法的通用封装格式。还有一种为这种文件格式定义的通用加密方法：CENC [ISO/IEC 23001-7]。此外，自适应流格式标准MPEG-Dash [ISO/IEC 23009-1]和[ETSI TS 103 285]是基于ISOBMFF的，不同的（有时是传统的）DRM系统使用其自身专有的ISOBMFF子格式（带有签名“品牌”标识符）。

API的一个区段允许**ECI客户端**规定它要从ISOBMFF文件取得哪些数据，以便执行此类解码，从而允许**CPE**应用程序使用ISOBMFF的专有（非CENC兼容）DRM应用程序。样本解扰的细节应该由**ECI主机**来管理：即或者是CENC兼容的，或者需要**ECI主机**中的专有扩展。

该API包含以下区段：

1) 开始和停止解扰。

2) 设置**ECI客户端**特定的安全数据获取过滤器。

3) 解密密钥（控制字）API。

9.6.2.4.2 适用规范

本节中提及的ISOBMFF文件应符合[ETSI TS 103 285]的要求。符合CENC要求的ISOBMFF文件（根据标准解密的要求）应符合[ISO/IEC 23001-7]的要求。

符合Dash要求的流数据应符合[ISO/IEC 23009-1]的要求。在适用**CPE**功能范围的情况下，执行Dash的**ECI主机**应（至少）遵守[ISO/IEC 23001-7]、[ISO/IEC 23001-9]和[ETSI TS 103 285]的要求。

9.6.2.4.3 ECI主机处理要求

9.6.2.4.3.1 解密系统识别检测

ECI主机应能够根据以下规则，从内容容器中获取适用之解密系统的列表：

1) 对于所有ISOBMFF和MP4文件，**ECI主机**应获取文件类型框（“ftyp”）和段类型框（“styp”），并使用major_brand字段和compatible_brands []字段将内容与**ECI客户端**进行匹配。

2) 对于ISOBMFF CENC编码的文件，**ECI主机**应从任何可能的位置恢复保护系统专用标题框（“pssh”）（参见[ISO/IEC 23001-7]），并从SystemID字段收集DRM系统的UUID（适合于解密内容）。这些文件可以通过包含Scheme_Type框（“schm”）的保护方案信息框（“sinf”）来认可，其中scheme_type字段等于“cenc”或“cbc1”，并且`scheme_version`字段的主版本设置为0x0001。“sinf”框的定义和位置在[ISO/IEC 23001-7]中规定。

3) 对于MPEG-Dash内容，**ECI主机**应获取MPD中的所有ContentProtection描述符，它包含@SchemeIdUri属性的特定UUID（以“urn:ID:xxxxx”开始，xxxxx为UUID），以便与**ECI客户端**DRM UUID进行匹配，或者包含@value属性中符合[ETSI TS 103 285]的条件接入系统ID（关于该通用标识符的定义，请参见[b-DASH-IF ID]）。**ECI主机**应获取所有ContentProtection描述符，以匹配**ECI客户端**能力。它应将其中包含的任何PSSH框转换为相应的ISOBMFF二进制表示。

第9.6.2.4.5.2.1节描述了将内容与**ECI客户端**匹配的过程。

9.6.2.4.3.2 加扰类型检测

ECI主机应根据以下规则向**ECI客户端**指示适用的解扰模式：

1) 对于ISOBMFF CENC编码的文件，它应能够应用[ISO/IEC 23001-7]中定义的规则，来检测密码（AES-CTR或AES-CBC），包括清除/加扰字节选择、填充和初始化，向量提取和应用在[ISO/IEC 23001-7]中定义。

2) 对于ISOBMFF格式的MPEG DASH内容，应按照[ETSI TS 103 285]中的定义，应用AES-CTR（带密钥旋转）来解扰。

9.6.2.4.3.3 默认内容容器安全数据过滤

ECI主机应在与解扰过程相关的时间，传递指定用于**ECI客户端**的容器中包含（不透明的）信息的任何框。这特别适用于ISOBMFF CENC编码文件中的以下框以及ISOBMFF格式的Dash内容：

1) 对于：

a) “moov”和“moof”框中的保护系统特定标题框与**ECI客户端**DRM系统ID的UUID相匹配，与当前或不远的未来正在进行的解码有关。

b) 保护方案信息框“sinf” – 在**ECI客户端**需要访问“sinf”框的情况下。

9.6.2.4.3.4 内容的解扰

ECI主机负责解释加扰模式，确定要解扰的数据，并利用适当的密钥ID，使用解扰器来确定**ECI客户端**提供的密钥。

为了让**ECI客户端**计算相关的密钥，**ECI主机**应及时将所需的安全控制数据从内容容器传递给**ECI客户端**。

9.6.2.4.4 基于文件和流媒质的媒质会话API

9.6.2.4.4.1 概述

ECI主机可以使用保留的**ECI客户端**资源，在开放的**媒质句柄**上启动解密内容。**ECI主机**应为**ECI客户端**提供初始化数据，以开始评估访问权限。

表9.6.2.4.4.1-1 – 媒质句柄TS内容解密API

消息	类型	方向	标签	描述
reqDcrFileStart	A	H→C	0x01	**请求ECI客户端**解扰或返回文件或流的解扰状态。
reqDcrFileStop	A	H→C	0x02	**ECI主机**请求**ECI客户端**停止**媒质句柄**的解扰操作的密钥处理。
reqDcrFileQuit	A	C→H	0x03	**ECI客户端**取消**ECI主机**的解扰操作。

9.6.2.4.4.2 reqDcrFileStart消息

H→C reqDcrFileStart(ushort**mH,** uchar**reqType,** uchar**dataType,** uint**initDataLen,** byte**initData[]) → C→H resDcrFileStart(**ushort**mH,** uchar**dcrStat)**

- 该消息请求**ECI客户端**返回与mH相关联的内容的解扰状态与/或开始解扰会话。根据容器/加密格式，**ECI主机**为**ECI客户端**提供初始数据，以开始任何许可获取和评估。

请求参数定义：

mH: ushort	文件的**媒质句柄**。
reqType: uchar	**请求**类型（解扰开始或者许可查询）在表9.6.2.4.4.2-1中定义。
dataType: uchar	InitData的类型。
initDataLen: uint	initData容器的长度（以字节为单位）。
initData: byte	来自dataType定义的内容的初始化数据。在表9.6.2.4.2-2中定义了initDat的编码。

表9.6.2.4.4.2-1 – reqType编码

名称	值	描述
ReqTypeDcr	0x01	解密开始；如果需要，与**用户**进行对话。
ReqTypeInq	0x02	询问解扰选项。
RFU	其他	保留以供未来使用。

表9.6.2.4.4.2-2 – initData编码

数据类型	值	Description
FmtIsoCenc	0x04	遇到与**ECI客户端**MatchSpecifier中DRM ID相匹配的ISOBMFF PSSH框（参见[ISO/IEC 23001-7]）。
FmtIsoCencDash	0x05	在MPD（参见[ISO/IEC 23007-1]）或初始化段（参见[ISO/IEC 23009-1]）中遇到的ISOBMFF PSSH框（参见ISO/IEC 23001-7）遇到**ECI客户端**MatchSpecifier中相匹配的DRM ID。
FmtIsoProp	0x06	**ECI主机**可以根据专有知识将数据传递给**ECI客户端**。**ECI客户端**应能够根据相同的、共同的专有知识来解释此数据。
FmtIsoPropDash	0x07	FmtIsoProp包括指明该数据是DASH来源的指示。
RFU	其他	保留以供未来使用。

响应参数定义：

mH: ushort	TS流的**媒质句柄**。
dcrStat: uchar	解扰状态；参见表9.6.2.4.4.2-3。

表9.6.2.4.4.2-3 – 解扰状态

名称	值	描述
DcrStatNo	0x00	没有可能的解扰（DRM系统具有解扰能力）。
DcrStatOk	0x01	解密开始；如果需要，与**用户**进行对话。
DcrStatDialog	0x02	需要与**用户**进行对话。
DcrStatPay	0x03	需要支付，也可能需要与**用户**进行对话。
DcrStatDrmNok	0xFE	DRM系统无法解密此内容。
RFU	其他	保留以供未来使用。

详细的语义：

- 在查询时，**ECI客户端**不会启动任何**用户**对话，但**ECI客户端**应通过不带**用户**对话的许可证服务器清除许可条件，来评估解扰内容的能力。

先决条件请求：

1) **媒质句柄**未决。

先决条件响应：

2) 如果ECI客户端可以对内容进行解扰并且reqType正确，则ECI客户端将准备生成解扰密钥。

表9.6.2.4.4.2-4给出了请求启动解密消息的错误代码。

表9.6.2.4.4.2-4 – reqDcrFileStartt错误代码

名称	描述
ErrDcrFileUserDelay	参见表9.6.2.4.7-1。
ErrDcrFileCardMissing	
ErrDcrFileServiceMissing	
ErrDcrFileResourceMissing	
ErrDcrFileMmiMissing	

9.6.2.4.4.3 reqDcrFileStop消息

H→C reqDcrFileStop(ushort**mH) →**
C→H resDcrFileStop(ushort**mH)**

- 该消息允许**ECI主机**停止文件解密操作。

请求参数定义：

mH：ushort	文件的**媒质句柄**

响应参数定义：

mH：ushort	文件的**媒质句柄**

先决条件响应：

1) **ECI客户端**已终止任何与解密内容有关的操作。

9.6.2.4.4.4 reqDcrFileQuit消息

C→H reqDcrFileQuit(ushort**mH,** uint **reason) →**
H→C resDcrFileQuit(ushort**mH)**

- 该消息允许**ECI客户端**通知**ECI主机**，它已经终止文件解密操作的密钥处理。在表9.6.2.4.4.4-1中定义了相关的错误代码。

请求参数定义：

mH：ushort	TS流的**媒质句柄**。
reason：uint	表9.7.2.5.9-1中定义的值。

响应参数定义：

mH：ushort	文件的**媒质句柄**。

先决条件响应：

1） 与解扰**mH**相关的所有**ECI主机**活动已终止或返回一个错误。

后置条件响应：

2） 所有与**mH**相关的**ECI客户端**活动都应立即终止，或者返回一个错误。

表9.6.2.4.4.4-1：reqDcrFileQuitl错误代码

名称	描述
ErrDcrFileDescrContinue	参见表9.6.2.4.7-1。

9.6.2.4.5 ECI客户端特定安全数据获取

9.6.2.4.5.1 概述

ECI主机应对要解码的数据执行标准数据获取，以获取**ECI客户端**执行密钥计算所需的信息。**ECI客户端**可以指定超出**ECI主机**提供之标准数据的特定数据获取。**ECI主机**应保留有限数量的过滤器，以获取此类数据。

表9.6.2.4.5.1-1 – 数据过滤器API

reqDcrFileFilter	req	C→H	0x04	**ECI客户端**请求**ECI主机**为安全数据获取设置一个数据过滤器。
reqDcrFileData	A	C→H	0x05	**ECI客户端**请求**ECI主机**通过文件过滤器获取数据。

9.6.2.4.5.2 文件过滤器规范

9.6.2.4.5.2.1 通用文件过滤器定义

文件数据过滤器规范基于文件格式的基本规范。在定义之文件格式的上下文中，定义了一个过滤器。通用的文件过滤器规范在表9.6.2.4.5.2.1-1中定义。

表9.6.2.4.5.2.1-1 – 通用文件过滤器规范

```
typedef struct dcrFileFilterSpec {
   ushort filterType;        //  is  defined  in  Table  9.6.2.4.5.2.1-2
   ushort filterLen;
   byte   filter[filterLen]; // shall be formatted according to filterType
} dcrFileFilterSpec;
```

表9.6.2.4.5.2.1-2：文件过滤器类型

FileFilterIsobmff	0×0001	用于ISMBMFF格式化数据的文件过滤器在第9.6.2.4.5.2.2节中定义。
RFU	其他	保留以供未来使用。

9.6.2.4.5.2.2 ISOBMFF特定的文件过滤器定义

在表9.6.2.4.5.2.2-1中定义了ISOBMFF格式文件的过滤器规范。

表9.6.2.4.5.2.2-1：ISOBMFF文件过滤器规范

```
#define MaxFilterFile 16 // maximum number of bytes in box that are filtered
#define MaxContainers  4 // maximum number of container boxes for a box
#define MaxUuidLen    16 // Length in bytes of a UUID

typedef struct BoxSpec {
    uint   boxType                  // 4CC code of box type
    byte   extendedType[MaxUuidLen]
                                    // UUID for boxType=='uuid', otherwise no
significance
    byte   filter[MaxFileFilter];       // shall match bytes of box following
    byte   filterMask[MaxFilter];
    ushortdataLen ;             // maximum amount of box data to be acquired
} BoxSpec;

typedef struct dcrFileFilterIsobmff {
    BoxSpec    container[MaxContainer];
    BoxSpec    box;
} dcrFileFilterIsobmff;

bool function boxMatch
   (byte *boxData, byte *filter, byte*filterMask; int boxLen) {
{
    bool match = true;
    int i;

    for( i=0; i<MaxFilterFile&&i<boxLen&& match; i++) {
        match &&= (boxData[i] &filterMask[i] == filter &filterMask[i]) ;
    }
    return match;
}
```

ECI主机应解析该文件，并获取匹配**框**字段的各框，包含在与任何**容器**数组相匹配的框中。**ECI主机**应跳过扫描[ISO/IEC 14496-12]或[ISO/IEC 23001-7]中未定义的各框。

dcrFileFilterIsobmff容器字段中的**boxType**可以设置为“****”，来表示通配符。在这种情况下，**容器**的其他字段应没有任何意义，并设置为0，以表示不匹配。

BoxSpec中的**filter**和**filterMask**字段应与待处理框类型字段之后的第一个字节相匹配。对于“全框”（参见[ISO/IEC 14496-12]），这是版本和标志字段。匹配应根据boxMatch函数来进行，**boxLen**参数设置为框boxtype和extended_type之后的字节数，**boxData**参数指向这些字节的开始处，**过滤器**参数指向**boxSpec.filter**字段，并将**filterMask**参数设置为**boxSpec.filterMask**字段。

由过滤器返回的数据为各框（按顺序），随着**ECI主机**解析文件，它们匹配过滤器。**ECI主机**可以方便地对这些框进行聚类，但不应不必要地延迟将各框传递给**ECI客户端**，因为这可能会阻止**ECI客户端**生成所需的解扰密钥。

9.6.2.4.5.2.3 reqDcrFileFilter消息

C→H setDrcFileFilter(ushort**mH,** uchar**filterNr,** dcrFilleFilterSpec ***dataFilter)**

- 该消息请求**ECI主机**根据dataFilter设置数据过滤器，以获取**ECI客户端**的安全数据。

参数定义：

mH: ushort	TS流的**媒质句柄**。
filterNr: uchar	**ECI主机**中文件过滤器的编号。
dataFilter: dcrFilleFilterSpec *	用于数据提取的过滤器规范。

后置条件请求：

- 该区段过滤器应由ECI主机来生效，直到reqDcrFileStop或reqDcrFileQuitl生效，或者reqDcrFileFilter设置为dataFilter == NULL。

9.6.2.4.5.2.4 reqDcrFileAcqData消息

H→C reqDcrFileAcqData(ushort**mH,** uchar**filterNr,** uint**dataLen**, byte **data[]) →**
C→H resDcrFileAcqData (ushort**mH,** uchar**filterNr)**

- 该消息请求ECI主机获取并发送由mH确定的、媒质文件或流情形下的一个或多个区段，以及将由**filterNr**确定的过滤器发送给**ECI客户端**。

请求参数定义：

mH: ushort	在其上设置默认区段过滤器的文件的**媒质句柄**。
filterNr: uchar	要编程的过滤器的编号。该值应介于0与7之间。
dataLen: uint	**数据**中的字节数。
data[]**:**字节	private_section序列（按网络顺序的字节）在[ISO/IEC 13818-1-1]第2.4.4.11节中定义。任何有CRC错误的区段都不会被传递给**ECI客户端**。

响应参数定义：

mH：ushort	媒质文件或流的**媒质句柄**。
filterNr：uchar	已编程的过滤器编号。

表9.6.2.4.5.2.4-1列出了相关的错误代码。

表9.6.2.4.5.2.4-1 – reqDerFileAcqData错误代码

名称	描述
ErrDcrAcqDataTimeout	参见表9.6.2.4.7-1。
ErrDcrAcqDataDataErr	

9.6.2.4.6 文件解扰控制字API

9.6.2.4.6.1 概述

内容解扰API区段允许密钥由**ECI客户端**来解密。**ECI主机**必须首先通过将密钥ID传递给**ECI客户端**来启动控制字的可用性。一旦密钥可用，**ECI主机**就可以将计算得到的控制字应用于（加密的）内容。表9.6.2.4.6.1-1中列出了与**媒质句柄**文件内容解扰API相关的API消息。

表9.6.2.4.6.1-1 – 媒质句柄文件内容解扰API

消息	类型	方向	标签	描述
reqDcrFileKeyComp	A	H→C	0x20	启动**ECI客户端**的任何必需的计算或其他活动，以创建一个带有可用Key-ID的控制字。

9.6.2.4.6.2 ECI主机处理要求

9.6.2.4.6.2.1 ISOBMFF CENC格式的内容

本节定义了以ISOBMFF + CENC格式对内容进行解扰的**ECI主机**处理要求。

ECI主机负责及时地将任何密钥ID信息传递给**ECI客户端**，以便**ECI客户端**能及时地获得/获取所需的控制字。允许之的其他约束条件应该比控制字的预期使用至少提前30秒。

密钥ID信息包含在与媒质样本相关的几个框中（（部分）加密媒质数据的序列）：例如，参见[b-DASH-IF V3]第5.4节。这些框中的数据允许提取密钥ID、IV，并允许在媒质样本中识别清楚的和加密的数据。

9.6.2.4.6.2.2 MPEG DASH格式内容

ECI主机必须支持的MPEG DASH格式的细节目前不在**ECI**规范的讨论范围中。

9.6.2.4.6.3 reqDcrFileKeyComp消息

H→C reqDcrFileKeyComp(ushort**mh,** byte **keyId**[MaxUuidLen]**)** →
C→H resDcrFileKeyComp(ushort**mH)**

- 该消息启动**ECI客户端**所需的计算和任何其他活动，以计算KeyId标识的控制字，并使之可用于解密内容。

请求参数定义：

mH: ushort	TS流的**媒质句柄**。
keyId[MaxUuidLen]**:**字节	KeyID作为网络字节顺序的UUID。

响应参数定义：

mH: ushort	TS流的**媒质句柄**。

先决条件响应：

1) 密钥可用或出现一个错误或超时。

详细的语义：

- 如果无法及时（60秒）提供请求的控制字，则**ECI客户端**应报告一个错误。即使报告了一个错误，**ECI客户端**也可继续尝试获取所请求的密钥。
- 在报告的错误中，**ECI主机**可重发**请求**。**ECI主机**最多可发出10个**请求**。

表9.6.2.4.6.3-1列出了相关的错误代码。

表9.6.2.4.6.3-1：reqDcrFileKeyComp错误代码

名称	描述
ErrDcrFileUserDelay	参见表9.6.2.4.7-1。
ErrDcrFileCardMissing	
ErrDcrFileServiceMissing	
ErrDcrFileResourceMissing	
ErrDcrFileMmiMissing	
ErrDcrFileKeyIdUnknown	
ErrDcrFileKeyOverflow	

9.6.2.4.7 解密文件和基于流的内容API的错误代码

API特定错误的值可由该API的**响应**消息返回，列于表9.6.2.4.7-1中。

所有文件特定的**媒质句柄**请求都会返回一个有关**媒质句柄**参数的错误代码，以防将其用于非文件**媒质句柄**。

表9.6.2.4.7-1：文件和流媒质的媒质会话API的错误代码

名称	值	描述
ErrDcrFileUserDelay	-256	需要很长的延时来等待完成操作所需的、来自**用户**的输入。操作未完成。
ErrDcrFileCardMissing	-257	会话所需的智能卡不可访问/可用。
ErrDcrFileServiceMissing	-258	在解密操作中支持ECI客户端所需的CPE外部服务（例如DRM服务器）不可用。
ErrDcrFileResourceMissing	-259	访问或解密内容所需的CPE内部未定义资源不可用。
ErrDcrFileMmiMissing	-260	ECI客户端无法访问MMI。
ErrDcrFileDescrContinue	-261	ECI主机继续尝试对此文件中的内容进行解扰。
ErrDcrAcqDataTimeout	-262	获取数据出现超时。
ErrDcrAcqDataDataErr	-263	在超时期限内，检索到区段，但存在错误。通常这意味着文件被损坏或者不符合适用的规范。
ErrDcrFileKeyIdUnknown	-300	ECI客户端/安全系统对此内容的密钥ID未知。
ErrDcrFileKeyOverflow	-301	短时间内密钥ID请求过多；等待ECI客户端对先前的处理请求做出响应。
ErrDcrFileKeyWithdrawn	-302	密钥不再可用；ECI客户端撤回的权利。

9.7 用于访问ECI主机重加密资源的API

9.7.1 重加密API介绍

9.7.1.1 第9.7节中定义的API列表

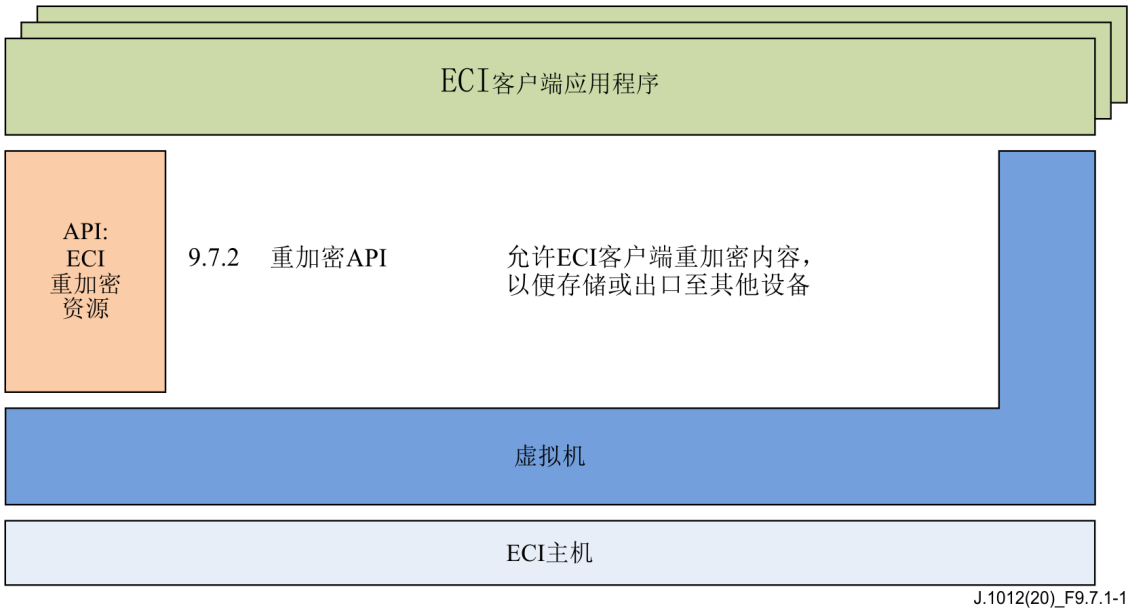


图9.7.1-1 – 第9.7节中定义的API的框图

表9.7.1-1列出了第9.7节中涵盖的API，图9.7.1-1说明了利用**ECI体系结构**在第9.7节中定义的API位置。另请参阅[b-Menezes]。

表9.7.1-1 – 第9.7节中定义的API列表

条款	API名称	描述
9.7.2.3	**出口连接**API	允许**ECI客户端**为输入的内容建立一个**出口连接**。
9.7.2.5	**入口连接**API	允许**ECI客户端**输入内容，它通过接入网络进行加密，并在**ECI客户端**的控制下进行解密。
9.7.2.6	**微客户端**解密API	允许**ECI客户端**解密输入和重加密的内容。

9.7.1.2 重加密的一般概念

ECI中的重加密允许独立的**微DRM系统**保护由CA或DRM **ECI客户端**提供的内容，以用于我们在**CPE**内部或外部的进一步应用。符合**ECI**要求的实施方案中的重加密系统称为**微DRM系统**。**微DRM系统**的应用可以是如时间转移的、PVR的和流的。重加密**ECI客户端**被称为**微服务器**。无论是**ECI**兼容的还是非**ECI**兼容的，可以解密重加密之内容的客户端称为**微客户端**。重加密的客户端图像和证书可以作为常规**ECI客户端**下载，通过微DRM主服务器来提供。图9.7.1.2-1显示了整个系统（不包括微DRM主服务器）。在本地存储的情况下，**微服务器**和**微客户端**在单个设备中实现。

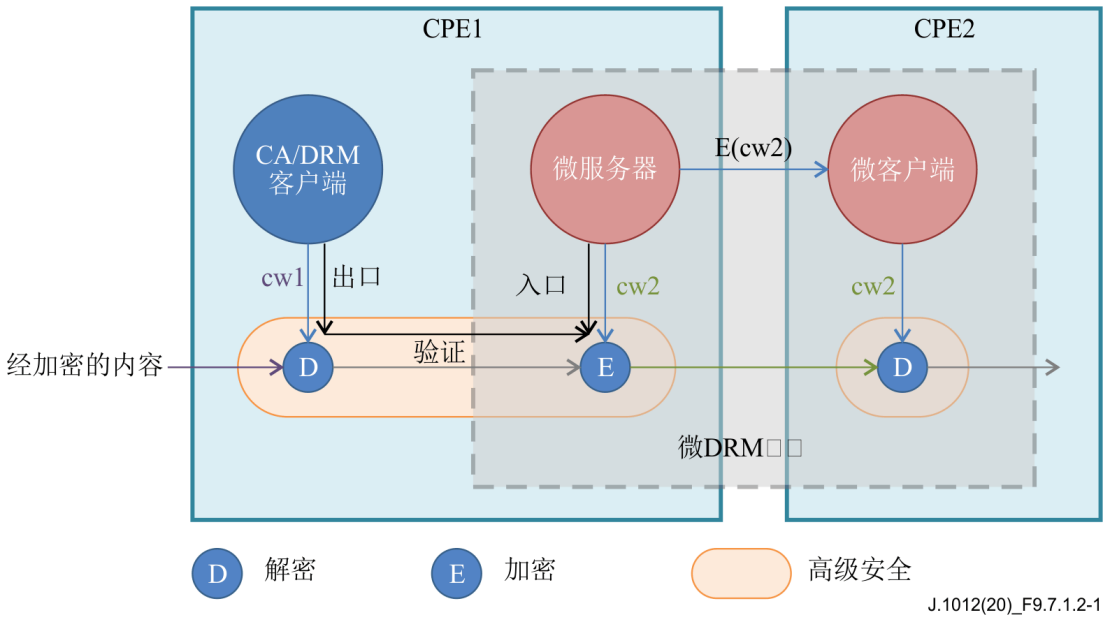


图9.7.1.2-1 – 微DRM系统图

最初解密内容的CA/DRM **ECI客户端**可以控制是否允许将内容出口到所安装的**微DRM系统**。它通过**高级安全**系统为此目的而对**微服务器**实施认证；认证受CA/DRM运营商控制。一旦出口内容，**微DRM系统**将负责保护内容。**高级安全**系统可以安全地支持解密、重加密和出口认证。原理如图9.7.1.2-1所示。

9.7.1.3 重加密API结构概述

图9.7.1.3-1做了更详细的图示，说明了重加密中涉及的不同API的作用。**ECI主机**通过解密API向解码**ECI客户端**提供所有必需的信息。解码**ECI客户端**通过高级安全API安全地建立用于内容解密的控制字。认证基本的内容属性（标记）。出口API允许**ECI主机**请求解码**ECI客户端**，以便为重加密建立到所需**微服务器**的**出口连接**。高级安全API允许出口**ECI客户端**对入口**微服务器**进行认证。**ECI主机**使用入口API来建立到**微服务器**的授权**出口连接**。重加密API允许**ECI主机**将**微服务器**引导到对应内容封装格式和应用的操作模式（流的、时间转移的或存储的），并对所需（经过认证的）目标**微客户端**的内容进行加密。

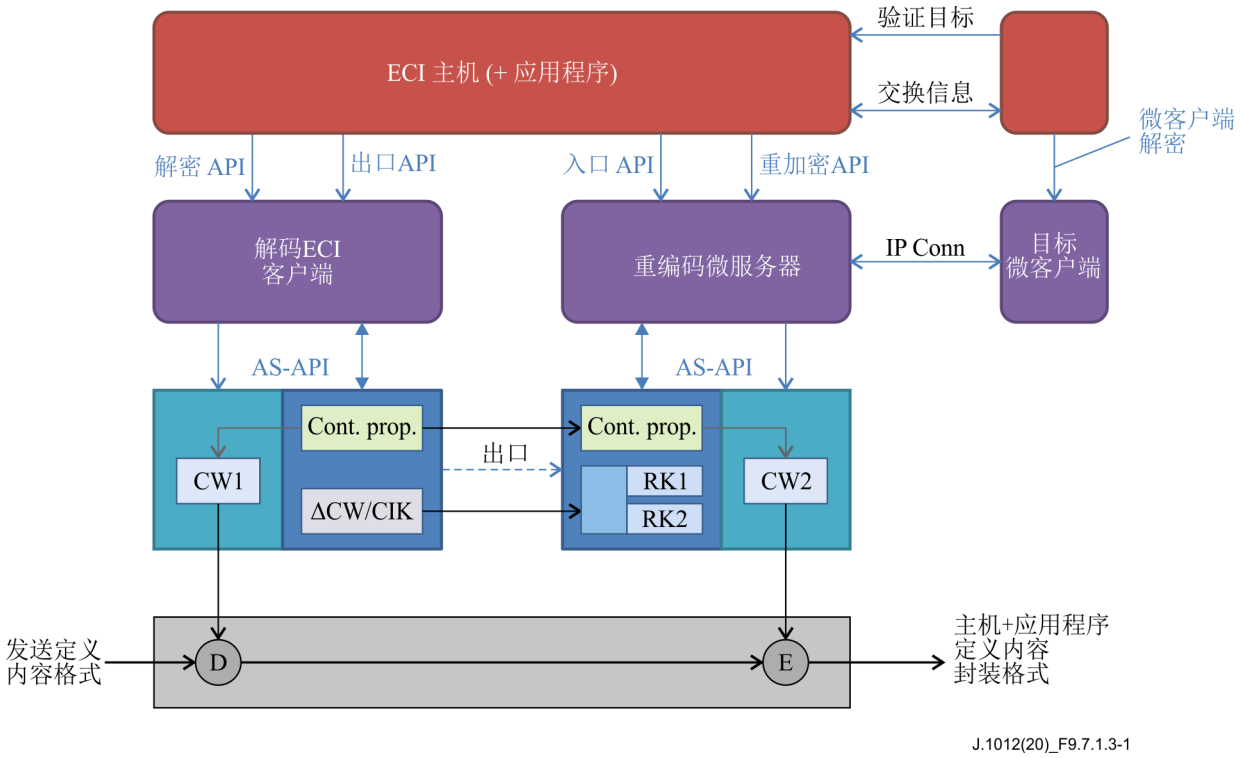


图9.7.1.3-1 – 解密和重加密功能的体系结构

图9.7.1.3-1和图9.7.1.3-2中的方案概述了解密、出口控制、入口控制、重加密和微客户端解密API中的主要消息。它显示从左到右的内容：从第一个CA/DRM传送**ECI客户端**通过**出口/入口连接**到**微服务器**，它对解密的内容进行加密，以便最终由**目标微客户端**来解码。

这四个主机-客户端API支持以下处理步骤：

- *发现阶段*使**ECI客户端**可以将其潜在的互通选项发布到**ECI主机**（与应用程序协作）。这使**ECI主机**能够将请求的内容与特定**ECI客户端**进行匹配。在所选**ECI客户端**不具备处理此内容的适当权限的情况下，**ECI主机**必须寻找其他的**ECI客户端**。在家庭网络和分布式PVR应用中，这可能涉及如DLNA等应用协议，请参见[b-DLNA]。*认证步骤*允许**ECI主机**在所需的**ECI客户端**与**微服务器**或者**微服务器**与**微客户端**之间建立一个经过认证的连接。认证可以是隐性的：即，用于认证的密码证明可体现为**ECI客户端**最终解密内容的能力。认证总是跟在内容流后。在某些情况下，需要一个反向协议。出于商业目的，**入口连接**可能必须经**微服务器**批准。
- *会话实例化步骤*允许**ECI主机**保留解密或加密**媒质句柄**相关之特定操作模式中内容所需的全部资源。入口和**目标**连接为**微服务器**上的reqEncrMhOpen而定义，或者隐含在常规CA/DRM **ECI客户端**中。请注意，**ECI主机**负责为整个媒质应用场景分配补充资源，如（解扰）加扰、解复用和解码处理资源，以便能够继续。**ECI客户端**最终使用**高级安全**API请求分配AS以及解密或加密资源。
- *会话控制步骤*允许**ECI主机**启动和停止处理**媒质句柄**上的内容。为了对路径上的内容进行无缝处理，需要从目的地到来源启动**ECI客户端**：即，如果**ECI客户端**以这种方式呈现，则它应准备好处理内容。

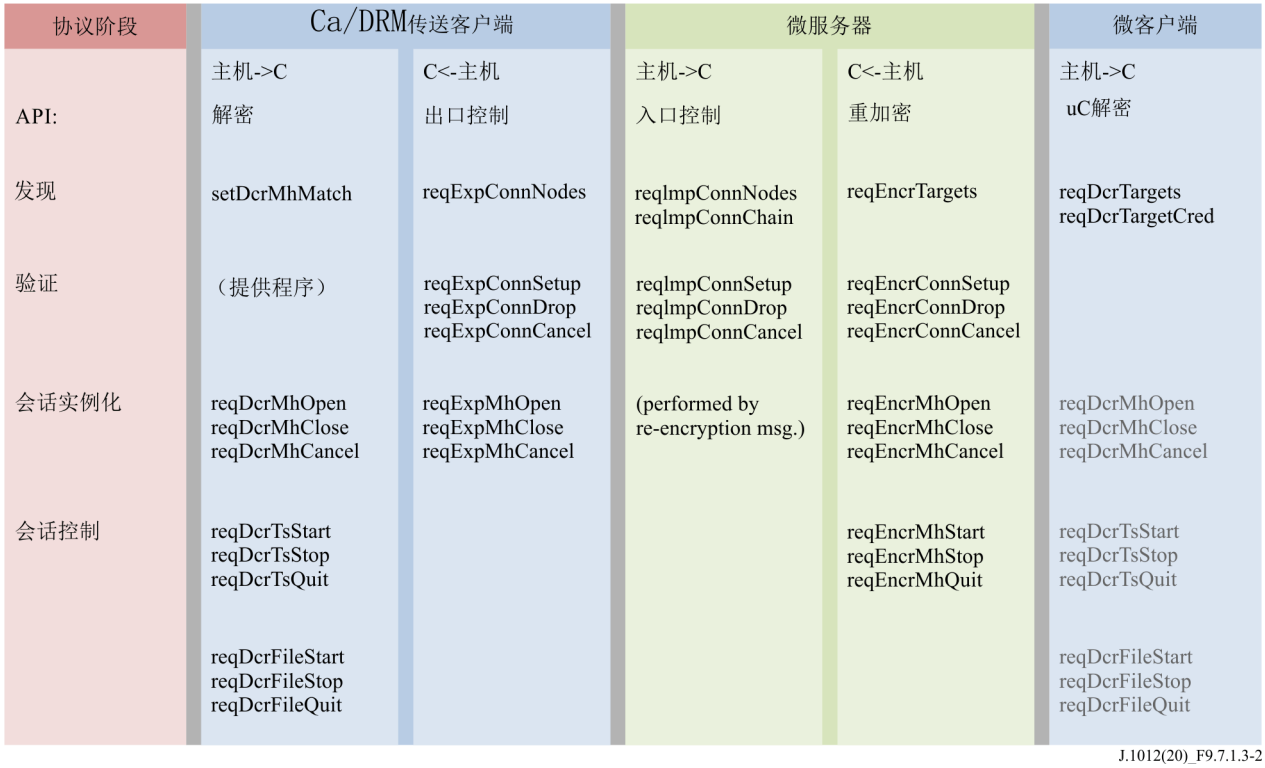


图9.7.1.3-2 – 加密/解密和入口/出口API概述

这些消息在其命名和语义中使用了某个特定的系统：

- 发现步骤允许**ECI客户端**发布其连接到另一个**ECI客户端**或内容的能力。消息setDcrMhMatch、reqExpConnNodes、reqImpConnNodes、reqEncrTargets、reqDcrTargets请求**ECI客户端**发布这些消息（以身份的形式）。
- 认证步骤使用设置、删除和取消消息来创建一个（已经认证的）连接，由**ECI客户端**解除先前的连接或取消这样的连接。连接的参考是**出口连接**（**ECI客户端**出口内容）、**入口连接**（**ECI客户端**入口内容）或**目标**连接（**微服务器**加密内容以供**目标**进行后续解密，反之亦然，例如，**微客户端**解密来自**微服务器**的内容）。
- 会话实例化步骤使用打开、关闭和取消来创建和终止会话，所有都会将**媒质句柄**称为通用参考。**ECI客户端**所需的MMI会话和**智能卡**资源管理也可以参考**媒质句柄**，以允许**ECI主机**在其应用情形中关联一个用户对话请求。
- 会话控制步骤为解密两种特定的内容格式定义了不同的消息：传输流和文件格式。处理可以由**ECI主机**来启动和停止，并且**ECI客户端**可以在缺少资源或权限问题的情况下退出处理。

注1 – 对于某些保护系统，可能不需要对所有阶段进行重要的处理。其**ECI客户端**可能只对一些消息执行少量的管理性处理。

注2 – **入口/出口连接**上的**ECI客户端**的性质与**微处理器**与**微客户端**之间的关系不同。在与**ECI客户端**的**入口/出口连接**上共享**ECI主机**，并且可以使用**ECI**定义的入口/出口证书链，通过AS出口机制来交换内容。**微处理器**与**微客户端**可以使用选择协议（**微DRM系统**的特性）来建立连接，只要它适合API框架并且可以使用**AS系统**来建立认证和公用密钥。在**出口/入口连接**上的内容交换是隐性的（通过**ECI主机**定义）；**微处理器**的真实性（用于出口目的）将通过**AS系统**来验证。**微处理器**与**微客户端**之间的内容交换需要**微处理器**与**微客户端**两者上的**媒质句柄**会话和会话控制。

9.7.2 ECI出口控制API

9.7.2.1 引言

ECI允许**ECI客户端**将解码后的内容出口到**微服务器**，以确保为了（允许）重新分配给其他设备或者（允许）存储内容以供后续播放之目的而进行重加密。为此，**ECI**定义了一个**证书**结构，用于定义允许的出口**微DRM系统**的组。解码的每个内容项都伴随着相应**出口组**的标识。从**出口组**，必须有一个**证书**链用于将出口授权给选定的**微服务器**。该链由高级安全系统处理来处理，以提供高度可靠的出口授权机制。

出口**ECI客户端**负责提供**出口组证书**和所有直接后代。入口**微客户端**负责提供补充证书信息，以便允许完成从出口到入口**ECI客户端**的链。

ECI主机可建立一个从解密**ECI客户端**到加密**微服务器**的重加密连接。一旦连接建立，**ECI主机**就可使用**媒质句柄**会话继续解密和重加密内容。**AS系统**将根据通过**AS系统**提供的证书，确保把来自解码**ECI客户端**的内容和相关的保护信息安全地传递给**微客户端**。

ECI主机为**ECI客户端**提供支持，以访问网络服务，接收用于出口和入口的最新证书，例如，通过数据轮播API（第9.5.4节）和IP HTTP API（第9.4.4.6节）。

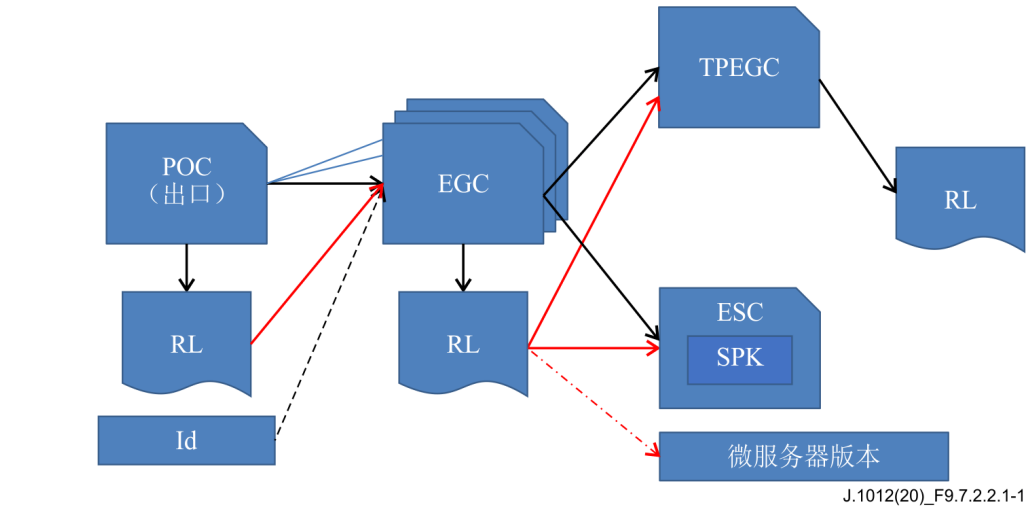
对于重加密的定位，**ECI主机**和应用程序必须建立允许解码内容的授权**微客户端**。这可以是单独的**CPE**（带有合适的客户端）以及组（基于共享的密钥）。而后**ECI主机**在**微服务器**与其匹配的**微客户端**（每个**微客户端**一个）之间建立一个经授权的连接。对于时间转移和记录应用程序，可以存储**ECI客户端**所需的信息（以便稍后能够解码内容）（例如，与重加密的内容一起）。对于实时流连接，当**微客户端**和**微服务器**驻于同一设备中时，**微服务器**与**微客户端**所需的会话控制消息可以通过**微客户端**来传递，也可以通过一个IP连接在**微客户端**之间直接进行通信。

注 – ECI客户端到ECI客户端通信的通信协议和相关的安全方面问题超出了有关ECI的讨论范围。

9.7.2.2 出口证书结构

9.7.2.2.1 总体结构

ECI出口机制基于**证书**。大多数**证书**都有关联的**撤销列表**，以允许对出口权限的更新。图9.7.2.2.1-1给出了用于解码**ECI客户端**的立即出口控制的**证书**结构。



POC: 平台操作证书
RL: 撤销清单
Id: 出口组ID（由内容权限定义）
SPK: 发送方公钥

EGC: 出口组证书
TPEGC: 第三方出口组证书
ESC: 出口系统

图9.7.2.2.1-1 – ECI证书分发结构

ECI客户端平台操作证书（POC）是**出口组**证书之**父**。**ECI** POC有一个特殊的撤销列表，允许**ECI客户端**控制**出口组**证书及其相关的撤销列表版本。每个**出口组**证书都是实际出口证书或其他（后代）**出口组**的**父**。有两种类型的出口证书：

1) 出口系统**证书**（ESC）通过其**发送方公钥**来标识允许的出口**微服务器**，以允许立即进行认证。此外，ESC的撤销版本号用于定义**微服务器**的最低版本号。

2) 第三方**出口组**证书（TPEGC）指的是由另一个组织管理的**出口组证书**。这允许使用单个出口**证书**对更大的异构**微DRM系统**组进行认证。

图9.7.2.2.1-2对第三方组出口**证书**的结构做了进一步说明。

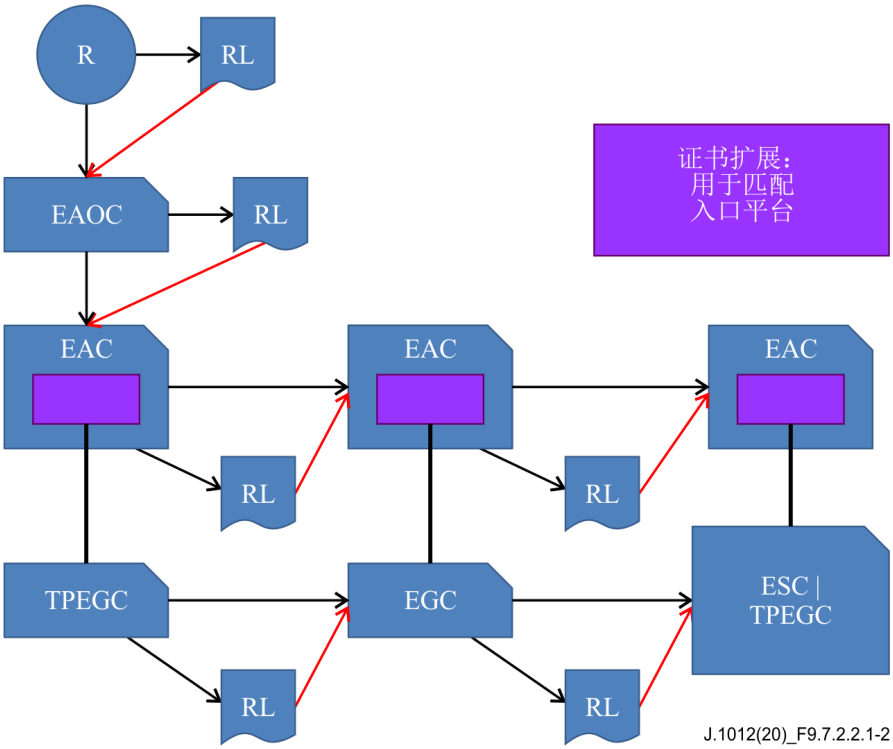


图9.7.2.2.1-2：第三方组出口证书结构

ECI根证书是出口授权运营商**证书**（EAOC）的**父**。**ECI根证书**为这些**证书**维护一个特殊的撤销列表。出口授权运营商**证书**（EAOC）是出口授权**证书**（EAC）的**父**。该**证书**与一个第三方**出口组**证书（TPEGC）相匹配。通过这种机制，第三方组进行双重认证，以提供额外的安全性。

第三方**出口组**证书是以下任何一方的**父**：

1) **出口组**证书（EGC），其本身可以是另一个EGC或任何下列**证书**的**父**。每个EGC都有一个关联的**撤销列表**。

2) 出口系统证书（ESC）。

3) （下一个）第三方**出口组**证书（TPEGC）。

每个**证书**还通过匹配的出口授权证书（EAC）进行额外验证，该证书形成一个与TPEGC/EGC树匹配的树。

表9.7.2.2.1-1提供了**证书**及其**父**的概述。

表9.7.2.2.1-1 – 不同出口证书概述

证书名称	缩写	描述	父
出口组	EGC	该**证书**允许出口**ECI客户端**来认证其允许出口的一组**微客户端**与/或第三方认证组。适用的**出口组**被定义为内容的经过认证的权限属性的一部分。	POC, TPEGC, EGC
第三方**出口组**	TPEGC	一个**证书**，用于认证由另一方（第三方）管理的一组**微DRM系统**。	EGC, TPEGC
出口授权运营商	EAOC	一个**证书**，为**运营商**提供基础，以便为第三方**出口组**提供授权服务。**证书**是其共同认证的、第三方**出口组**的出口授权**证书**树的**父**。	ECI根
出口授权	EAC	该**证书**提供由第三方管理的第三方**出口组**证书或**出口组**证书的共同认证。	EAC, EAOC
出口系统	ESC	该**证书**认证一个**微客户端**的**平台操作**证书。	EGC, TPEGC

9.7.2.2.2 出口证书定义

9.7.2.2.2.1 出口组证书和撤销列表

ECI出口组证书（EGC）的证书定义应符合第5.2节给出的通用**ECI证书**定义。EGC使用ECI**证书**的标识符字段，具有表9.7.2.2.2.1-1中给出的字段定义。

表9.7.2.2.2.1-1 – ECI出口组ID定义

语法	位数	助记符
ECI_EGC_Id {		
type **/* see** Table 5.3-1*/	4	uimsbf
export_group_id /* see Table 5.3-1 ***/**	20	uimsbf
export_group_version	8	uimsbf
}		

语义：

Type	按照表5.2-1的值。
export_group_id：整数	由管理**出口组**的实体分配给**出口组**的ID。保留值0x00000和0xFFFFF0-0xFFFFF。
export_group_version：整数	带有标识符**export_group_id**的**出口组证书**的版本。

依据第5.3节、尤其是表5.3-1，出于**子证书**的认证目的，EGC应附有一个撤销列表。

9.7.2.2.2.2 第三方出口组证书和撤销列表

ECI第三方**出口组证书**（TPEGC）的**证书**定义应符合第5.2节给出的通用ECI_certificate定义。TPEGC使用**ECI证书**的标识符字段，具有表9.7.2.2.2.2-1中给出的字段定义。

表9.7.2.2.2.2-1 – TPEGC标识符字段定义

语法	位数	助记符
ECI_TPEGC_Id {		
type /* see Table 5.2-1***/**	4	uimsbf uimsbf
tp_export_group_id /* see Table 5.3-1 ***/**	20	uimsbf uimsbf
tp_export_group_version	8	uimsbf uimsbf
}		

语义：

类型	按照表5.3-1的值。
tp_export_group_id：整数	由管理第三方**出口组**的实体分配给第三方**出口组**的ID。保留值0x00000和0xFFFFF0-0xFFFFF。
tp_export_group_version：整数	带有标识符**tp_export_group_id**的第三方**出口组证书**的版本。

表9.7.2.2.2.2-2中定义的TPEGC的扩展字段应包含以下结构，使用表9.7.2.2.2.4-1中的**export_authorization_operator_id**和表9.7.2.2.2.5-1中的**export_authorization_id**的定义。

表9.7.2.2.2.2-2 – TPEGC扩展字段定义

语法	位数	助记符
ECI_TPEGC_Extension {		
export_authorization_operator_id	20	uimsbf
export_authorization_id	20	uimsbf
padding(4)		
Extension_field**extension**		
}		

语义：

export_authorization_operator_id：整数	出口授权**运营商证书**的ECI标识符，用于对此**证书**进行共同认证。
export_authorization_id：整数	共同认证此**证书**的出口授权**证书**的**ECI**标识符（参见第9.7.1.2.2.5条）。
extension: Extension_field	此结构的扩展。

依据第5.3节和表5.3-1，出于**子证书**的认证目的，TPEGC应附有一个撤销列表。

9.7.2.2.2.3 出口授权运营商证书的根撤销列表

出于认证的目的，根据第5.3节和表5.3-1，一个出口认证链必须以根撤销列表开始。

9.7.2.2.2.4 出口授权运营商证书

ECI出口授权**运营商证书**（EAOC）的证书定义应符合第5.2节中给出的通用**ECI证书**定义。EAOC使用**ECI证书**的标识符字段，具有表9.7.2.2.2.4-1中给出的字段定义。

表9.7.2.2.2.4-1 – EAOC标识符字段定义

语法	位数	助记符
ECI_EAOC_Id {		
type **/* see** Table 5.3-1***/**	4	uimsbf
export_authorization_operator_id **/* see** Table 5.3-1 ***/**	20	uimsbf
export_authorization_operator_version	8	uimsbf
}		

语义：

type	按照表5.3-1的值。
export_authorization_operator_id: 整数	分配给出口授权运营商的ID。保留值0x00000和0xFFFFF0-0xFFFFF。
export_authorization_operator_version: 整数	带有标识符**export_authorization_operator_id**的出口授权运营商**证书**的版本。

依据第5.3节和表5.3-1，出于**子证书**的认证目的，EAOC应附有一个撤销列表。

9.7.2.2.2.5 出口授权证书和撤销列表

ECI出口授权**证书**（EAC）的**证书**定义应符合第5.2节中的通用**ECI证书**定义，使用一个特定的非空扩展字段。EAC使用**ECI证书**的标识符字段，具有表9.7.2.2.2.5-1中给出的字段定义。

表9.7.2.2.2.5-1 – EAC扩展字段定义

语法	位数	助记符
ECI_EAC_Id {		
type **/* see** Table 5.3-1***/**	4	uimsbf
export_authorization_id **/* see** Table 5.3-1***/**	20	uimsbf
export_authorization_version	8	uimsbf
}		

语义：

type	按照表5.3-1的值。
export_authorization_id：整数	分配给出口授权**证书**的ID（在其**父**的情形下）。保留值0x00000和0xFFFFF0-0xFFFFF。
export_authorization_version：整数	带有标识符**export_authorization_id**的出口授权**证书**的版本。

EAC的扩展字段应包含将被授权出口的**证书**结构（参见第5.1.3节），不包括**签名**字段，后跟一个扩展字段。

出于**子证书**的认证目的，如果需要对**子**证书进行认证，则根据第5.3节和表5.3-1，EAC应附带一个撤销列表。

9.7.2.2.2.6 出口系统证书

ECI出口系统**证书**（ESC）的**证书**定义应符合第5.2节中定义的通用ECI_certificate定义。**证书**的public_key字段应包含**微处理器**使用的SPK值。ESC使用**ECI证书**的标识符字段，其字段定义如表9.7.2.2.2.6-1所示。

表9.7.2.2.2.6-1 – ESC扩展字段定义

语法	位数	助记符
ECI_ESC_Id {		
type /* **see** Table 5.3-1/	4	uimsbf
export_system_id /* **see** Table 5.3-1 */	20	uimsbf
export_system_version	8	uimsbf
}		

语义：

Type	按照表5.3-1的值。
export_system_id：整数	分配给出口系统**证书**的ID（在其**父**的情形下）。保留值0x00000和0xFFFFF0-0xFFFFF。
export_system_version：整数	带有标识符**export_system_id**的出口系统**证书**的版本。

9.7.2.2.3 出口证书链的验证

带有经预先验证链和补充出口授权链的出口**ECI客户端**应创建请求的**入口/出口连接**。出口**ECI客户端**和入口**ECI微服务器**，**负责**其部分链，在发出与/或尝试获取更新链的情况下，应为**用户**提供信息。**ECI客户端**应提供这些链，以供**AS系统**处理，以便创建所需的**出口/入口连接**。在**AS系统**在任何链或补充出口授权中发现验证错误的情况下，则**ECI客户端**将无法建立所需的连接。

出口授权证书用于共同认证出口**证书**。共同认证的处理规则是：

1) 出口授权**证书**以及要共同认证的**证书**具有有效的签名（由其各自的**父**定义）并且不被撤销。
2) 除了签名外，**证书**中所有要共同认证的数据都将与出口授权**证书**相应扩展字段中的数据进行比较。在未匹配的情况下，共同认证不成功。

为了建立**出口/连接**，CPS应遵循以下处理规则：

1) 应适用第5.4.2节中列出的、**证书链**的所有CPS处理规则。
2) CPS应根据表5.2-2验证**父证书**的**子**类型是否合适。
3) 出口**ECI客户端**的**出口链**的**父**应是客户的**ECI** POC。**出口组**伴随的撤销列表应适用于验证**子出口组证书**。**出口组**的POC撤销列表的版本号应大于客户端的minClientVersion（参见[ITU-T J.1014]）。
4) CPS应接受最大2个级别的EGC，用于出口**ECI客户端**，即第二级EGC的**子**应是TPEGC或ESC。

5) CPS应确保任何TPGC随附一个EAC，它经过一个从根到EAOC再到EAC的链的共同认证（具有伴随的撤销列表）。出口授权运营商证书的根撤销列表的版本应用于确定“系统完整性验证”的最大撤销列表版本号。

6) CPS应确保从TPEGC下降的任何EGC、ESC和TPEGC经一个EAC共同认证，它是验证该**证书**之**父**的EAC的**子**。

出口**ECI客户端**和**微DRM服务器**应在其链上提供足够的预处理，并提供最新的可用版本，以便避免CPS中的撤销。

9.7.2.2.4 用于出口证书的传输协议

9.7.2.2.4.1 概述

出口**ECI客户端**和**微服务器**可以定义用于传输证书数据的自身格式。**ECI**定义用于承载这些数据的标准化文件格式。通过用于广播媒质的**ECI**轮播访问API，**ECI客户端**可访问这些标准化文件。对于客户端的在线提供，**ECI**为此目的定义了标准化的万维网API调用。

9.7.2.2.4.2 出口树文件格式

对于**出口组**树，文件格式在表9.7.2.2.4.2-1中定义。

表9.7.2.2.4.2-1 – ECI出口树文件定义

语法	位数	助记符
ECI_Export_Tree_File {	24	
magic = 'EET'		
image_header_version	8	uimsbf
if (image_header_version == 0x01) {		
ECI_Operator_Id**operator_id**	32	uimsbf
ECI_Platform_Operation_Id**platform_operation_id**	32	uimsbf
ECI_RL_Tree**export_group_tree**		
Extension_Field**extensions**		
}		
}		

语义：

magic:字节[3]	幻数用于验证以下数据的格式。它具有字符“EET”的三个8位ASCII表示的值。**ECI客户端**应检查该字段的值，以验证**ECI**文件是否具有用于其他数据完整性的预期格式。
image_header_version:字节	图像标题的格式版本。值0x01是当前定义的版本；所有其他值都保留。**ECI客户端**应忽略任何未识别版本号的图像。
operator_id: ECI_Operator_Id	文件中所含出口树的**ECI客户端**的**运营商**ID。operator_version字段对应于export_group_tree的根。
Platform-operation_id: ECI_Platform_Operation_Id	文件中所含出口树的**ECI客户端**的**平台操作**ID。
export_group_tree: ECI_RL_Tree	ECI_RL_Tree结构，从**出口组**的**出口组**撤销列表开始。对于不需要补充**撤销列表**的证书，该结构应包含一个空的**撤销列表**，带一个无需与**证书**相匹配的签名。
extensions: Extension_field	由运营商定义的其他数据。

9.7.2.2.4.3 入口链文件格式

对**微服务器**的**入口链**，文件格式在表9.7.2.2.4.3-1中定义。

表9.7.2.2.4.3-1 – ECI入口链文件定义

语法	位数	助记符
ECI_Import_Chain_File {	24	
magic = 'EIC'		
image_header_version	8	uimsbf
if (image_header_version == 0x01) {		
ECI_Operator_Id**operator_id**	32	uimsbf
ECI_Platform_Operation_Id**platform_operation_id**	32	uimsbf
nr_chains	16	uimsbf
padding(4)		
for (i=0; i<nr_chains; i++){		
ECI_Operator_Id**eaoc_id**	32	uimsbf
ECI_Platform_Operation_Id **eac_id**	32	uimsbf
ECI_Certificate_Chain **import_chain**		
}		
Extension_Field**extensions**		
}		
}		

语义：

magic:字节[3]	幻数用于验证以下数据的格式。它具有字符“EIC”的三个8位ASCII表示的值。**ECI客户端**应检查该字段的值，以验证**ECI**文件是否具有用于其他数据完整性的预期格式。
image_header_version:字节	图像标题的格式版本。值0x01是当前定义的版本；所有其他值都予保留。**ECI客户端**应忽略任何版本号未被认可的图像。
operator_id: ECI_Operator_Id	此**入口链**所针对之**微服务器**的**运营商**的ID。
platform_operation_id: ECI_Platform_Operation_Id	此**入口链**所针对之**微服务器**的**平台操作**的ID。
nr_chains:整数	文件中**入口链**的数量。
eaoc_id: ECI_Operator_Id	**入口链**的授权**运营商**的ID。
eac_id: ECI_Platform_Id	共同授权**入口链平台操作**的EAC的ID。
import_chain: ECI_Certificate_Chain	从**入口平台操作证书**到标识**微客户端**之ESG的**ECI证书链**。该链可包含多个TPEGC。每个有效的**入口链**都应分开表示：即，如果链1由两个第三方子链组成，且第二个子链也可单独作为**入口链**使用，则应分开表示。对于不需要补充**撤销列表**的**证书**，该结构应包含一个空的**撤销列表**，带一个无需与**证书**相匹配的签名。
extensions: Extension_field	由运营商定义的其他数据。

9.7.2.2.4.4 出口授权文件格式

对**微服务器**的**出口链**授权，文件格式在表9.7.2.2.4.4-1中定义。

表9.7.2.2.4.4-1 – ECI出口授权文件定义

语法	位数	助记符
ECI_Export_Authorization_File {	24	
magic = 'EEA'		
image_header_version	8	uimsbf
if (image_header_version == 0x01) {		
ECI_Operator_Id**operator_id**	32	uimsbf
ECI_Platform_Operation_Id**platform_operation_id**	32	uimsbf
nr_chains	16	uimsbf
padding(4)		
for (i=0; i<nr_chains; i++){		
direct_flag	1	uimsbf
padding(4)		
ECI_Operator_Id **o_id**	32	uimsbf
ECI_Platform_Operation_Id**po_id**	32	uimsbf
ECI_Certificate_Chain**chain**		
}		
Extension_Field**extensions**		
}		
}		

语义：

magic:字节[3]	幻数用于验证以下数据的格式。它具有字符“EEA”的三个8位ASCII表示的值。**ECI客户端**应检查该字段的值，以验证**ECI**文件是否具有用于其他数据完整性的预期格式。
image_header_version:字节	图像标题的格式版本。值0x01是当前定义的版本；所有其他值都予保留。**ECI客户端**应忽略任何版本号未被认可的图像。
operator_id: ECI_Operator_Id	此**入口链**所针对之**微服务器**的**运营商**的ID。
Platform_operation_id: ECI_Platform_Operation_Id	此**入口链**所针对之**微服务器**的**平台操作**的ID。
nr_chains:整数	此文件中出口授权链的数量。
direct_flag:位	如果值为0b1，则以下链直接授权一个ESC子链，而**o_id**和**po_id**不相关。 如果值为0b0，则以下链授权TPEGC子链，而**o_id**和**po_id**表示授权**证书**id。
o_id: ECI_Operator_Id	第三方**运营商**的ID，由以下出口认证链进行认证的、一个临时的第三方**出口链**。
po_id: ECI_Platform_Öperation_Id	第三方**平台操作**的ID，由以下出口认证链进行认证的、一个临时的第三方**出口链**。
chain: ECI_Certificate_Chain	从**ECI根证书**到认证第一个TPEGC、ESG之EAC的ECI**证书链**。
extensions: Extension_field	由运营商定义的其他数据。

9.7.2.2.4.5 承载出口证书的广播轮播

运营商可以部署如第7.7.2节中定义的**ECI**定义轮播，以承载其选择支持的**ECI客户端**的出口与/或入口证书。不过，对于任何特定的**ECI客户端**，**ECI主机**只需监控数据轮播的单个位置DSI的更新情况。即，为了使用标准轮播格式来承载出口或入口证书，**运营商**应使用相同的轮播，来为此类**ECI客户端**承载客户端图像、**平台操作**证书和撤销数据等。另请参见第7.7.2.1节。

轮播模块的数据格式应符合表7.7.2.6-1的规定。由descriptorType字段等于0xB0的compatibilityDescriptor指定的模块，将承载具有单个ECI_Export_Tree_File结构的模块，那些descriptorType字段等于0xB1的模块，将承载具有单个ECI_Import_Chain_File结构的模块，而那些descriptorType字段等于0xB2的模块，将承载具有单个ECI_Export_Authentication_File结构的模块。

建议监控轮播中更新情况的**ECI客户端**与**ECI主机**要为其他**ECI客户端**数据执行的相一致，以便实现高效的电源管理。

9.7.2.2.4.6 在线提供出口证书

为了允许**ECI客户端**的标准结构访问运营商在线服务器的出口证书，本建议书保留了以下万维网API URL结构。

关于tail_extension的定义和记法约定，参考第7.7.3节：

```
tail_extension*              ::=
        client_export          |
        client_import          |
        client_exp_auth .
```

记法tail_extension *表示其他扩展名可能在本建议书的未来版本中。

为入口/出口定义了以下万维网API请求：

```
client_export ::= 'client-export/' operator_id '/' platform_operation_id .
```

对于由**operator_id**、**platform_operatio_id**指定的**ECI客户端**，它将返回格式为ECI_Export_Tree_File的出口树文件的最新版本。

```
client_import ::= 'client-import/' operator_id '/' platform_operation_id .
```

这将返回格式为ECI_Import_Chain_File的**入口链**文件的最新版本，用于由**operator_id**、**platform_operation_id**指定的**微服务器**客户端。

```
client_exp_auth ::= 'client-exp_auth/' operator_id '/' platform_operation_id .
```

这将返回格式为ECI_Export_Authentication_File的出口认证文件的最新版本，用于**operator_id**、**platform_operation_id**指定的**微服务器**客户端。

9.7.2.3 出口连接API

9.7.2.3.1 概述

ECI客户端可以向**ECI主机**提供出口信息。这允许**ECI主机**将出口系统与来自**微服务器**的匹配**入口链**进行配对。**ECI主机**（和应用程序）可以从所有可能的选项中定义要创建的实际连接。它可以通过利用目标入口**ECI客户端**的**入口链**向出口**ECI客户端**发送一个连接请求，来尝试连接出口和匹配入口**ECI客户端**。出口**ECI客户端**以及**ECI主机**可请求取消连接或者在更新入口证书的情况下重新初始化连接。表9.7.2.3.1-1中列出了可用的**出口连接**消息。

表9.7.2.3.1-1 – 出口连接API消息

消息	类型	方向	标签	描述
reqExpConnNodes	A	H→C	0x0	**ECI主机**从**ECI客户端**请求出口选项节点。
reqExpConnSetup	A	H→C	0x1	**ECI主机**请求**ECI客户端**根据**入口链**将一个**出口连接**初始化为一个入口**ECI客户端**。
reqExpConnDrop	A	H→C	0x2	**ECI主机**取消任何先前初始化的、出口**ECI客户端**到入口**ECI客户端**的连接。
reqExpConnCancel	A	C→H	0x3	**ECI客户端**利用一个入口**ECI客户端**来终止一个初始化的**出口连接**。
reqExpMhOpen	A	H→C	0x4	**ECI主机**请求**ECI客户端**根据先前初始化的**出口连接**来创建一个出口会话。
reqExpMhClose	A	H→C	0x5	**ECI主机**关闭一个出口会话。
reqExpMhCancel	A	C→H	0x6	**ECI客户端**取消一个出口会话。

9.7.2.3.2 reqExpConnNodes消息

H→C reqExpConnNodes() →
C→H resExpConnNodes(ExpConnOption **connNodes [])**

- 该消息请求**ECI客户端**返回其可能带有**出口连接**的列表；**响应**消息返回列表。表9.7.2.3.2-2中列出了相关的错误代码。

响应参数定义：

connNodes: ExpConnOption[]	该列表提供了第三方或**ECI客户端**的**ECI**身份，为出口，**ECI客户端**可连接之。每个选项都有一个优先级：优先级越高，出口无法成功完成的机会就越小。ExpConnNode在表9.7.2.3.2-1中定义。

表9.7.2.3.2-1 – ExpConnNode类型定义

```
typedef  struct  ExpConnNode  {
   uint   targetType;
   uint   operatorId;
   uint   targetId;
   uint   targetPriority;
} ExpConnNode;
```

字段定义：

targetType:uint	目标类型：值等于1的是EAC（第三方），值等于2的是POC（直接出口）。其他值未定义。
operatorId:uint	表示目标出口**运营商**20位的**ECI**证书ID：EAC目标的export_authorization_operator_id，以及POC目标的operator_id。
targetId: uint	表示目标出口20位的**ECI**证书ID，它是EAC目标的export_authorization_id，以及POC目标的platform_operation_id。
targetPriority: uint	选择特定出口的优先级是两部分之和： • 值为1 024的倍数，表示要连接到某个特定**微服务器**的出口的特定（商业）优先级。 • 介于0与1 023之间的值，表示一个分数减去1 024分之一的预期内容使用情况，它可使用该出口**微DRM系统**来导出。 **ECI主机**应使用此信息来自动选择最合适的**微DRM系统**（假设最高优先级系统满足微DRM应用程序的应用要求）与/或在手工选择情况下将上述内容作为一个优先项提供给**用户**。

表9.7.2.3.2-2 – reqExpNodeInfo错误代码

名称	描述
ErrExpConnNwAccess	参见表9.7.2.3.9-1。
ErrExpConnAuthProblem	
ErrExpUninitState	

9.7.2.3.3 reqExpConnSetup消息

H→C reqExpConnSetup (CertChainSerial **Import,**CertChainSerial**Auth[],**ushort**connId)** →

C→H resExpConnSetup ()

- 该消息请求ECI客户端使用入口链入口、出口认证链Auth和ECI客户端链目标，利用带clientId的ECI客户端，来初始化（或重新初始化）出口连接connId。

请求参数定义：

Import: CertChainSerial	**入口链**（从出口TPEGC到ESC）。
Auth: CertChainSerial[]	从根到EAC的出口认证链，用于认证单个第三方子链中的第一个TPEGC。**Auth**中各链的顺序为从出口连接TPEGC到入口POC。
connId:ushort	**出口连接**的ID，由**ECI主机**分配。

CertChainSerial类型和数组类型定义

CertChainSerial是表5.4.1-1中定义的ECI_Certificate_Chain的网络次序表示（大的字节存储次序），填充至32位的倍数。

CertChainSerial[]由以下（准C）数据结构定义：

```
typedef                struct                CertChainSerial                {
   uint   numberElements;     /* the number of elements in the chain array*/
   uint   elementIndex[];     /* the index of the start of each element in
                                    chainElements data container */
   uint   chainElements[];    /*    data    container    with    numberElements
                                    SertChainSerial  representations  of  the
                                     successive chains in the array. */
  } CertChainSerial;
```

elementIndex和chainElements应表示为certChainSerialArray数据结构中的内联数据数组。

详细的语义：

- **ECI主机**可代表现有连接发出reqExpConnSetup请求，以通知出口**ECI客户端**（潜在的）入口**ECI客户端**的新入口证书。除非可立即中断当前连接，否则建议出口**ECI客户端**推迟更新与入口**ECI客户端**的连接，直到此时没有进行任何活动的会话。

表9.7.2.3.3-1列出了相关的错误代码。

表9.7.2.3.3-1 – reqExpConnSetup错误代码

名称	描述
ErrExpConnNwAccess	参见表9.7.2.3.9-1。
ErrExpConnAuthProblem	
ErrExpUninitState	
ErrExpInvalidChain	

9.7.2.3.4 reqExpConnDrop消息

H→C reqExpConnDrop(ushort**connId)** →
C→H resExpConnDrop()

- 该消息请求**ECI客户端**删除由**connId**标识的与客户端的**出口连接**。

请求参数定义：

connId： ushort	**出口连接**的ID。

先决条件请求：

1) **出口连接**（由**connId**标识）先前已建立。

后置条件响应：

2) **出口连接**（如果存在的话）已关闭。

表9.7.2.3.4-1列出了相关的错误代码。

表9.7.2.3.4-1 – reqExpConnDrop错误代码

名称	描述
ErrExpConnNone	参见表9.7.2.3.9-1。

9.7.2.3.5 reqExpConnCancel消息

C→H reqExpConnCancel(ushort**connId)** →
H→C resExpConnCancel()

- 该消息通知ECI主机，由connId标识的出口连接已被ECI客户端终止。

请求参数定义：

connId:ushort	分配给连接的ID。

先决条件请求：

1) 由**connId**标识的**出口连接**先前已建立。

9.7.2.3.6 reqExpMhOpen消息

H→C reqExpMhOpen(ushort**mhExp,** ushort**mhDcr,** ushort**connId)** →
C→H resExpMhOpen(ushort**mhExp)**

- 该消息请求ECI客户端创建一个经出口连接connId由媒质句柄mh标识的出口会话。

请求参数定义：

mhExp:ushort	**媒质句柄**由**ECI主机**分配给**出口连接**。
mhDcr: ushort	要出口的解密会话的**媒质句柄**。
connId: ushort	分配给**出口连接**的ID。

响应参数定义：

mhExp:ushort	**媒质句柄**由**ECI主机**分配给**出口连接**

先决条件请求：

1) 出口连接connId先前已建立。
2) 解密会话mhDcr先前已建立。

后置条件请求：

3) **出口连接**已建立或出现一个错误。

详细的语义：

- 出口**ECI客户端**可在现有会话中暂停和恢复出口，例如，基于**出口组**包含的连接。

表9.7.2.3.6-1列出了相关的错误代码。

表9.7.2.3.6-1 – reqExpMhOpen错误代码

名称	描述
ErrExpConnNone	参见表9.7.2.3.9-1。
ErrExpDcrMhNone	

9.7.2.3.7 reqExpMhClose消息

H→C reqExpMhClose(ushort**mhExp) →**
C→H resExpMhClose(ushort**mhExp)**

- 该消息请求ECI客户端关闭经出口连接connId由媒质句柄mh标识的出口会话。

请求参数定义：

mhExp:ushort	**媒质句柄**由**ECI主机**分配给**出口连接**。

响应参数定义：

mhExp:ushort	**媒质句柄**由**ECI主机**分配给**出口连接**。

先决条件请求：

1) 出口会话**mhExp**先前已建立，但尚未终止。

后置条件请求：

2) 出口会话**mhExp**已停止。

表9.7.2.3.7-1列出了相关的错误代码。

表9.7.2.3.7-1 – reqExpMhClose错误代码

名称	描述
ErrExpMhNone	参见表9.7.2.3.9-1。

9.7.2.3.8 reqExpMhCancel消息

C→H reqExpMhCancel(ushort**mhExp) →**
H→C resExpMhCancel(ushort**mhExp)**

- 该消息通知ECI主机，ECI客户端已停止出口会话mhExp。

请求参数定义：

mhExp:ushort	**媒质句柄**由**ECI主机**分配给**出口连接**。

响应参数定义：

mhExp:ushort	**媒质句柄**由**ECI主机**分配给**出口连接**。

先决条件请求：

1) 出口会话**mhExp**先前已建立。

2) **ECI客户端**终止会话。

9.7.2.3.9 出口连接API的错误代码

API特定错误的值可由该API的**响应**消息返回，列于表9.7.2.3.9-1中。

表9.7.2.3.9-1 – TS媒质的媒质会话API的错误代码

名称	值	描述
ErrExpConnNwAccess	-256	访问提供所需信息的网络是不可能的，或者意外地变慢且无法完成。
ErrErrConnAuthProblem	-257	检测到在已提供的数据中存在内部不一致问题，阻止要完成的请求。
ErrEcxpConnUninitState	-258	**ECI客户端**首先需要提供与/或其他执行功能，以便能够对该请求做出响应。
ErrExpConnInvalidChain	-259	提供给**ECI客户端**的一个链，被认定为无效与/或无法使用认证链来对其进行认证。
ErrExpConnNone	-260	连接不存在。
ErrExpMhNone	-261	**ECI客户端**不支持由**媒质句柄**指示的出口会话。
ErrExpDcrMhNone	-262	**ECI客户端**不支持由**媒质句柄**指示的解密会话。

9.7.2.4 入口连接API

9.7.2.4.1 概述

ECI客户端可以将其**入口链**提供给**ECI主机**。这允许**ECI主机**将入口**ECI客户端**连接到来自**微服务器**的匹配出口选项。**ECI主机**和应用程序可以从可用的连接选项中选择建立要创建的连接。**ECI主机**可以开始在出口与入口**ECI客户端**之间建立连接，方法是首先请求入口客户端允许将其连接到出口**ECI客户端**。入口客户端可以基于以下来拒绝这种连接，如其运营商的商业考虑。在建立连接的情况下，入口**ECI客户端**以及**ECI主机**可以请求取消连接或在更新的入口证书的情况下重新初始化连接。

输入链由其第一个节点来标识，即TPEGC的EAOC和EAC的**ECI** id。这在下面的表9.7.2.4.1-1中被称为*入口节点*。

表9.7.2.4.1-1 – 入口连接API消息

消息	类型	方向	标签	描述
reqImpConnNodes	A	H→C	0x0	**ECI主机**请求入口**ECI客户端**提供其入口节点。
reqImpConnChain	A	H→C	0x1	**ECI主机**请求入口**ECI客户端**为特定入口节点提供输入链。
reqImpConnChainRenew	A	C→H	0x2	**ECI客户端**请求**ECI主机**使用更新的**入口链**重新初始化连接。
reqImpConnSetup	A	H→C	0x3	**ECI主机**请求入口**ECI客户端**通过一个入口节点来初始化与特定出口**ECI客户端**的**入口连接**。
reqImpConnDrop	A	H→C	0x4	**ECI主机**删除与指定出口**ECI客户端**的**入口连接**。
reqImpConnCancel	A	C→H	0x5	**ECI客户端**终止与指定出口**ECI客户端**的**入口连接**。

9.7.2.4.2 reqImpConnNodes消息

H→C reqImpConnNodes () →
C→H resImpConnNodes(ImpConnNode **nodes[])**

- 该消息使**ECI主机**能够请求入口**ECI客户端**提供其入口节点。

响应参数定义：

nodes[]**:** ImpConnNode	入口节点数组和第三方中介数量。ImpConnNodes的结构在表9.7.2.4.2-1中定义。

表9.7.2.4.2-1 – ImpConnOption类型定义

```
typedef str uctImpConnNode {
   uint  targetType;
   uint  operatorId;
   uint  targetId;
   uint  intermediaries
} ImpConnNode;
```

字段定义：

targetType:uint	目标类型：1为EAC（第三方），2为POC（直接出口）。其他值未定义。
operatorId:uint	表示**目标**入口**运营商**的20位**ECI**证书ID：EAC目标的export_authorization_operator_id，或POC目标的operator_id。
targetId: uint	表示**目标**入口的20位**ECI**证书ID：EAC目标的export_authorization_id，或POC目标的platform_operation_id。
intermediaries: uint	表示从输入节点到入口**ECI客户端**POC的中间第三方的数量。**ECI主机**应为出口选项在可选项中选择最短的**入口链**，对出口**ECI客户端**，它们有相同的targetPriority。

表9.7.2.4.2-2列出了相关的错误代码。

表9.7.2.4.2-2：reqExpConnInfo错误代码

名称	描述
ErrImpConnNwAccess	参见表9.7.2.4.7-1。
ErrImpConnAuthProblem	
ErrImpUninitState	

9.7.2.4.3 reqImpConnChain和reqImpConnChainRenew消息

H→C reqImpConnChain(ImpConnNode**node)** →
C→H resImpConnChain(CertChainSerial **Import,**CertChainSerial **Auth[])**

- 该消息使**ECI主机**能够请求入口**ECI客户端**为特定的入口节点提供输入链。

C→H reqImpConnChainRenew(CertChainSerial **Import,**CertChainSerial**Auth[])** →
H→C resImpConnChainRenew()

- 该消息使**ECI客户端**能够请求**ECI主机**使用更新的**入口链**重新初始化连接。

reqImpConnChain的请求参数：

node: ImpConnNode	入口节点，为此**入口链**应返回给**ECI主机**。

reqImpConnChainRenew的请求参数定义和reqImpConnChain的响应参数定义：

Import: CertChainSerial	**入口链**（从出口TPEGC到ESC）。
Auth: CertChainSerial[]	从根到EAC的出口验证链，用于验证单个第三方子链中的第一个TPEGC。**Auth**中的链依次从出口TPEGC到进口POC。

先决条件reqImpConnChainRenew请求：

1) 先前，使用提供的链中的元素，与**ECI客户端**建立了**入口连接**。

reqImpConnChainRenew的详细语义：

- **ECI主机**应立即将更新的链信息传递给受影响的出口**ECI客户端**。
- 建议运营商在弃用上一个链之前提供更新链，以确保不中断服务提供。

表9.7.2.4.3-1列出了reqImpConnChain相关的错误代码。

表9.7.2.4.3-1 – reqImpConnChain错误代码

名称	描述
ErrImpConnNwAccess	参见表9.7.2.4.7-1。
ErrImpConnAuthProblem	
ErrImpConnUninitState	

表9.7.2.4.3-2列出了reqImpConnChainRenew相关的错误代码。

表9.7.2.4.3-2 – reqImpConnChainRenew错误代码

名称	描述
ErrImpConnNoConn	参见表9.7.2.4.7-1。

9.7.2.4.4 reqImpConnSetup消息

H→C reqImpConnStart (ImpConnNode**node,** ushort**exportClientId,** ushort**connId)** →
C→H resImpConnStart()

- 该消息使**ECI主机**能够请求入口**ECI客户端**，以便通过入口节点与特定的出口**ECI客户端**建立**入口连接**。

请求参数：

node: ImpConnNode	通过其建立连接的入口节点。
exportClientId: ushort	出口**ECI客户端**的**ECI主机**标识。
connId: ushort	**分配给入口连接的ID。**

详细的语义：

- **ECI客户端**可能会根据其运营商的商业考虑拒绝**入口连接**。

表9.7.2.4.4-1列出了相关的错误代码。

表9.7.2.4.4-1 – reqExpConnStart错误代码

名称	描述
ErrImpConnNwAccess	参见表9.7.2.4.7-1。
ErrImpConnAuthProblem	
ErrImpConnUninitState	
ErrImpConnRefuseComm	
ErrImpConnUnknError	

9.7.2.4.5 reqImpConnDrop消息

H→C reqImpConnDrop (ushort**connId)** →
C→H resImpConnDrop()

- 该消息使**ECI主机**能够删除与指定的出口**ECI客户端**的**入口连接**。

请求参数：

connId:ushort	要删除的**入口连接**的**ECI主机**标识。

先决条件请求：

1) **入口连接**（由**connId**标识）先前已初始化。

后置条件响应：

2) **出口连接**（如果存在的话）已关闭。

表9.7.2.4.5-1列出了相关的错误代码。

表9.7.2.4.5-1 – reqExpConnInfo错误代码

名称	描述
ErrImpConnNwAccess	参见表9.7.2.4.7-1。
ErrImpConnAuthProblem	
ErrImpConnUninitState	
ErrImpConnNoConn	

9.7.2.4.6 reqImpConnCancel消息

C→H reqImpConnCancel (ushort**connId)** →
H→C resImpConnCancel()

- 该消息使**ECI客户端**能够终止与指定的出口**ECI客户端**的**入口连接**。

请求参数：

connId:ushort	**入口连接**（由connId标识）先前已被初始化。

先决条件请求：

1) **入口连接**先前已通过**ECI主机**客户端ID **exportClientId**与客户端建立，并已关闭。

9.7.2.4.7 出口连接API的错误代码

这些值指的是API特定的错误，它们可由该API的**响应**消息返回，列于表9.7.2.4.7-1中。

表9.7.2.4.7-1 – TS媒质的媒质会话API的错误代码

名称	值	描述
ErrImpConnNwAccess	-256	提供关于所需信息之信息的网络访问意外变慢。
ErrImpConnAuthProblem	-257	检测到已提供数据中的内部不一致，从而阻止要完成的请求。
ErrImpUninitState	-258	**ECI客户端**首先需要提供与/或其他执行功能，以便能够对该请求做出响应。
ErrImpConnRefuseComm	-259	提供给**ECI客户端**的链被认定为无效与/或无法使用验证链对其进行验证。
ErrImpConnRefuseComm	-260	入口**ECI客户端**拒绝根据商业条件连接到出口**ECI客户端**。
ErrImpConnUnknError	-261	入口**ECI客户端**遇到未知的错误。
ErrExpConnNone	-262	连接不存在。

9.7.2.5 重加密API

9.7.2.5.1 概述

重加密API允许**微服务器**对来自特定于一组客户端中之一的**入口连接**的内容进行重加密，以便随后**微客户端**进行解码。解码可能必须近实时完成（流连接），并且可能不允许在随后的会话上进行重播，或者可替代地，利用解码**微客户端**的相关解密信息存储或时移重解密的内容，稍后由**微客户端**来解码。

发现阶段允许应用程序将**微服务器**与可能的**目标**（**微客户端**或一组**微客户端**）相匹配，并将所需的认证信息从**微客户端**交换到**微服务器**，以允许对**微客户端**进行认证，并允许作为内容可信交换的基础。**ECI主机**可以选择双向通信模式（基于IP或通过**ECI主机**传递的消息），以便支持**微服务器**与**微客户端**之间更加复杂的认证协议。

基于对**目标**和**入口连接**的重加密连接，**ECI主机**可以实例化应用程序所需之模式（重加密、同步和数据格式模式）的**媒质句柄**会话，可通过**微服务器**支持之。

一旦建立了重加密连接，**ECI主机**就可以使用**微服务器**来实例化一个**媒质句柄**会话，并开始从**目标**（**ECI客户端**或**ECI客户端**组）的已建立**入口连接**对内容进行重加密。可以实例化相同内容的多个同时重加密，每个使用它自己的**媒质句柄**会话。**ECI主机**负责确保重加密**媒质句柄**会话的内容来源于**出口连接**上经认证的出口**媒质句柄**。未经授权的错误连接将导致一个出口认证故障。

重加密控制字应用于入口的解密内容，并使用AS系统将新标记（URI等）应用于重加密的内容。

可以有三种主要的*加密模式*：

1) 在线流模式：**微服务器**和**微客户端**同时处于活动状态。它们直接（通过一个IP信道）交换消息，或者通过其**ECI主机**直接交换消息。
2) 离线流模式：**微服务器**将“即时”加密内容，并定期发布**微客户端**解密所需的新数据。结果可能会被延迟（时间偏移模式）或存储。
3) 离线存储模式：**微服务器**加密内容，并在最后生成**微客户端**在开始解码内容时所需的数据。

图9.7.2.5.1-1给出了不同加密模式的示意图。

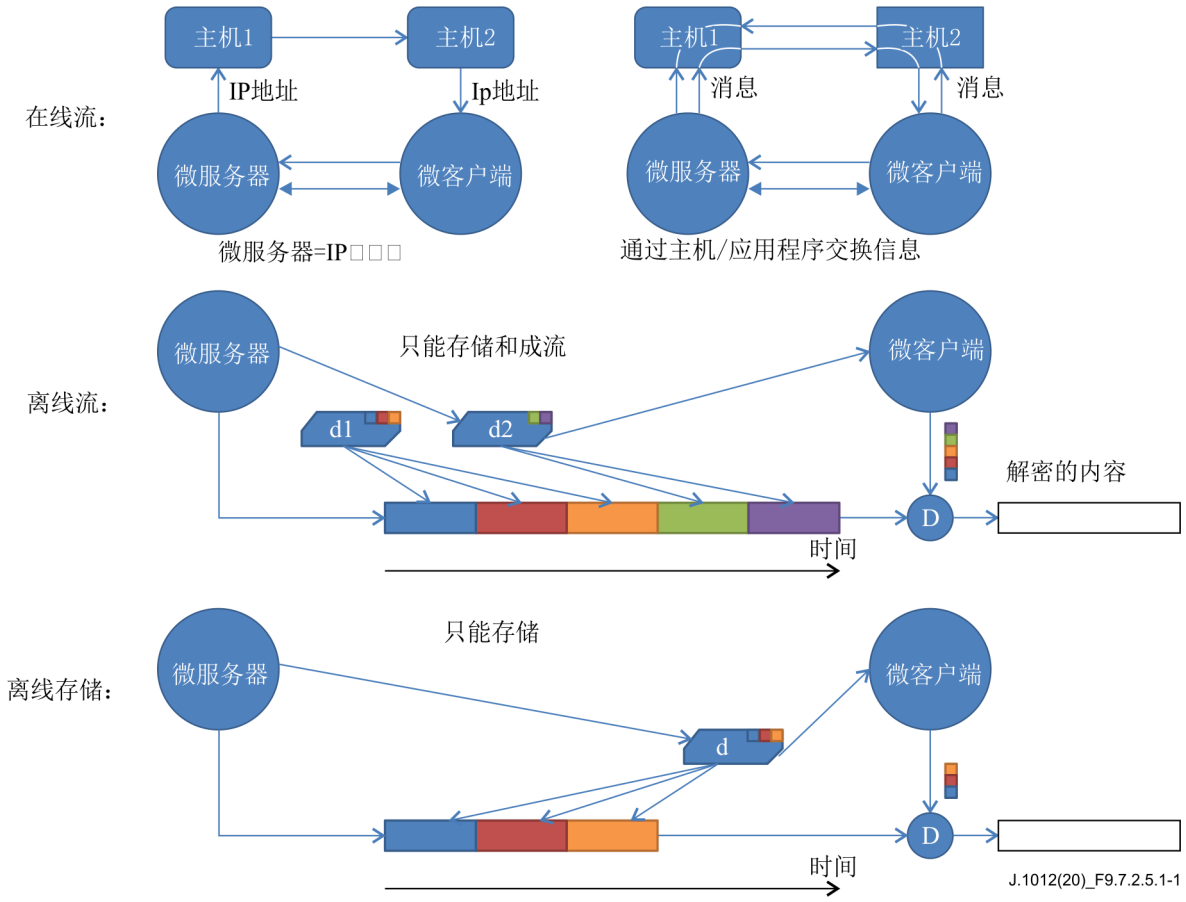


图9.7.2.5.1-1 – 微DRM会话的加密模式

在**微服务器**与**微客户端**之间、在两种离线加密模式下、对要交换的内容进行解密所需的数据可通过以下数据格式模式来传递：

4) 通用模式：**微服务器**生成不透明的数据容器，其中包含**微客户端**解密内容所需的信息。

5) ISOBMFF模式（仅适用于同步模式等同文件模式）：**微服务器**生成包含在ISOBMFF文件[ISO/IEC 14496-12]中的PSSH框。**ECI主机**可以通过在ISOBMFF MOOV或MOOF框中适当包含PSSH框，使用它们来创建ISOBMFF文件。

支持同步模式的两种机制允许将正确的控制字与某个内容区段相关联，适用于所有上述重加密模式：

6) 在传输流（交替位）模式下，**微服务器**生成ECM区段，它可以通过**ECI主机**打包并插入传输流中。ECM插入在加密周期之前，它提供允许计算控制字的信息。

7) 在文件模式下，**微服务器**产生加密的控制字，在附加的解密信息中，它们通过明确的KeyID标识符来引用。**ECI主机**应保留用特定控制字加密的、内容区段的KeyID关联，以便**微客户端**能够为解扰产生正确的控制字。

在离线模式下，解密或计算KeyId或ECM所需的同步附加数据明确引用数据相对于KeyId或ECM编号的时间依赖性关系。

并非所有的**微服务器**都必须支持所有的操作模式。初始化时，在使用发现API后，**微服务器**立即发令告知它可以支持的模式（加密模式、数据格式模式和同步模式的组合）。

一旦**媒质句柄**会话得到实例化，**ECI主机**就可以启动和停止之，并可通过**ECI客户端**来取消之。

表9.7.2.5.1-1列出了重加密API的消息。

表9.7.2.5.1-1 – 重加密API消息

消息	类型	方向	标签	描述
setEncrModes	set	C→H	0x0	**微服务器**通知**ECI主机**它支持的模式（加密模式、数据格式模式和同步模式）。
reqEncrTargets	A	H→C	0x1	**ECI主机**请求**微服务器**提供它可以进行认证以解密的**目标**节点。
reqEncrConnSetup	A	H→C	0x2	**ECI主机**请求**ECI客户端**创建一个重加密**目标**连接，并对重加密**目标**进行预认证，以便随后在设置**媒质句柄**会话中参考。
reqEncrConnDrop	A	H→C	0x3	**ECI主机**请求**ECI客户端**删除先前经预认证的重加密连接上的任何信息。
reqEncrConnCancel	A	C→H	0x4	**ECI客户端**取消先前建立的加密**目标**连接。
reqEncrMhOpen	A	H→C	0x5	**ECI主机**请求**ECI客户端**打开**媒质句柄**会话，以重加密来自输入**入口连接**的内容，以建立重加密连接。
reqEncrMhClose	A	H→C	0x6	**ECI主机**利用**ECI客户端**关闭**重加密会话**。
reqEncrMhCancel	A	C→H	0x7	**ECI客户端**终止与指定的出口**ECI客户端**的**入口连接**。
reqEncrMhStart	A	H→C	0x8	**ECI主机**请求**ECI客户端**启动**媒质句柄**会话的重加密操作。
reqEncrMhStop	A	H→C	0x9	**ECI主机**请求**ECI客户端**停止**媒质句柄**会话的重加密操作。
reqEncrMhQuit	A	C→H	0xA	**ECI客户端**通知**ECI主机**已终止**媒质句柄**重加密操作。
reqEncrIpServer	A	H→C	0xB	**ECI主机**请求**微服务器**的IP服务器地址，以便允许**微客户端**创建IP连接。
reqEncrMsgSend	A	C→H	0xC	**微服务器**请求**ECI主机**将消息转发给**媒质句柄**会话的目标。
reqEncrMsgRecv	A	H→C	0xC	**ECI主机**为**微服务器**提供一条来自**媒质句柄**会话**目标**的消息。
reqEncrTsData	A	C→H	0xE	**微服务器**向**ECI主机**提供数据，以便将之转发给**媒质句柄**的**目标微客户端**以进行解密，包括与ECM相关的同步信息。
reqEncrTsEcm	A	C→H	0xF	**微服务器**发出**微客户端**所需的ECM区段，以便在下一个加密周期进行解密。
reqEncrFileData	A	C→H	0x10	**微服务器**为**ECI主机**提供一条消息，以便将之转发给**媒质句柄**的**目标微客户端**以进行解密，包括KeyID相关的同步信息。

9.7.2.5.2 setEncrModes消息

C→H setEncrModes(EciEncrModes **modes)**

- 该消息允许**微服务器**通知**ECI主机**它支持的模式（加密模式、数据格式模式和同步模式）。

请求参数定义：

模式：EciEncrModes	**微服务器**支持的加密模式。表9.7.2.5.2-1中指定了EciEncrModes类型。

表9.7.2.5.2-1 – EciEncrModes类型定义

```
Typedef uintEciEncrModes;
```

位定义：

名称	位	微服务器模式支持值等于0b1
OnlineIpMode	0	支持在线IP模式。
OnlineMsgMode	1	支持在线消息模式。
OfflineStreamMode	2	支持离线流模式。
OfflineStorageMode	3	支持离线存储模式。
OfflineDataMode	4	在离线模式下支持解密数据的默认数据格式容器。如果未选择任何离线模式，则不相关。
OfflineIsobmffMode	5	在离线模式下支持ISOBMFF格式PSSH框来解密数据。如果未选择任何离线模式，则不相关。
SyncTs	6	将控制字同步到传输流格式，交替使用位分隔的内容密码周期。
SyncFile	7	使用KeyID标识同步到文件类型格式，以将内容区段与其控制字相关联。
其他	RFU	保留以供未来使用。

9.7.2.5.3 reqEncrTargets消息

H→C reqEncrTargets()→
C→H resEncrTargets(EncrTarget**target[])**

- 该消息允许**ECI主机**请求**微服务器**提供它可以进行验证的加密目标。

响应参数定义：

目标：EncrTarget []	**微服务器**可以进行验证的加密目标列表。TargetClient的类型定义在表9.7.2.5.3-1中指定。

表9.7.2.5.3-1 – EncrTarget类型定义

```
typedef struct EncrTarget {
   uint   targetType;
   byte   target[8];
} EncrTarget;
```

字段定义：

targetType：uint	加密目标的类型：值等于1为单个客户端，值等于2意味着客户端组，保留其他值以供未来使用。
target：字节[8]	代表目标的ID。该值在**微DRM系统**的范围内定义。**ECI主机**的匹配情况根据**目标类型**和**目标**字段的等同性来确定。

详细的语义：

- **ECI主机**可以根据**目标**来匹配潜在的**目标微客户端**。由应用程序与/或**ECI主机**来定位潜在的候选**微客户端**。
- 希望执行本地PVR和时移功能的**ECI主机**（使用可存储加密之内容和相关数据的、集成的或连接的/联网的存储介质）可以尝试将能够工作于**OfflineStreamMode**下的**微服务器**与安装于相同**ECI主机**上的**微客户端**进行匹配。

9.7.2.5.4 reqEncrConnSetup消息

H→C reqEncrConnSetup(ushort**targetConnId**, EciEncrTarget**target**, ushort**credLen**, byte **cred[])**
C→H resEncrConnSetup(ushort**targetConnId)**

- 该消息允许**ECI主机**请求**微服务器**创建到**目标**的重加密连接，并对**目标**进行（预）认证。错误代码在表9.7.2.5.19-1中定义。

请求参数定义：

targetConnId:ushort	用于进一步参考**ECI主机**与**微服务器**之间目标的ID。
target: EciEncrTarget	代表认证**目标**的ID。该值在**微DRM系统**的范围内进行定义。由**ECI主机**进行的匹配根据**目标类型**和**目标**字段的等同性来定义。
credLen: ushort	cred参数的长度，以字节为单位。
cred:字节[]	来自将要由**微服务器**进行认证的**目标**的证书信息。

响应参数定义：

targetConnId：ushort	用于进一步参考**ECI主机**与**微服务器**之间**目标**的ID。

详细的语义：

- 在targetConnId等于先前由ECI主机使用但之后未丢弃的targetConnId的情况下，隐含的是与targetConnId关联的前一个目标已被替换或更新。

先决条件请求：

1) **目标**应该等于先前由**微服务器**在**resEncrTargets**消息中提供给**ECI主机**的**目标**。否则，为该参数返回一个错误。
2) **目标**应该匹配于**微客户端**提供的**目标**，并允许使用**cred**进行认证。

后置条件响应：

3) 返回认证状态。请注意，结果并不一定是结论性的，并且例如可能提供导致无法解码之加密内容的错误证书。
4) ECI主机可以通过targetConnId引用（预）验证的目标。

表9.7.2.5.4-1 – reqEncrConnSetup错误代码

名称	描述
ErrEncrAuthFail	参见表9.7.2.5.19-1。
ErrEncrAuthInconclusive	

9.7.2.5.5 reqEncrConnDrop消息

H→C reqEncrConnDrop(ushort**targetConnId) →**
C→H resEncrConnDrop(ushort**targetConnId)**

- 该消息允许**ECI主机**请求**微服务器**放弃先前预验证的重加密连接上的任何信息。

请求参数定义：

targetConnId：ushort	**目标**连接的Id将由**微服务器**删去。

响应参数定义：

targetConnId：ushort	**目标**连接的Id从**微服务器**中删去。

先决条件请求：

1) targetConnId应存在于微服务器中。

先决条件响应：

2) **微服务器**不再将**targetConnId**与预验证的**目标**连接相关联，并已释放与**targetConnId**预验证相关的任何资源。

9.7.2.5.6 reqEncrConnCancel消息

C→H reqEncrConnCancel(ushort**targetConnId) →**
H→C resEncrConnDrop(ushort**targetConnId)**

- 该消息允许**微服务器**在**ECI主机**中通知它已取消先前预验证的重加密连接。

请求参数定义：

targetConnId:ushort	**目标**连接的Id由**微服务器**取消。

响应参数定义：

targetConnId:ushort	**目标**连接的Id由**微服务器**取消。

先决条件请求：

3) targetConnId应存在于微服务器中。

先决条件响应：

4) TargetConnId值已被释放，并可由**ECI主机**重新分配，作为随后的reqEcnrConnSetup消息的一部分。

9.7.2.5.7 reqEncrMhOpen消息

H→C reqEncrMhOpen(ushort**mh,** ushort**impConn,** ushort**targetConnId,** EncrMode **mode) →**
C→H resEncrMhOpen(ushort**mh)**

- 该消息允许**ECI主机**请求**ECI客户端**打开一个**媒质句柄**会话，以便在来自**入口连接**的**微服务器**的控制下重加密内容，从而转发给一个经预认证的目标。错误代码在表9.7.2.5.7-1中进行定义。

请求参数定义：

mh: ushort	要打开的加密会话的**媒质句柄**，由**ECI主机**来分配。
impConn: ushort	入口连接的Id，从中将对内容进行重加密。
targetConnId: ushort	**目标**连接的Id，为之将对内容进行重加密。
mode: EncrMode	单个模式（加密模式、数据格式模式、同步模式）的规范，用于**微服务器**的操作，从**微服务器**模式功能中选择，如**setEncrModes**所示。

响应参数定义：

mh: ushort	用于要打开的加密会话的**媒质句柄**，由**ECI主机**来分配。

先决条件请求：

5) **ECI主机**保留了要创建之会话所需的全部资源。

6) impConn和targetConnId由ECI主机利用微服务器来建立。

先决条件响应：

7) 在成功取得结果的情况下，**微服务器**已保留通常为所请求会话的重加密内容所需的所有资源。这应该包括访问解密操作通常所需的任何外部资源（DRM服务器、**智能卡**等）。

注 – 异常需要的资源，或者当需要的被排除时，通常可获得的资源。

8) 在返回**ErrEncrUserDelay**的情况下，**微服务器**等待**用户**输入以打开会话（例如，获得对**智能卡**的访问或获取**用户**认证）。**ECI主机**可以重复发送**reqEncrMhOpen**请求（使用相同的参数），直到返回一个肯定的结果或者返回一个确定的错误，或者可选地可发送一个**reqEncrMhClose**来终止未决的会话。在无法获得所需**用户**输入的情况下，**微服务器**可取消**reqEncrMhCancel**。

表9.7.2.5.7-1 reqEncrMhOpen错误代码

名称	描述
ErrEncrUserMissing	参见表9.7.2.5.19-1。
ErrEncrCardMissing	
EnrEncrServiceMissing	
ErrEncrResourceMissing	
ErrEncrMmiMissing	
ErrEncrClientAuthError	

9.7.2.5.8 reqEncrMhClose消息

H→C reqEncrMhClose(ushort**mh) →**
C→H resEncrMhClose(ushort**mh)**

- 该消息允许ECI主机关闭与微服务器的重加密会话。

请求参数定义：

mh: ushort	将关闭有关加密会话的**媒质句柄**。

响应参数定义：

mh： ushort	将关闭有关加密会话的**媒质句柄**。

先决条件请求：

1) **媒质句柄**会话处于打开状态（否则将发生错误）。

先决条件响应：

2) 释放**微服务器**维护会话所需的资源。

3) 客户端关闭**mh**状态。

9.7.2.5.9 reqEncrMhCancel消息

C→H reqEncrMhCancel(ushort**mh,** uchar **reason) →**
H→C resEncrMhCancel(ushort**mh)**

- 该消息允许ECI客户端使用指定的出口ECI客户端（微服务器）来关闭重加密会话。

请求参数定义：

mh: ushort	由**微服务器**取消的、加密会话的**媒质句柄**。
reason: uchar	取消解密会话的原因。这些值在表9.7.2.5.9-1中定义。

表9.7.2.5.9-1 – reqEncrMhCancel原因值

名称	值	描述
EncrMhUndefined	0x00	**微服务器**中出现一个未定义的错误，要求它取消会话。
EncrMhCardMissing	0x01	**智能卡**需要重加密，但无法成功（重新）连接，并协助在合理的时间内重加密内容。
EncrMhServiceMissing	0x02	支持**微服务器**提供维护解密会话所需之加密服务的服务（在**CPE**外部）在合理的时间内不可用。
EncrMhResourceMissing	0x03	提供重加密服务所需的资源（在**CPE**内部）在合理的时间内对**微服务器**不可用（不包括**DcrMhMmiMissing**）。
EncrMhMmiMissing	0x04	**微服务器**未成功获得在合理的时间内维护**重加密会话**所需的、用于**用户**交互的MMI会话资源。
RFU	其他	保留以供未来使用。

响应参数定义：

mh: ushort	已取消的加密会话的**介质句柄**。

先决条件请求：

1) **ECI客户端**已发布专门用于会话的任何资源。

后置条件请求：

2) **ECI主机**可以释放与**媒质句柄**相关的任何资源。

后置条件响应：

3) **媒质句柄**会话由**ECI主机**来关闭。

9.7.2.5.10 reqEncrMhStart消息

H→C reqEncrMhStart(ushort**mh) →**
C→H resEncrMhStart(ushort**mh)**

- 该消息允许**ECI主机**请求**微服务器**启动**媒质句柄**会话的重加密操作。

请求参数定义：

mh：ushort	要启动加密会话的**媒质句柄**。

响应参数定义：

mh：ushort	已启动加密会话的**媒质句柄**。

先决条件请求：

1) **媒质句柄**会话处于打开状态（否则将发生错误）。

先决条件响应：

2) **媒质句柄**会话启动（或发生错误）。

详细的语义：

- 内容加密将随着内容由出口**ECI客户端**提供而继续进行。
- 用于认证内容出口之**微服务器**的出口**ECI客户端**的任何URI冲突或失败都将不会生成经加密的内容，**微服务器**的输出控制URI状态设置为OcAnyOther等于0b1，所有其他输出控制位都将设置为 0b0（表示不允许任何输出）。在允许的情况下，**微服务器**将继续尝试对内容进行加密。
- 为此目的，任何初始化消息都可以通过相应的消息提供给**微客户端**。对于重加密模式等于**OfflineStreamMode**的会话，用于解密内容的第一个初始化数据在**resEncrMhStart**消息后不久即生成。
- 在结束加密过程之前发送第二个**reqEncrMhStart**将结束前一个过程并开始下一个过程。

9.7.2.5.11 reqEncrMhStop消息

H→C reqEncrMhStop(ushort**mh) →**
C→H resEncrMhStop(ushort**mh)**

- 该消息允许**ECI主机**请求**微服务器**停止**媒质句柄**会话的重加密操作。

请求参数定义：

mh：ushort	要结束加密会话的**媒质句柄**。

响应参数定义：

mh：ushort	已结束加密会话的**媒质句柄**。

先决条件请求：

3) **媒质句柄**会话处于已启动状态（或者将发生错误）。

先决条件响应：

4) **媒质句柄**会话结束。

后置条件响应：

5) **媒质句柄**会话值可被**ECI主机**重新使用。

详细的语义：

- 在加密模式等于**OfflineStorageMode**的会话上，最终解密数据在**微服务器**发送**resEncrMhStop**之前生成。这也适用于任何最终的解密数据，它们在其他类型的会话的解密中也可能需要。

9.7.2.5.12 reqEncrMhQuit消息

C→H reqEncrMhQuit(ushort**mh,** uchar **reason)** →
C→H resEncrMhQuit(ushort**mh)**

- 该消息允许**微服务器**通知**ECI主机**，**媒质句柄**关联的重加密操作已终止。

请求参数定义：

mh: ushort	加密会话的**媒质句柄**已终止。
reason: uchar	如表9.7.2.5.9-1中给出的原因。

响应参数定义：

mh: ushort	加密会话的**媒质句柄**已终止。

先决条件请求：

1) **媒质句柄**会话处于已启动状态，但现在予以终止。

先决条件响应：

2) **ECI主机**知晓会话加密的未启动状态。

详细的语义：

- 在出现准永久性错误的情况下，**微服务器**也可以取消**媒质句柄**会话本身。
- 在**微服务器**在终止**重加密会话**之前可以生成有效解密数据的情况下，在加密模式等于**OfflineStorageMode**的会话上，最终的解密数据在**微服务器**发送**resEncrMhQuit**之前生成。这也适用于在其他类型的会话中解密可能需要的任何最终解密数据。

9.7.2.5.13 reqEncrIpServer消息

H→C reqEncrIpServer(ushort**mh)** →
C→H resEncrIpServer(ushort**mh.** Addrinfo**addr)**

- 该消息允许**ECI主机**请求**微服务器**为来自**微客户端**的输入IP连接提供**目标**IP地址。

请求参数定义：

mh：ushort	加密会话的**媒质句柄**，对之，需要一个输入消息或连接的IP地址。

响应参数定义：

mh：ushort	加密会话的**媒质句柄**，对之，需要一个输入消息或连接的IP地址。
addr：Addrinfo	**微客户端**输入消息或连接的IP协议/地址/端口。

先决条件请求：

1) 媒质句柄会话以OnlineIpMode模式来打开。

先决条件响应：

2) **ECI主机**知晓会话加密的未启动状态。

详细的语义：

- **微客户端**与**微服务器**之间的IP交换特定于**微DRM系统**。这包括协议选择以及用于终止内容流会话上的连接或交换的任何约定。
- 该消息可在重加密过程尚未启动的**媒质句柄**会话上发出。

表9.7.2.5.13-1 – reqEncrIpServer错误代码

名称	描述
ErrEncrIpNone	参见表9.7.2.5.19-1。

9.7.2.5.14 reqEncrMsgSend消息

C→H reqEncrMsgSend(ushort**mh,** uint **length,** byte**msg[]) →**
C→H resEncrMsgSend(ushort**mh)**

- 该消息允许**微服务器**请求**ECI主机**将消息转发给与**媒质句柄**关联的**目标微客户端**或**微客户端**（在组目标的情况下）。

请求参数定义：

mh: ushort	加密会话的**媒质句柄**，对之必须将消息转发给**目标微客户端**。
length: uint	**msg**字段的长度，以字节为单位。
msg[]: 字节	要转发给**微客户端**的消息。

响应参数定义：

mh: ushort	加密会话的**媒质句柄**。

先决条件请求：

1) 媒质句柄会话以OnlineMsgMode模式打开。

先决条件响应：

2) 该消息已被转发给微客户端；ECI主机已准备好接受新的reqEncrMsgSend。

详细的语义：

- **ECI主机**应能够一次处理和转发至少一条消息到**微客户端**。消息应按顺序发送。**ECI主机**没有义务为多个同时未完成的**reqEncrMsgSend**请求提供任何特定的缓冲。一个安全的**微服务器**实施方案应该使用**resEncrMsgSend**作为一次控制流握手。
- **ECI主机**转发机制应具有足够的可靠性，以使常规应用程序不会失败（消息丢失或者每10000个失序1个）。建议应用程序（当中，加密内容的基本访问信息可能会永久丢失，或者在此期间高价值查看可能会受损）采取额外的应用程序级别的预防措施。

9.7.2.5.15 reqEncrMsgRecv消息

H→C reqEncrMsgRecv(ushort**mh,** uint **length,** byte**msg[]) →**
C→H resEncrMsgRecv(ushort**mh)**

- 该消息允许**ECI主机**向**微服务器**提供一条来自目标**微客户端**的消息。

请求参数定义：

mh: ushort	加密会话的**媒质句柄**，对之**微服务器**从目标**微客户端**中获取消息。
length: uint	**msg**字段的长度，以字节为单位。
msg: 字节[]	要由**微服务器**接收的消息。

响应参数定义：

mh: ushort	需要输入消息或连接之IP地址的加密会话的**媒质句柄**。

先决条件请求：

1) 媒质句柄会话以OnlineMsgMode模式打开。

先决条件响应：

2) 该消息已由**微服务器**进行处理，并已准备好接受新的**reqEncrMsgRecv**。

详细的语义：

- **微服务器**一次至少处理一条消息。**微服务器**没有义务为多个同时未完成的**reqEncrMsgSend**请求提供某种特定的缓冲，尽管它应该谨慎，但它已准备好处理有关它其他响应要求的后续消息。一个安全的**ECI主机**实施方案应该使用**resEncrMsgRecv**作为一次控制流握手。

- **微客户端**与**微服务器**之间转发服务的可靠性与第9.7.2.5.14节中为**reqEncrMsgSend**定义的相同。

9.7.2.5.16 reqEncrTsData消息

C→H reqEncrTsData(ushort**mh,** TsSync **sync,** uint **length,** byte**msg[]) →**
C→H resEncrTsData(ushort**mh)**

- 该消息允许**微服务器**向**ECI主机**提供要转发到**媒质句柄**的**目标微客户端**的数据，以启用内容解密，包括与ECM相关的同步信息。

请求参数定义：

mh: ushort	加密会话的**媒质句柄**。
sync: TsSync	该信息与内容相关的ecmId同步。详情参见表9.7.2.5.16-1。
length: uint	要转发的消息的长度，以字节为单位。
msg: 字节[]	要转发给**微客户端**的消息。

表9.7.2.5.16-1 – TsSync typedef定义

```
typedef struct TsSync {
   uint  ecmId;
   uint  precTime;
} TsSync;
```

字段定义：

ecmId：uint	与内容相关的ECM的标识号，**微客户端**的该数据消息应位于其前。
precTime：uint	就内容播放时间而言，以100ms为单位的实时时间（最大为300秒），该消息应该在将带ecmId的ECM应用于内容解码处理之前。

响应参数定义：

mh：ushort	需要输入消息或连接之IP地址的加密会话的**媒质句柄**。

先决条件请求：

1) 打开媒质句柄会话，会话处于*重加密模式*OfflineStream或OfflineStorage模式，使用*数据格式模式*OfflineDataMode和*同步模式*SyncTs。

先决条件响应：

2) **ECI主机**已准备好接收下一条数据消息。

详细的语义：

- **ECI主机**应确保向**微客户端**提供符合同步要求的数据以及经加密的内容。

- **ECI主机**应适当地缓冲消息的数据（作为与内容的相关数据），并应在[b-ITU-T J Suppl. 7]中提议的期限内对下一个数据做出响应。
- 当工作于**OfflineStream**模式时，**微服务器**可能会在启动的**重加密会话**之前生成一条或多条数据消息。
- 在**OfflineStorage**模式下，**微服务器**将在加密会话结束时产生至多一条数据消息。该数据信息可能处于它应该同步的ECM之前，因此为“离线存储”模式。通常，该数据消息应该由**微客户端**在任何内容和ECM之前进行处理。

9.7.2.5.17 reqEncrTsEcm消息

C→H reqEncrTsEcm(ushort**mh,** uint**ecmId,** uint **length,** byte**ecm[]) →**
C→H resEncrTsEcm(ushort**mh)**

- 该消息允许**微服务器**在下一个加密期发出一个需要解密的ECM区段。

请求参数定义：

mh：ushort	加密会话的**媒质句柄**。
ecmId：uint	出于同步数据消息的目的，由**微服务器**分配的ECM的标识号。
length: uint	**ecm**参数的长度，以字节为单位；ecm具有单区段格式。
ecm: 字节[]	将在下一个加密期内插入的ECM消息。

响应参数定义：

mh：ushort	加密会话的**媒质句柄**。

先决条件请求：

1) 打开**媒质句柄**会话，会话使用同步模式**SyncTs**。

先决条件响应：

2) **ECI主机**已准备好插入下一个ECM。

详细的语义：

- 在收到消息后，**ECI主机**应在某个时隙内将ECM插入传输流中，时隙值在[b-ITU-T J Suppl. 7]中提出。ECM应以合理的间隔重复（如[ISO/IEC 13818-1-1]中所定义的那样）。ECM PID应是一个自由的PID，由**ECI主机**生成。
- **ECI主机**可以更新流中的任何PMT信息，可以反映ECM PID，或者应转发ECM PID信息，以允许**微客户端**稍后恢复所需的解密信息。
- 在一个内容项发生改变与/或另一个更高层的加密发生改变时，**微服务器**可针对相同的即将到来的加密周期发出两个连续的但不同的ECM消息。**ECI主机**应至少为周期的剩余时间插入最后一个。在时移/存储模式下，它应为整个密码周期插入最后一个ECM。

9.7.2.5.18 reqEncrFileData消息

H→C reqEncrFileData(ushort**mh,** byte**syncKid[**MaxUuidLen**],** uint**datalength,** byte **data[]) C→H resEncrFileData(**ushort**mh)**

- 该消息允许**微服务器**向**ECI主机**提供一条消息，该消息将转发给**媒质句柄**的**目标微客户端**进行解密，包括与KeyID相关的同步信息。

请求参数定义：

mh：ushort	加密会话的**媒质句柄**。
syncKid [MaxUuidLen]：字节	KeyId将用于加密文件的下一个“区段”，对之，**微客户端**需要用于解密的相关数据。
datalength：uint	数据的长度，以字节为单位。
data[]：字节	发往**微客户端**的数据用于解密目的。如果数据格式模式为**OfflineDataMode**，则数据格式不透明，并且在数据格式模式为**OfflineIsobmffMode**的情况下，该数据格式是包含在ISOBMFF MOOV或MOOF框中的PSSH框。

响应参数定义：

mh：ushort	加密会话的**媒质句柄**。

先决条件请求：

1) 打开媒质句柄会话，会话处于重加密模式OfflineStream或OfflineStorage模式以及同步模式SyncFile。

先决条件响应：

2) **ECI主机**准备好接收下一条数据消息。

详细的语义：

- **ECI主机**必须确保为任何**目标微客户端**提供符合同步要求的数据以及加密内容。
- **ECI主机**应创建一个包含所提供PSSH框的有效ISOBMFF文件，或者确保数据随文件内容一起传递给**微客户端**，并根据数据同步要求提供给**微客户端**。
- **ECI主机**应适当地缓冲**reqEncrMsgRecv**消息的数据（作为内容的关联数据）。**响应**时间要求的值在[b-ITU-T J Suppl. 7]中提出。
- 当工作于**OfflineStream**模式时，**微服务器**可能会在启动的重加密会话之前生成一条或多条数据消息。
- 在**OfflineStorage**模式下，**微服务器**将在加密会话结束时产生至多一条数据消息。通常，该数据信息必须由**微客户端**在任何内容之前进行处理。

9.7.2.5.19 重加密API的错误代码

表9.7.2.5.19-1 – 重加密API的错误代码

名称	值	描述
ErrEncrAuthInconclusive	1	仅部分地处理认证，并没有定论，但也没有出现任何错误。
ErrEncrAuthFail	-256	无法识别内容项的父认证状态，但已正确执行父认证。
ErrEncrUserMissing	-257	**用户**不会向**微服务器**提供重要输入以继续或继续重加密内容。
ErrEncrCardMissing	-258	需要**智能卡**进行重加密，但无法成功（重新）连接，并协助在合理的时间内对内容进行重加密。
EnrEncrServiceMissing	-259	在解密会话中支持**微服务器**的服务（在**CPE**外部）在合理的时间内不可用。
ErrEncrResourceMissing	-260	处理与/或重加密内容所需的**CPE**内的未指定资源不可用。
ErrEncrMmiMissing	-261	**微服务器**对MMI的访问是必需的，但不可用。
ErrEncrClientAuthError	-262	**微服务器**无法认证**目标微客户端**。
ErrEncrIpNone	-263	**微服务器**无法为**微客户端**通信提供IP地址。

9.7.2.6 微客户端解密API

9.7.2.6.1 概述

微客户端解密API允许**微客户端**解密来自**微服务器**的内容。

发现阶段允许**微客户端**发布解密目标，为之，它可以提供解密服务，并可以提供证书，通过之，作为一个目标，**微服务器**可以创建一个与之的经认证连接。

微客户端必须支持解密模式，以涵盖由其补充**微服务器**提供的加密模式。基于共同支持的模式之一，**微客户端**可以解密服务：这基于通用解密API。

其他支持消息，用于在各种模式下在**微服务器**与**微客户端**之间来回传递要求之解密的数据，是该API的一部分。

在表9.7.2.6.1-1中列出了有关**微客户端**解密API的消息。

表9.7.2.6.1-1 – 解密API消息

消息	类型	方向	标签	描述
setDcrModes	设置	C→H	0x0	**微客户端**通知**ECI主机**它支持的模式（加密模式、数据格式模式和同步模式）。
reqDcrTargets	A	H→C	0x1	**ECI主机**请求**微客户端**提供它可以解密服务的加密目标。
reqDcrTargetCred	A	H→C	0x2	**ECI主机**请求**ECI客户端**为通常用于目标认证的**微服务器**连接提供初始化数据。
reqDcrIpServer	A	C→H	0xA	**微客户端**请求**ECI主机**提供**微服务器**的IP地址，以便进一步进行与**媒质句柄**会话相关的通信。
reqDcrMsgSend	A	C→H	0xB	**微客户端**请求**ECI主机**向**媒质句柄**会话的**微服务器**发送一条消息。
reqDcrMsgRecv	A	H→C	0xC	**ECI主机**向**微客户端**提供一条来自**媒质句柄**会话的**微服务器**的消息。
reqDcrTsData	A	C→H	0xD	**微服务器**向**ECI主机**提供数据，以便将其转发给**媒质句柄**的**目标微客户端**以进行解密，包括与ECM相关的同步信息。
reqEncrFileData	A	C→H	0xF0	**微服务器**为**ECI主机**提供一条消息，以便将其转发给**媒质句柄**的**目标微客户端**以进行解密，包括KeyID相关的同步信息。

9.7.2.6.2 setDcrModes消息

C→H setDcrModes(EciEncrModes **modes)**

- 该消息允许**微客户端**告知**ECI主机**它支持的模式（加密模式、数据格式模式和同步模式）。

请求参数定义：

模式：EciEncrModes	**微客户端**支持的解密模式。ECIEncrModes类型在表9.7.1.5.2-1中指定。

9.7.2.6.3 reqDcrTargets消息

H→C reqDcrTargets()→
C→H resDcrTargets(EncrTarget**target[])**

- 该消息允许**ECI主机**请求**微客户端**提供它可以解密的加密目标。

响应参数定义：

target[]: EncrTarget	**微服务器**可以认证的加密目标列表。TargetClient的类型定义在表9.7.2.5.2-1中指定。

详细的语义：

- **ECI主机**可以基于**目标**来匹配潜在的**目标微客户端**。由应用程序与/或**ECI主机**来定位潜在的候选**微客户端**。

9.7.2.6.4 reqDcrTargetCred消息

H→C reqDcrTargetsCred(EncrTarget**target)→**
C→H reqDcrTargetsCred(uint**credLen**, byte **cred[])**

- 该消息允许**ECI主机**请求**微客户端**提供**微服务器**加密证书。

请求参数定义：

目标： EncrTarget []	**微客户端**必须为其提供实际证书以供**微服务器**对内容进行加密的加密**目标**。

响应参数定义：

credLen： uint	cred参数的长度，以字节数为单位。
cred[]： 字节	以特定于**微服务器**的格式编码的证书，它将加密要由**微客户端**来解密的内容。

详细的语义：

- 该消息允许**ECI主机**请求**微客户端**提供与目标参数相对应的**证书**，以便**微服务器**认可**目标**可以加密**微客户端**的内容。

9.7.2.6.5 reqDcrIpServer消息

C→H reqDcrIpServer(ushort**mh) →**
C→H resDcrIpServer(ushort**mh,** Addrinfo**addr)**

- 该消息允许**微客户端**请求**ECI主机**提供**微服务器**的IP地址，以便进行与**媒质句柄**会话相关的进一步通信。表9.7.2.6.5-1中定义了相关的错误代码。

请求参数定义：

mh: ushort	解密会话的**媒质句柄**，为此请求一条**微服务器**IP地址发送/接收消息。

响应参数定义：

mh: ushort	解密会话的**媒质句柄**，为此提供一条**微服务器**IP地址发送/接收消息。
addr: Addrinfo	该**媒质句柄**的**微服务器**IP协议/地址/端口。

先决条件请求：

1) 媒质句柄会话以OnlineIpMode模式来打开。

先决条件响应：

2) **ECI主机**知晓会话加密的未启动状态。

详细的语义：

- **微客户端**与**微服务器**之间的IP交换特定于**微DRM系统**。这包括协议选择以及用于终止内容流会话上的连接或交换的任何约定。
- 该消息可能在重加密过程尚未启动的**媒质句柄**会话上发出。

表9.7.2.6.5-1 – reqDcrIpServer错误代码

名称	描述
ErrDcrIpNone	参见表9.7.2.6.10-1。

9.7.2.6.6 reqDcrMsgSend消息

C→H reqDcrMsgSend(ushort**mh,** uint **length,** byte**msg[]) →**
C→H resDcrMsgSend(ushort**mh)**

- 该消息允许**微客户端**请求**ECI主机**将消息转发给与**媒质句柄**相关联的**目标微服务器**。

请求参数定义：

mh：ushort	解密会话的**媒质句柄**，为之需转发一条消息给**微服务器**。
length：uint	**msg**字段的长度，以字节为单位。
msg[]：字节	要转发给**微服务器**的消息。

响应参数定义：

mh：ushort	加密会话的**媒质句柄**。

先决条件请求：

1) 媒质句柄会话以OnlineMsgMode模式打开。

先决条件响应：

2) 该消息已被转发到微服务器；ECI主机已准备好接受新的reqDcrMsgSend。

详细的语义：

- **ECI主机**每次应至少能够处理一条消息并转发给**微服务器**。消息应按顺序传送。**ECI主机**没有义务为多个同时发出的**reqDcrMsgSend**请求提供任何特定的缓冲。一个安全的**微客户端**实施方案应该使用**resDcrMsgSend**作为一次控制流握手。
- **微服务器**与**微客户端**之间转发服务的可靠性与第9.7.2.5.14节中对**reqEncrMsgSend**的定义相同。

9.7.2.6.7 reqDcrMsgRecv消息

H→C reqDcrMsgRecv(ushort**mh,** uint **length,** byte**msg[]) →**
C→H resDcrMsgRecv(ushort**mh)**

- 该消息允许**ECI主机**向**微客户端**提供一条来自目标**微服务器**的消息。

请求参数定义：

mh：ushort	解密会话的**媒质句柄**，**微客户端**为之从**微服务器**处获取一条消息。
length：uint	**msg**字段的长度，以字节为单位。
msg[]：字节	要从**微服务器**处接收的消息。

响应参数定义：

mh：ushort	解密会话的**媒质句柄**。

先决条件请求：

1) 媒质句柄会话以OnlineMsgMode模式打开。

先决条件响应：

2) 该消息已由**微客户端**处理，并已准备好接受新的**reqDcrMsgRecv**。

详细的语义：

- **微客户端**每次至少处理一条消息。**微客户端**没有义务为多个同时发出的**reqDcrMsgSend**请求提供任何特定的缓冲，尽管它应该小心，它已准备好处理有关其他响应性要求的后续消息。一个安全的**ECI主机**实施方案应该使用**resDcrMsgRecv**作为一次控制流握手。
- **微客户端**与**微服务器**之间转发服务的可靠性与第9.7.2.5.14节中对**reqEncrMsgSend**的定义相同。

9.7.2.6.8 reqDcrTsData消息

H→C reqDcrTsData(ushort**mh,** uint **length,** byte**msg[]) →**
C→H resDcrTsData(ushort**mh)**

- 该消息允许**ECI主机**为**微客户端**提供（不远的）未来需要的数据，用于解密**媒质句柄**上的内容。

请求参数定义：

mh：ushort	解密会话的**媒质句柄**。
length：uint	要转发消息的长度，以字节为单位。
msg[]：字节	要转发给**微客户端**的消息。

响应参数定义：

mh：ushort	解密会话的**媒质句柄**。

先决条件请求：

1) 打开媒质句柄会话，会话处于*重加密模式*OfflineStream或OfflineStorage，使用*数据格式模式*OfflineDataMode和*同步模式*SyncTs。

先决条件响应：

2) **ECI主机**已准备好接收下一条数据消息。

详细的语义：

- **ECI主机**应确保向**微客户端**提供符合**微服务器**规定之同步要求的数据以及要解密的加密内容。
- **微客户端**应在**OfflineStorage**模式的解密会话开始时至多收到一条数据消息。因此为“离线存储”模式。

9.7.2.6.9 reqDcrFileData消息

H→C reqDcrFileData(ushort**mh,** uint**datalength,** byte **data[])**
C→H resDcrFileData(ushort**mh)**

- 该消息允许**ECI主机**向**微客户端**提供解密**媒质句柄**内容所需的、来自**目标微服务器**的数据。

请求参数定义：

mh:ushort	解密会话的**媒质句柄**。
datalength:uint	数据的长度，以字节为单位。
data[]:字节	发往**微客户端**的数据用于解密目的。如果数据格式模式为**OfflineDataMode**，则数据的格式不透明，并且在数据格式模式为**OfflineIsobmffMode**的情况下，它是包含在ISOBMFF MOOV或MOOF框内的PSSH框。

响应参数定义：

mh：ushort	加密会话的**媒质句柄**。

先决条件请求：

1) **媒质句柄**会话打开，会话处于重加密模式OfflineStream或OfflineStorage模式以及同步模式SyncFile。

先决条件响应：

2) 微客户端已准备好接收下一条数据消息。

详细的语义：

- **ECI主机**必须确保向**微客户端**提供符合同步要求的数据以及加密的内容。
- **ECI主机**可能会提取一个PSSH文件夹，以形成一个有效的ISOBMFF文件，并根据解码ISOBMFF文件的数据同步要求提供给**微客户端**。
- **ECI主机**应在**OfflineStorage**模式下的加密会话结束时最多提供一条数据消息。通常，该数据消息必须由**微客户端**在任何内容前进行处理。

9.7.2.6.10 微客户端解密API的错误代码

- 表9.7.2.6.10-1中列出了**微客户端**解密API的错误代码。

表9.7.2.6.10-1 微客户端解密API相关的错误代码

名称	值	描述
ErrDcrIpNone	-256	**ECI主机**没有用于与**微服务器**通信的IP地址/端口。

9.8 用于内容属性相关资源的API

9.8.1 第9.8节中定义的API列表

图9.8.1-1 – 第9.8节中定义的API的框图

表9.8.1-1列出了第9.8节中涵盖的API，图9.8.1-1说明了使用**ECI体系结构**在第9.8节中定义的API的位置。

表9.8.1-1：内容保护相关资源的API

条款	API名称	描述
9.8.2.3	标准URI消息API	允许**ECI客户端**将与某个特定内容元素相关的标准URI信息传递给**ECI主机**，反之亦然。
9.8.2.4	客户URI API	允许**ECI客户端**将与某个特定内容元素相关的**用户**URI信息传递给**ECI主机**，反之亦然。
9.8.2.5	基本URI API	允许**ECI客户端**将与某个特定内容元素相关的基本URI信息传递给**ECI主机**，反之亦然。
9.8.2.6	输出控制API	允许**ECI客户端**将与某个特定内容元素相关的输出控制信息传递给**ECI主机**，反之亦然。
9.8.2.7	水印API	允许**ECI客户端**将与某个内容元素相关的水印信息传递给**ECI主机**，反之亦然。
9.8.2.8	父控制API	允许**ECI客户端**将与某个特定内容元素相关的父控制职责信息传递给**ECI主机**。
9.8.2.9	内容属性同步API	允许同步各种各样的内容属性更改。
9.8.2.10	父认证API	允许**ECI客户端**将父认证委托给**ECI主机**中的标准父认证功能。
9.8.2.11	父认证委托API	允许**ECI客户端**取消委托的父认证请求。
9.8.2.12	保护控制API	允许**ECI客户端**提供**平台运营商**对输出保护系统的特定控制。

9.8.2 用于访问使用权和父控制资源的API

9.8.2.1 引言

有关**ECI客户端/主机**API的本节允许**ECI客户端**以安全的方式来设置适用于解密内容的权限和条件。

权限和条件API规定以下几方面内容：

- 标准URI（使用权限信息）：由**ECI客户端**生成，并由**ECI主机**用来控制内容在工业标准输出和应用程序上的应用。
- 基本URI：由**ECI客户端**生成，并由**ECI主机**的**高级安全**和硬件子系统用来设置内容的基本使用权限。这允许**ECI客户端**对内容需要使用的基本权限属性使用强健的硬件保护。
- 输出控制：这允许**ECI客户端**选择性地阻止输出，它们在URI的条件下可能是活跃的，但从权限角度来看，它们被认为不适于使用。
- **ECI主机**驱动的水印控制：这允许**ECI客户端**通过**CPE**驻留水印系统利用**ECI客户端**指定的标记来对外传内容进行标记。
- 父控制条件允许**ECI客户端**转发认证父的要求，以授权保护系统（内容输出给它）访问内容。
- 内容属性同步允许同时发生多个内容属性更改，并予识别。
- 父认证功能可以通过**ECI客户端**自身来执行，也可以委托给**ECI主机**中的一个中央行业标准功能。**ECI主机**可以依次选择特定的**ECI客户端**来代表它执行父认证。委托选项用于允许跨多个**ECI客户端**和**ECI主机**来执行单个的父认证。

新权限属性的应用与新控制字的应用安全地链接在一起，以解扰内容。这确保权限适用于与其相关联的内容。

内容属性API有一个设置消息和一个获取消息。**ECI客户端**使用设置消息来解密内容，以表示与计算得到的下一个控制字相关联的内容属性。**微服务器**使用获取函数来对内容进行重加密，以获取输入内容的内容属性，以便为重加密内容的信令内容属性构建适当的认证和信令数据。

作为发现API一部分的API版本有效地对齐了所使用之内容属性的版本。

对每个内容属性，**ECI主机媒质句柄**情形都应为不同内容区段至少保留两个值。特别是对基于文件的解密，对每个内容属性，应至少保留两个用单独的KeyID解码的内容区段。表9.8.2.1-1列出了API函数。权限API函数被分组到独立的API中，以允许独立的版本管理。

表9.8.2.1-1 – 使用权限和父控制API的消息列表

API	消息	类型	方向	标签	描述
ApiStdUri	setDcrStdUri	set	C→H	0x0	为要解密的内容设置标准URI。
ApiStdUri	getEncrStdUri	get	C→H	0x1	获取要重加密的内容的标准URI。
ApiCustUri	setDcrCustUri	set	C→H	0x0	为要解密的内容设置自定义URI。
ApiCustUri	getEncrCustUri	get	C→H	0x1	获取要重加密的内容的自定义URI。
ApiBasicUri	setDcrBasicUri	set	C→H	0x0	为要解密的内容设置基本URI。
ApiBasicUri	getEncrBasicUri	get	C→H	0x1	获取要重加密的内容的基本URI。
ApiOC	setDcrOutputCtl	set	C→H	0x0	为要解扰的内容设置输出控制限制。
ApOC	getEncrOutputCtrl	get	C→H	0x1	获取要重加密的内容的输出控制限制。
ApiDcrMark	getDcrMarkSyst	get	H→C	0x0	获得支持的标记系统。
ApiDcrMark	setDcrMarkMeta	set	C→H	0x1	设置标记系统控制值。
ApiDcrMark	getDcrMarkMeta	get	H→C	0x2	读取标记系统属性。
ApiDcrMark	setDcrMarkBasic	set	C→H	0x3	为要解扰的内容设置基本标记有效载荷。
ApiDcrMark	setDcrMarkExt	set	C→H	0x4	为要解扰的内容设置扩展标记有效载荷。
ApiPar	setDcrParCtl	set	C→H	0x0	为要解扰的内容设置父控制条件。
ApiPar	getEncrParCtrl	get	C→H	0x1	获取要解密的内容的父控制条件。
ApiCpSync	setCpSync	set	C→H	0x0	**ECI客户端**表示当前的一组内容属性是一致的，并且可用于即将被控制字解扰的内容。
ApiCpSync	reqCpChange	req	H→C	0x1	**ECI主机**表示要重加密的内容的内容属性即将发生变化。
ApiParAuth	reqParAuthChk	req	C→H	0x0	请求**ECI主机**代表**ECI客户端**执行父认证。
ApiParAuth	reqParAuthChkCan	req	C→H	0x1	取消先前向主机提出的父认证请求。
ApiParAuth	reqParAuthCid	req	H→C	0x2	为要解码（未来）的内容项目请求父 pin码授权。这可能会触发父身份验证对话。
ApiParAuthDel	reqParAuthDel	req	H→C	0x0	**ECI主机**将父身份验证委托给**ECI客户端**。
ApiParAuthDel	reqParAuthDelCan	req	H→C	0x1	**ECI主机**取消先前向**ECI客户端**提出的父认证请求。
ApiProtCtrl	getProtSystCtrl	get	C->H	0x0	**ECI客户端**从**ECI主机**获取输出保护系统及其对SRM（系统可更新性消息）和设备ID阻止服务的支持的列表。
ApiProtCtrl	reqSrmMsg	req	C->H	0x1	**ECI客户端**为输出保护系统提供SRM。
ApProtCtrl	reqInfoDevId	req	H->C	0x2	**ECI主机**提供设备的ID，作为解密会话的一部分，输出保护系统向该设备提供受保护的内容。
ApiProtCtrl	reqBlockDevId	req	C->H	0x3	**ECI客户端**提供一个设备ID，作为解密会话的一部分，输出保护系统不应向其提供任何内容。
ApProtCtrl	setBlockProtSyst	set	C->H	0x4	**ECI客户端**指示保护系统被认为不足以保护解密会话的内容。

9.8.2.2 安全问题和同步

ECI规范允许上述内容属性信息由**ECI主机**来认证，以防止未经授权地处理此信息。该机制还可确保将适当的权限设置应用于与其关联的内容。这在[ITU-T J.1014]中进行定义。

对于内容属性信息，**ECI主机**可以使用高级安全块中的密钥，代表**ECI客户端**，来促进对权限信息的验证，从而确保最高级别的验证完整性。**ECI客户端**需要为此适当使用**ECI主机**的AS服务。这也在[ITU-T J.1014]中进行定义。

如果内容属性要求在输出上应用特定的输出保护属性，但**ECI主机**无法提供此类输出保护属性（或更安全或受约束的版本），则**ECI主机**不得输出内容并提供适当的消息给**用户**。在**ECI生态系统**合规性制度中将提供更多详细信息。

9.8.2.3 标准URI消息API

9.8.2.3.1 setDcrStdUri消息

C→H setDcrStdUri(ushort**mh,** byte **keyId**[MaxUuidLen], StdUri**stdUri**)

- 该消息将与**keyId**相关联的标准URI设置为**uri**。

参数定义：

mH: ushort	要解码的内容的**媒质句柄**。
keyId[MaxUuidLen]**:** 字节	KeyID作为网络字节顺序的UUID，在文件格式解码的情况下，URI适用于这个UUID，字节0承载TS格式流的0x00（偶数）或0x01（奇数）以指示对下一个CW的适用性。
stdUri: StdUri	内容的标准URI在表9.8.2.3.1-1中定义。这些字段的语义对应于[ETSI TS 103 205]和[b-CI Plus]中定义的那些字段。

表9.8.2.3.1-1 – 标准URI类型规范

```
typedef struct StdUri {

   uintMajorVersion: 4;
uinttmc: 1;   /* trick_mode_control_info in [CI+ v1.4] */
 unit reserved1: 3;
   uint aps: 2; /* aps_copy_control_info in [CI+ v1.4] */
   uint emi: 2; /* emi_copy_control_info in [CI+ v1.4] */
   uintict: 1;  /* ict_copy_control_info in [CI+ v1.4] */
   uintrct: 1;  /* rct_copy_control_info in [CI+ v1.4] */
   uint reserved2: 1; /* reserved bit */
   uint dot: 1; /* dot_copy_control_info in [CI+ v1.4] */
   uint rl: 8;     /* rl_copy_control_info in [CI+ v1.4] */

} StdUri;

The following rules shall apply (expressions over field shall evaluate to True)
in line with [CI+v1.4]
emi == 0b00 || rct == 0b0
emi == 0b11 || (dot == 0b0 &&rl == 0x00)
emi == 0b01 || tmc == 0b0

The protocol_version field value 0x03 is defined for the definition above; other
values are reserved for future use.
```

StdUri字段的语义：

MajorVersion:uint: 4	此标准URI的主要版本。**ECI客户端**应将MajorVersion设置为0b0000。**ECI主机**应执行所有版本，直至其达到此字段的合规级别，并将任何高的值解释为未实施的URI，因此没任何使用权限可用。
reserved1: unit: 3	保留位。应由**ECI客户端**设置为0b000，并应被符合此版本stdUri的**ECI主机**所忽略。
reserved2: unit: 1	保留位。应由**ECI客户端**设置为0b0，并应被符合此版本stdUri的**ECI主机**所忽略。
其他字段	语义与上述结构定义中CI Plus v1.4 URI [ETSI TS 103 205]的指示字段的定义相同。

详细的语义：

- 对于传输流解扰模式，URI应适用于要解码的内容，使用适用于下一个解密密钥的密钥。在[ITU-T J.1014]的第8.2.4.7节中定义有关解密密钥计算的细节。
- **ECI客户端**应处于解密模式。

9.8.2.3.2 getEncrStdUri消息

C→H StdUrigetEncrStdUri(ushort**mh,** byte **keyId**[MaxUuidLen])

- 该消息为即将到来的内容设置标准URI。

属性定义：

- 标准URI如表9.8.2.3.1-1中所定义。

参数定义：

mH：ushort	要加密的内容的**媒质句柄**。
keyId：字节 [MaxUuidLen]	KeyID作为网络字节顺序的UUID，在文件格式解码的情况下，URI适用于这个UUID，字节0承载TS格式流的0x00（偶数）或0x01（奇数）以指示对下一个CW的适用性。

详细的语义：

- **ECI客户端**应处于加密模式。

9.8.2.4 客户URI API

9.8.2.4.1 setDcrCustUri消息

C→H setDcrCustUri(ushort**mh,** byte **keyId**[MaxUuidLen], unit **custUriLen**, byte ***custUri)**

- 该消息将设置一个自定义URI，将**keyId**与uri相关联。

参数定义：

mH：ushort	要解码的内容的**媒质句柄**。
keyId：字节 [MaxUuidLen] t	KeyID作为网络字节顺序的UUID，在文件格式解码的情况下，URI适用于这个UUID，字节0承载TS格式流的0x00（偶数）或0x01（奇数）以指示对下一个CW的适用性。
custuriLen：unit	自定义URI字段的长度，以字节为单位。
custUri：字节 *	表9.8.2.4.1-1定义了内容的自定义URI。字节0和字节1将作为自定义URI格式的msB和lsB。保留字节0和字节1的所有值，除了0x80和0x00，它们对后面的字节将具有特定的应用意义。

表9.8.2.4.1-1 – 自定义URI类型规范

名称	值字节0,1	描述
CustUriPrivate	0x80，0x00	字节1后面的字节的含义是私有的。通过**ECI客户端**与**微服务器**或保护系统之间的其他通信来定义其余字段的适当解释。
RFU	其他	保留以供未来使用。

详细的语义：

- 对于传输流解扰模式，URI应适用于要解码的内容，使用适用于下一个解密密钥的密钥。在[ITU-T J.1014]的第8.2.4.7节中定义有关解密密钥计算的细节。
- 对一个控制字，最多允许设置四个单独的自定义URI。
- **ECI客户端**应处于解密模式。

9.8.2.4.2 getEncrCustUri消息

C→H custUrigetEncrCustUri(ushort**mh,** byte **keyId**[MaxUuidLen], unit **custUriMaxLen)**

- 该消息为即将到来的内容获取自定义URI。

属性定义：

- 自定义URI如表9.9.1-1中所定义。

参数定义：

mH：ushort	要加密的内容的**媒质句柄**。
keyId：字节 [MaxUuidLen]	KeyID作为网络字节顺序的UUID，在文件格式解码的情况下，URI适用于这个UUID，字节0承载TS格式流的0x00（偶数）或0x01（奇数）以指示对下一个CW的适用性。
custUriMaxLen：uint	自定义URI结果的最大长度（以字节为单位）；任何额外的内容将被截断。

详细的语义：

- **ECI客户端**应处于加密模式。

9.8.2.5 基本URI API

9.8.2.5.1 setDcrBasicUri消息

C→H setDcrBasicUri(ushort**mh,** byte **keyId**[MaxUuidLen], BasicUri**basicUri**)

- 该消息将与**keyId**相关联的基本URI设置为**basicUri**。基本URI为解密内容提供简化但高度稳健的权限管理。

参数定义：

mH：ushort	要解码的内容的**媒质句柄**。
keyId[MaxUuidLen]：字节	KeyID作为网络字节顺序的UUID，在文件格式解码的情况下，URI适用于这个UUID，字节0承载TS格式流的0x00（偶数）或0x01（奇数）以指示对下一个CW的适用性。
basicUri：BasicUri	表9.8.2.5.1-1中定义了内容的基本URI。这些字段的语义对应于[ETSI TS 103 205]中定义的那些字段。

表9.8.2.5.1-1 – 基本的URI类型规范

```
typedef byteBasicUri;
```

名称	位	描述
BasicUriVersion	7	基本URI的大版本。如果**ECI主机**尚未实现版本，则**ECI主机**不允许解密和使用内容。值0b0定义版本0.A；保留其他值且不被允许。
BasicUriV0_0Ext	2..6	保留以供未来使用，未在v0.0中使用。为此字段定义的唯一值是0b00000。其他值不允许。仅实现基本Uri v0.0的**ECI主机**应忽略该字段的值：即，这可用于v0.0未来后向兼容的扩展，例如，以v0.0权限控制宽松形式。
BasicUriV0_0	0,1	基本URI版本0.0。在表9.8.2.5.1-2中定义了该字段的值和含义。

表9.8.2.5.1-2：基本的URI V0.0定义

名称	值	描述
NoBasicProtection	0b00	没有任何通过基本URI的权限控制。
RedistributionProtected	0b01	加密应开启，应关闭防止重播。
ViewOnly	0b10	加密应开启，应开启防止重播。
ViewOnlyStrict	0b11	加密应开启，应开启防止重拨，输出应限制在特定的合格的（安全的）输出。

详细的语义：

- 对于传输流解扰模式，URI应适用于要解码的内容，使用适用于下一个解密密钥的密钥。在[ITU-T J.1014]的第8.2.4.7节中定义有关解密密钥计算的细节。
- 基本URI使**ECI客户端**能够通过权限实施方案进行控制，它处于**ECI主机**支持的最高级别的稳健性上。它提供了对两种保护机制的控制：加密，确保内容在任何输出或存储介质上始终都是加密的，并防止重放，从而确保加密的内容只能在实时连接中解扰（即不能存储）。具体请参见[ITU-T J.1015]。
- **ECI客户端**应处于解密模式。

9.8.2.5.2 getEncrBasicUri消息

C→H BasicUrigetEncrBasicUri(ushort**mh,** byte **keyId**[MaxUuidLen])

- 该消息为即将到来的内容获取基本URI。

属性定义：

- 基本URI如表9.8.2.5.1-1中所定义。

参数定义：

mH: ushort	要加密的内容的**媒质句柄**。
keyId[MaxUuidLen]**:** 字节	KeyID作为网络字节顺序的UUID，在文件格式解码的情况下，URI适用于这个UUID，字节0承载TS格式流的0x00（偶数）或0x01（奇数）以指示对下一个CW的适用性。

详细的语义：

- **ECI客户端**应处于加密模式。

9.8.2.6 输出控制API

9.8.2.6.1 setDcrOutputCtl消息

C→H setDcrOutputCtl(ushort**mh,** byte **keyId**[MaxUuidLen], ushort**ocVector**)

- 将与**keyId**相关联的输出控制设置设置为**ocVector**。

参数定义：

mH: ushort	要解码的内容的**媒质句柄**。
keyId[MaxUuidLen]**:** 字节	KeyID作为网络字节顺序的UUID，在文件格式解码的情况下，URI适用于这个UUID，字节0承载TS格式流的0x00（偶数）或0x01（奇数）以指示对下一个CW的适用性。
ocVector: unit	标准输出的输出控制向量依照表9.8.2.6.1-1的定义。

表9.8.2.6.1-1 – 输出控制向量规范

名称	位	描述
MajorVersion	7	ocVector参数的版本。值0b0是为版本1定义的。保留任何其他值且不被允许。 如果**执行主版本**1的**ECI主机**接收到的值不是0xb0，则表示不允许有任何输出。
OcAnyOther	6	下面列出的任何输出资质准则不涵盖任何其他的**ECI主机**输出。如果在这些输出上允许值为0b0的输出，那么若值为0b1，则不允许输出。 该位的值改变了下面字段的编码。如果值为0b0，则输出限制应如下所示。如果值为0b1，则编码应按位反转。即，如果允许在IP连接上输出OcAnyOther == 0b1和OcIP == 0b1。 见注2。
OcIP	0	如果值为0b0，则允许任何IP连接上的输出，并且值不允许为0b1。
OcUSB	1	如果值为0b0，则允许任何USB连接上的输出；如果值为0b1，则不允许输出。 对此的先决条件是解密的内容不受任何**ECI**认可之输出保护系统与/或在解密**ECI客户端**控制下的**ECI微DRM系统**的保护。
OcDtcpIp	2,3	如果值为0b0，则允许DTCP-IP受保护连接上的输出，并且值不允许为0b1。
OcHdcp	3,4	任何HDCP受保护的输出。 对于OcAnyOther等于0b0： • 值0b00：允许HDCP保护的输出。 • 值0b01：如果HDCP版本低于2.2，则不允许输出；如果HDCP版本为2.2或更高，则允许输出。 • 值0b10：保留，该值是不允许的；**ECI主机**应将该值解释为等于0b11。 • 值0b11：不允许任何HDCP保护的输出。 对于OcAnyOther等于0b1： • 值0b00：不允许任何HDCP输出。 • 值0b01：保留，**ECI主机**应将该值解释为等于0b00。 • 值0b10：如果HDCP版本为2.2或更高，则允许输出；如果HDCP输出版本低于2.2，则不允许输出。 • 值0b11：允许任何HDCP保护的输出。 HDCP 2.2或更高版本意味着不得将版本低于2.2的HDCP的应用程序应用于内容；即不允许输出到符合HDCP1.x、HDCP2.0或HDCP2.1的中继器或符合HDCP1.x的设备。参见[b-HDCP2.3]中定义的“类型1内容流”。
OcWm	5	如果该位的值为0b1，则只有运用通过相关内容元素中的**CPE**插入的水印，才允许输出解码的内容元素。 见注3。

表9.8.2.6.1-1 – 输出控制向量规范

名称	位	描述
OcDtcp	6,7	任何受DTCP保护的输出。 对于OcAnyOther等于0b0： • 值0b00：允许受DTCP保护的输出。 • 值0b01：如果DTCP版本低于2，则不允许输出；如果DTCP版本为2或更高，则允许输出。 • 值0b10：保留；此值是不允许的。**ECI主机**应将此值解释为等于0b11。 • 值0b11：不允许受DTCP保护的输出。 对于OcAnyOther等于0b1： • 值0b00：不允许DTCP输出。 • 值0b01：保留，**ECI主机**应将此值解释为等于0b00。 • 值0b10：如果DTCP版本为2或更高版本，则允许输出，如果DTCP输出版本小于2，则不允许输出。 • 值0b11：允许任何受DCTP保护的输出。
OCDownResHDCP1	8	如果**OcHdcp**字段值为0b01，并且内容缩小到720p或更小（如果该字段的值是0b0），则允许在受HDCP1.x保护的输出上输出内容；如果该字段的值为0b1，则不允许这样做。
reserved	9-13	符合该版本规范的**ECI客户端**应将该字段的值设置为0b00000。符合该版本规范的**ECI主机实施方案**可能会忽略该字段。
注1 – 模拟输出控制由标准URI**dot** 和**ict**字段来有效提供。 注2 – OcAnyOther有效地将输出控制字段从输出黑名单（当值等于0b0时）切换到输出白名单（当值等于0b1时）。如果输出字段为0b1，则意味着它实际上“在列表上”。 注3 – 适用于此应用的水印系统可能需要经过批准。具有广播或多播功能的**ECI主机**应支持水印。作为将水印系统应用于基于ECI的CPE的定义的一部分，应有可能唯一地标识芯片组，例如，通过从水印恢复ChipsetID。		

如果多个ocVector字段适用于一个输出（例如，受DTCP-IP保护的IP输出），则应使用最严格的条件。

详细的语义：

- **ECI客户端**应处于解密模式。

9.8.2.6.2 getEncrOutputCtrl消息

C→H uintgetEncrOutputCtrl(ushort**mh,** byte **keyId**[MaxUuidLen])

- 该消息为即将到来的内容获取输出控制。

属性定义：

- 输出控制如表9.8.2.6.1-1中所定义。

参数定义：

mH: ushort	要加密的内容的**媒质句柄**。
keyId[MaxUuidLen]**:** 字节	KeyID作为网络字节顺序的UUID，在文件格式解码的情况下，URI适用于这个UUID，字节0承载TS格式流的0x00（偶数）或0x01（奇数）以指示对下一个CW的适用性。

详细的语义：

- **ECI客户端**应处于加密模式。

9.8.2.7 水印API

9.8.2.7.1 概述

标记API允许**ECI客户端**通过**ECI主机**发现可用的嵌入式（水印）标记系统，然后与这些系统进行“设置”控制对话。标记系统可以仅与有限数量的**ECI客户端**进行对话，并且可以同时仅标记有限数量的**媒质句柄**会话。

标记系统可能希望与授权的**ECI客户端**进行交互。这种授权可以通过标记系统定义的授权对话，使用setMarkMeta和getMarkMeta消息来建立。

ECI客户端可以通过成功完成约定对话，来保留对标记系统的访问。该**ECI客户端**（由其**ECI客户端**ID来标识）应保持与标识系统的接合，直至从**CPE**中移去它或者它脱离为止。

9.8.2.7.2 getDcrMarkSyst消息

C→H MarkSystDescr**getDcrMarkSyst()**

- 该消息允许**ECI客户端**读取可用标记系统的描述符。

属性定义：

结果类型MarkSystDescr应符合表9.8.2.7.2-1中的定义。

表9.8.2.7.2-1 – MarkSystDescr类型定义

```
#define MaxMarkSystDescr 16;

typedef ushortMarkId;  /* ECI Marking ID allocated to a marking system */
// markId values: 0x8xxx are used for proprietary marking systems.
//                0x0000 shall mean no marking system
//                All other values are reserved by ECI, allocation of new
//                IDs and their publication is defined elsewhere.

typedef struct MarkSystDescrElem {
   MarkIDmarkId;         /* ID of the marking system */
   ucharnrClients;      /* number of Clients that can still be supported */
   uchar  markSystFlags  /* field as defined below */
} MarkSystDescr[MaxMarkSystDescr];
// Any available marking systems shall be listed as the first elements
// of MarkSystDescr. The remaining elements shall use markId==0x0000.


// markSystFlags:
//   bit 0 signals authorization required (0b1) or not (0b0)
//   bit 1 signals scrambled stream support (0b1) or not (0b0)
//   bit 2 signals multi simultaneous stream support (0b1) or not (0b0)
//   other bits are reserved and shall be ignored by Clients complying
//   to this Recommendation
```

9.8.2.7.3 setDcrMarkMeta消息

C→H **setDcrMarkMeta(**MarkID**markId,** uchar **index,** byte **data[32])**

- **该消息使ECI主机能够**为标记系统设置控制（元）数据。

参数定义：

markId: MarkID	标记系统ID，要为其设置属性定义。
index: uchar	要为标记系统设置的子属性。
data[32]:字节	适用于由索引指示的子属性的值。

9.8.2.7.4 getDcrMarkMeta消息

C→H byte[32] getDcrMarkMeta(MarkID**markId**, uchar**index)**

- 该消息使**ECI客户端**能够为标记系统获取控制（元）数据。

属性定义：

- 具有标记ID**markID**的子属性**索引**系统的元数据。

参数定义：

markId: MarkID	标记系统ID，要为其读取属性定义：结果类型MarkSystDescr应符合表9.8.2.7.4-1中的定义。
index: uchar	标记系统的子属性读取。

9.8.2.7.5 SetDcrMarkBasic消息

C→H setDcrMarkBasic(ushort**mH,** byte **keyId**[MaxUuidLen], MarkID**markId**, byte **data**[16])

- 该消息使**ECI客户端**可以设置最大值。128位数据利用指定的密钥来标记待解扰内容。

参数定义：

mH: ushort	要解码的内容的**媒质句柄**。
keyId[MaxUuidLen]**:** byte	KeyID作为网络字节顺序的UUID，在文件格式解码的情况下，URI适用于这个UUID，字节0承载TS格式流的0x00（偶数）或0x01（奇数）以指示对下一个CW的适用性。
markId: MarkID	标记系统ID。
data[16]**:** 字节	128位值。

9.8.2.7.6 SetDcrMarkExt消息

C→H setDcrMarkExt(ushort**mH,** byte **keyId**[MaxUuidLen], ushort**markId**, uint**dataLen**, byte **data**[]**)**

- 对要用指定的密钥标记的待解扰内容，**该消息使ECI客户端能够**为标记系统设置一个扩展的有效载荷。

参数定义：

mH:ushort	要解码的内容的**媒质句柄**。
keyId:byte[MaxUuidLen]	KeyID作为网络字节顺序的UUID，在文件格式解码的情况下，URI适用于这个UUID，字节0承载TS格式流的0x00（偶数）或0x01（奇数）以指示对下一个CW的适用性。
markId:ushort	用于标记内容的标记系统ID。
datalen:uint	数据字段的长度。
Data[]**:**字节	标记系统的有效载荷数据。

9.8.2.8 父控制API

9.8.2.8.1 setDcrParCtl消息

C→H setDcrParCtl(ushort**mH,** byte **keyId**[MaxUuidLen], ParCond**pC)**

- 对要用指定的密钥解扰的mH的内容，**该消息使ECI客户端能够**设置父等级条件（**pC**）。

参数定义：

mH: ushort	要解码的内容的**媒质句柄**。
keyId[MaxUuidLen]**:** 字节	KeyID作为网络字节顺序的UUID，在文件格式解码的情况下，父控制条件**pC**适用于这个UUID，字节0承载TS格式流的0x00（偶数）或0x01（奇数）以指示对下一个CW的适用性。
pC: ParCond	适用于内容的父控制条件。ParCond的定义参见表9.8.2.8.1-1。

表9.8.2.8.1-1 – 父条件类型规范

```
typdef struct ParCond {
   byte basicCondition; /* see 9.8.2.8.1-2*/
   byte extendedQualifier[16];
} ParCond;
```

表9.8.2.8.1-2 – 父条件基本条件定义

名称	位	描述
AuthRequired	7	0b1表示在呈现内容之前需要进行父认证。0b0表示根据extendedQualifier可能需要进行父认证。
ToggleBit	6	该位在流中更替，以指示对该位值改变的新的父认证要求。
Reserved	4,5	应设置为0b00。
QualifierFormat	0..3	指示extendedQualifier字段的格式。 值0x0指示“无值”，将extendedQualifier字段设置为零。 值0x1指示ExtendedQualifier字段包含如[ETSI EN 300 468]中定义的DVB父评级描述符。剩余字节的值应为零。如果适用的国家/地区的所需评级超过父设定的限制（如DVB父评级描述符的语义所定义的那样），即使AuthRequired == 0b0，也需要父认证。 保留值0x2..0xF以供未来使用。

详细的语义：

- **ECI**允许父等级认证条件与内容一起进行传送，作为保护解扰内容的系统的一项义务。
- **ECI客户端**应处于解密模式。

9.8.2.8.2 getEncrParCtrl消息

C→H ParCondgetEncrParCtrl(ushort**mh,** byte **keyId**[MaxUuidLen])

- 该消息使**ECI客户端**能够为即将到来的内容获取父控制条件。

属性定义：

- 父控制URI如表9.8.2.8.1-2中所定义。

参数定义：

mH：ushort	要加密的内容的**媒质句柄**。
keyId[MaxUuidLen]: 字节	KeyID作为网络字节顺序的UUID，在文件格式解码的情况下，URI适用于这个UUID，字节0承载TS格式流的0x00（偶数）或0x01（奇数）以指示对下一个CW的适用性。

详细的语义：

- **ECI客户端**应处于加密模式。

9.8.2.9 控制属性同步API

9.8.2.9.1 setCpSync消息

C→H setCpSync(ushort**mH,** byte **keyId**[MaxUuidLen]**)**

- 该消息向**ECI主机**发信号通知，即将由keyId指示的内容区段将具有通过标准URI、自定义URI、基本URI、输出控制、水印和父控制API设置的内容属性。

参数定义：

mH：ushort	要解码的内容的**媒质句柄**。
keyId[MaxUuidLen]: 字节	KeyID作为网络字节顺序的UUID，在文件格式解码的情况下，父控制条件**pC**适用于这个UUID，字节0承载TS格式流的0x00（偶数）或0x01（奇数）以指示对下一个CW的适用性。

详细的语义：

- 该消息将触发**ECI主机**，来为内容属性中即将发生的变化做适当准备。这将包括通过与该**媒质句柄**会话的**入口/出口连接**，来向任何**微服务器**发送reqCpChange消息。
- **ECI客户端**应处于解密模式。

9.8.2.9.2 reqCpChange消息

H→C reqCpChange(ushort**mh,** byte **keyId**[MaxUuidLen])

- 该消息触发**微服务器**，来准备内容属性更改，它基于解密内容之内容属性的最近未来值，它由**微服务器**重加密。

属性定义：

- 父控制URI如表9.8.2.8.1-2中所定义。

参数定义：

mH：ushort	要加密的内容的**媒质句柄**。
keyId[MaxUuidLen]: 字节	KeyID作为网络字节顺序的UUID，在文件格式解码的情况下，URI适用于这个UUID，字节0承载TS格式流的0x00（偶数）或0x01（奇数）以指示对下一个CW的适用性。

详细的语义：

- **ECI客户端**应处于加密模式。
- **ECI客户端**应为解密流中即将到来的、与KeyId相关的内容获取内容属性，并为新内容准备一个新的加密设置（可能需要一个新的CW）。

9.8.2.10 父认证API

9.8.2.10.1 概述

父批准的认证可以由**ECI客户端**使用MMI会话直接来执行。或者，**ECI客户端**可以请求**ECI主机**执行（或已经执行）父认证，以便协调pin码管理以及通过在**ECI主机用户**接口中自然地集成pin请求来改善用户接口体验。反过来，通过**ECI主机**的**用户**可以在可用的候选者中选择一个**ECI客户端**，以使用第9.8.2.11节中定义的父认证委托API ParAuthDel来执行父认证。在处理许多内容项的**ECI客户端**不能委托其父认证但可代表**ECI主机**执行父认证的情况下，这会很有用。

该API还允许**ECI客户端**在打开媒质会话之前开始对内容项的父认证，例如，用于未来记录事件的父认证。

9.8.2.10.2 标准父认证功能

本节为基于4字符pin码的标准父评价功能定义了一组要求，如果**ECI客户端**提出请求，则**ECI主机**应能执行，或者在它通过父认证委托API提供这种服务的情况下，**ECI客户端**将代表**ECI主机**执行。

如果这样的功能至少提供本节定义之机制的父认证完整性，则**ECI主机**或**ECI客户端**可提供一个替代的认证功能，而不是本节后面所述的认证功能。

以下功能适用于基于标准PIN码的父认证机制：

1) 父认证基于至少4个字母数字字符（来自由至少10个字符（如数字）组成的最小集合）的pin码。
2) pin码设置应受pin码自身或主认证机制的保护，它保护对资产的访问或材料服务的价值，它们被认为非常不适合由未成年人来访问，对之，需要保护内容。
3) 任何适用的父等级限制设置都应通过上面2）中的pin码或主认证机制来保护。
4) 对潜在的主认证机制的需求应创建一个认证完整性，至少对本节中定义的pin码机制，不需要基于主认证机制。
5) 在购买主机时，用于父等级的初始pin码或用于通过主认证进行认证的方式，应仅被传递给所有者。
6) 在安装新客户端时，**运营商**只能将初始pin码或通过主认证进行认证的方式，传递给所有者。
7) **制造商**或其代理可提供一种方式来将pin码重置为其初始值，或者提供一种服务，通过之，所有者可将pin码设置为将仅传递给所有者的一个新值。
8) **运营商**可提供一种方式将PIN码重置为其初始值，或者提供一种服务，通过该之，所有者可将pin码设置为将仅传递给所有者的一个新值。
9) 如果在15分钟内连续5次认证失败，则至少在15分钟内父认证功能应拒绝进行新的认证。
10) 通过应用常规**用户**软件、运行于**CPE**或任何**用户**接口或常规接口上的下载应用程序，不可能恢复或重置pin码。

9.8.2.10.3 reqParAuthChk消息

C→H reqParAuthChk(ushort**mH) →**
C→H resParAuthChk(ushort**mH,** bool **ok)**

- 该消息允许**ECI客户端**请求**ECI主机**使用**ECI主机**的标准父验证功能来执行父验证检查（参见第9.8.2.10节），并将结果返回给响应消息。

请求参数定义：

mH：ushort	**媒质句柄**要解码的内容。

响应参数定义：

mH：ushort	**媒质句柄**要解码的内容。
ok: bool	真对应认证成功，否则为假，包括超时。

详细的语义：

- 每个**媒质句柄**只有一个未完成的父认证检查应由**ECI主机**来区分。在前一个响应或取消之前，在同一个**媒质句柄**上发出第二个请求将导致两个相同的**响应**。
- **reqParAuthChkECI主机**应该使用超时值来请求父认证，如果没有人或者不愿意执行[b-ITU-T J Suppl. 7]中提议的认证，则该认证将在合理的时间内终止。

9.8.2.10.4 reqParAuthChkCan消息

C→H reqParAuthChkCan(ushort**mH) →**
H→C resParAuthChkCan(ushort**mH)**

- **ECI客户端**取消以前对**ECI**进行父验证的任何请求。

请求参数定义：

mH：ushort	要解码内容的**媒质句柄**。

响应参数定义：

mH：ushort	要解码内容的**媒质句柄**。

后置条件响应：

1) 在resParAuthChkCan消息之前，ECI主机可向ECI客户端返回对先前reqParAuthChk消息的响应，但此后不会。

9.8.2.10.5 reqParAuthCid消息

H→C reqParAuthCid(uint**cidLength,** byte **cid**[]**) →**
C→H resParAuthCid(bool **ok)**

- 该消息允许**ECI主机**请求**ECI客户端**对由**cid**识别的未来内容项目执行任何所需的认证。

请求参数定义：

cidLength:uint	cid参数的长度。
cid[]: 字节	识别父验证的内容（如果需要的话）。第一个字节表示内容识别参数的格式，如表9.8.2.10.5-1所定义。

表9.8.2.10.5-1 – 内容标识格式

名称	值	描述
CidDvbEvent	0×01	DVB事件标识。在cid中跟随字节的字节具有以下序列值：原始网络ID（2字节）、传输流ID（2字节）、服务ID（2字节）、事件ID（2字节），如[ETSI EN 300 468]中的定义。序列中的所有2字节字段都按网络顺序表示（最重要的字节在前）。
RFU	其他	保留以供未来使用。

响应参数定义：

ok: bool	如果父验证成功或不需要验证，则为真。

详细的语义

- **ECI客户端**应维护一个非易失的内容标识记录，它们已通过该功能进行验证。在缺少存储空间的情况下，它可丢弃最老的记录和未来将不再要求的记录。该内容识别缓冲的最低要求在[b-ITU-T J Suppl. 7]中提出。

表9.8.2.10.5-2列出了相关的错误代码。

表9.8.2.10.5-2 – TS媒质的媒质会话API的错误代码

名称	值	描述
ErrParAuthCidUnknOk	1	无法识别内容项目的父验证状态，但已执行父验证并发现是正确的。

如果对所需网络资源的访问不可用，则也可以返回上述错误状态。

9.8.2.11 父认证委托API

9.8.2.11.1 概述

该API允许**ECI客户端**指示它可以执行第9.8.2.10.2节中定义的标准父验证功能，并允许**ECI主机**将PIN码认证委托给此类**ECI客户端**。

ECI客户端可以在**ECI客户端**初始化时使用配置API来指示对委托验证API的支持。

注 – 与此同时，例如出于如商业、安全或法律等方面的考虑，**ECI客户端**可选择不委派自己的父认证。

ECI主机应提供设置功能，允许**用户**选择**ECI主机**进行标准父控制验证，或将标准父控制验证委托给其中一个提供此功能的**ECI客户端**。

9.8.2.11.2 reqParAuthDel消息

H→C reqParAuthDel(ushort**mh**) →
C→H resParAuthDel(ushort**mH,** bool **ok)**

- 该消息允许**ECI主机**请求**ECI客户端**代表其为mH上的内容执行委托的父认证。

请求参数定义：

mH: ushort	要解码的内容的**媒质句柄**。

响应参数定义：

mH: ushort	要解码的内容的**媒质句柄**。
ok: bool	如果父认证成功，则为真，如果不成功或超时，则为假。

详细的语义：

- 每个**媒质句柄**只有一个未完成的父认证检查应由**ECI客户端**来区分。在响应或取消前一个之前，在同一个**媒质句柄**上发出第二个请求将导致两个相同的响应。
- **ECI客户端**应对请求父认证使用一个超时值，如果没有任何人存在或不愿意执行[b-ITU-T J Suppl. 7]中提议的认证，则该认证将在合理的时间内终止。

9.8.2.11.3 setParAuthDelCan消息

H→C reqParAuthDelCan(ushort**mH)** →
C→H resParAuthDelCan(ushort**mH)**

- 该消息允许**ECI主机**取消委托的父认证请求。

响应参数定义：

mH：ushort	要解码的内容的**媒质句柄**。

响应参数定义：

mH：ushort	要解码的内容的**媒质句柄**。

后置条件响应：

- 在resParAuthDelCan消息之前，**ECI主机**可能会向**ECI客户端**返回对先前reqParAuthDel消息的响应，但此后不会。

9.8.2.12 保护系统控制API

9.8.2.12.1 引言

由**ECI客户端**解密的内容可以提供给**CPE**的不同输出。输出通常由输出保护系统提供保护。输出保护系统可能有从**ECI客户端**接受系统可更新性消息（SRM）的选项，并在其设备ID（在输出保护系统范畴内）列为遭到破坏的情况下，为**ECI客户端**提供选项，以阻止输出到通过输出保护系统连接的设备上。

一个保护系统可以支持多个输出。

9.8.2.12.2 getProtSystCtrl消息

C->H getProtSystCtrl()

- 该消息使**ECI客户端**可以读取CPE支持的输出保护系统的列表、其版本以及它们对SRM（系统可更新性消息）和设备ID阻止服务的支持。

表9.8.2.12.2-1 – 保护控制阵列规范

```
typedef struct ProtCtrlElem {
   ushort protSysType;     // protection system type according to table sect-2
   uint   srmSupp:4;       // level of support for SRMs according to table sect-3
   uint   devIdSupp:1;     // 0b0 means no support for device ID services,
                           // 0b1 means support for device ID services
   uint   reserved:11;     // reserved; shall have value 0b00000000000
} ProtCtrlElem;

#define MaxProtCtrlArr 32
typedef ProtCtrlElemProtCtrlArr[MaxProtCtrlArr];
   // A protection system as listed in the array may protect multiple outputs.
   // Each value of ProtCtrlElem except where protSustType=0x0000 shall appear
   // only once in ProtCtrlArr.All ProtCtrlElem with ProtColElem unequal 0x0000
   // shall be in the lowest index elements of ProtCtrlArr,
   // values equal 0x0000 shall be at the end of the array
```

表9.8.2.12.2-2 – 输出保护系统类型值

名称	值	输出保护系统类型
OpNoProtSyst	0x0000	无输出保护系统
OpHDCP_1	0x0010	HDCP版本1
OpHDCP_21	0x0011	HDCP版本2.0或2.1
OpHDCP_22	0x0012	HDCP版本2.2或更高
OpDTCP_1	0x0020	DCTP版本1
OpDTCP_2	0x0021	DTCP版本2或更高
OpDTCP_IP1	0x0030	DTCP IP
所有权	0x8xxx	可能在本规范范围之外定义
保留	其他值	保留以供未来使用

表9.8.2.12.2-3 – SRM支持值

保护	值	输出保护系统类型
SrmNone	0x0	不支持SRM
SrmProtSysSpecV1	0x1	根据输出保护系统规范版本1（但不更高）支持SRM
SrmProtSysSpecV2	0x2	根据输出保护系统规范版本2（但不更高）支持SRM
SrmProtSysSpecV3	0x3	根据输出保护系统规范版本3（但不更高）支持SRM
SrmProtSysSpecV4	0x4	根据输出保护系统规范版本4（但不更高）支持SRM
保留	0x5..0xC	保留以供未来使用
所有权	0xD-0xF	可能在本规范范围之外定义

语义：

- 设备ID服务支持意味着保护系统应支持使用reqBlockDevId、resBlockDevId消息来识别和阻止与设备的任何受保护连接。
- 输出保护功能的配置在客户端的“生命周期”内应是静态的。

9.8.2.12.3 reqSrmMsg消息

C→H reqSrmMsg(ushort**protSysType**, uint**srmLen**, byte **srmData**[])→
H→C resSrmMsg()

- 该消息使**ECI客户端**可将SRM发送到保护系统类型。

请求参数定义：

protSysType[]: ushort	该SRM面向的保护系统类型。注意：SRM可能适用于同一系列保护系统的多种类型。在这种情况下，仅将SRM发送到主机一次就足够了，而不需要将每种类型都发送到主机。
srmLength: uint	SRM的长度
srmData:byte[]	SRM

先决条件请求：

- 未发送前一个**reqSrmMsg**消息，或者未收到之最后一个**reqSrmMsg**消息的**resSrmMsg**消息。

详细的语义：

- ECI主机应尽快发送**resSrmMsg**。

表9.8.2.12.3-1 – reqSrmMsg错误代码

名称	描述
ErrReqSrmMsgOverflow	参见第9.8.2.12.7节。

9.8.2.12.4 reqInfoDevId消息

H→C reqInfoDevId(ushort**mh**, ushort**protSysType**, uint**lenDevId**, byte **devId**[])→
C→H resInfoDevId(ushort**mh**)

- 该消息允许**ECI主机**指示设备（通过**devId**）使用解密会话**mh**中的保护系统**protSysType**将设备可以解密的内容发送到该设备。

请求参数定义：

mh: ushort	解密会话的媒质句柄，对之，使用带有**devId**的设备。
protSysType: ushort	用于保护将要传送给**devId**的内容的保护系统 – 参见[b-ITU-T J Suppl. 7]中的表6.4.2-1。
lenDevId: uint	**devId**字段的长度（以字节为单位）。
devId[]: byte	设备ID – 在补充规范中定义特定的编码。

响应参数定义：

mh: ushort	解密会话的**媒质句柄**，对之，提供响应。

先决条件请求：

- 未发送mh会话中的前一个reqInfoDevId消息，或者未收到至mh会话中最后一个reqInfoDevId消息的resInfoDevId消息。

详细的语义：

- **ECI主机**应尽快发送连接到**mh**会话输出的每个设备的**devId**。

表9.8.2.12.4-1 – reqInfoDevId错误代码

名称	描述
ErrReqInfoDevOverflow	参见第9.8.2.12.7节。

9.8.2.12.5 reqBlockDevId消息

C→H reqBlockDevId(ushort**mh**, ushort**protSysType**, uint**lenDevId**, byte **devId**[])→
H→C resBlockDevId(ushort**mh**)

- 该消息允许**ECI客户端**阻止带有**devId的**设备，在解密会话**mh**中，使用保护系统**protSysType**来将解密内容发送到该设备。

请求参数定义：

mh: ushort	解密会话的**媒质句柄**，对之，使用带有**devId**的设备。
protSysType: ushort	用于保护将要传送给**devId**的内容的保护系统 – 参见[b-ITU-T J Suppl. 7]中的表6.4.2-1。
lenDevId: uint	**devId**字段的长度（以字节为单位）。
devId[]**:** byte	设备ID – 在补充规范中定义特定的编码。

响应参数定义：

mh: ushort	解密会话的**媒质句柄**，对之，提供响应。

先决条件请求：

- 未发送mh会话中的前一个reqBlockDevId消息，或者未收到至mh会话中最后一个reqBlockDevId消息的resBlockDevId消息。

语义：

- 在一个有效的**reqBlockDevId**上，**ECI主机**应以**ErrReqOkNoId**进行响应（参见表9.3.4-1），并确保阻止输出至带有**devId的**设备。

9.8.2.12.6 setBlockProtSyst消息

C→H setBlockProtSyst(ushort**mh**, ushort**protSysType**bool **block)**

- 该消息允许**ECI客户端**在解密会话**mh**中阻止使用保护系统**protSysType**发送的所有解密内容。

参数定义：

mh: ushort	解密会话的**媒质句柄**，对之，应阻止内容。
protSysType: ushort	用于保护将要传送给**devId**的内容的保护系统 – 参见[b-ITU-T J Suppl.2]中的表6.4.2-1。7]
block: bool	如果将阻止内容，则为**真**，否则为**假**。

语义：

- 对mh上的一个protSysType，在将阻止从真设为假的情况下，如果protSysType允许（如getProtSystCtrl所指示），则应使用reqInfoDevId，由ECI主机来传送mh上用于输出的该protSysType的所有devID。

9.8.2.12.7 保护系统控制API的错误代码

- 表9.8.2.12.7-1中列出了保护系统控制API的错误代码。

表9.8.2.12.7-1 – 保护系统控制API相关的错误代码

名称	值	描述
ErrReqSrmMsgOverflow	-256	**ECI主机**指示它尚不能接受下一个**ReqSrmMsg**消息。
ErrReqInfoDevOverflow	-257	**ECI客户端**指示它尚不能接受下一个**ReqInfoDev**消息。

9.9 用于ECI客户端和应用程序通信的API

9.9.1 在本节中定义的API列表

表9.9.1-1列出了本节中涵盖的各API。

表9.9.1-1 – 用于ECI客户端和应用程序通信相关资源的API

条款	API名称	描述
9.9.2	客户端间通信API	使**ECI客户端**能够建立至另一个**ECI客户端**的一条直接通信路径。

9.9.2 客户端间通信API

9.9.2.1 概述

ECI主机以入口/出口信息、URI和内容形式，在**ECI客户端**与**ECI客户端**之间提供标准化的信息交换环境。**ECI客户端**可以相互通信，以便提供额外的（目前不是**ECI**定义的）功能。**ECI客户端**可以通过发现资源注册其基本能力和意愿，以支持客户端与客户端之间的通信（参见第9.4.2节）。系统初始化后，它们可以读取其他**ECI客户端**的身份，包括建立的**入口/出口连接**。**ECI客户端**可以向一个潜在的对应端打开一个通信信道（称为管道），并通过管道来交换消息。双方都可以取消管道。**ECI客户端**的管道在**ECI主机**停止与/或重新初始化其对应端的**ECI客户端**时关闭。

ECI主机提供**ECI客户端**身份，它们使用**ECI客户端**提供的**ECI证书链**来认证。**ECI客户端**应提供额外的独立认证机制，以防与对应端的通信可能导致安全隐患。

如果一个解码内容的**ECI客户端**与另一个随后重加密该内容的**ECI客户端**（**微服务器**）之间进行通信，则建立管道的建议是该管道由**微服务器**来启动（打开）。

表9.9.2.1-1显示了客户端间通信API的消息。

表9.9.2.1-1：客户端间通信API消息

消息	类型	方向	标签	描述
getIccMaxClients	S	C→H	0x0	**ECI客户端**读取**ECI主机**可支持的最大**ECI客户端**数量。
reqIccSystemReady	A	H→C	0x1	**ECI主机**通知**ECI客户端**所有**ECI客户端**都已被初始化。
getIccClientInfo	S	C→H	0x2	**ECI客户端**读取系统中另一个**ECI客户端**的身份和连接状态。
reqIccPipeOpen	A	C→H	0x3	请求打开至另一个**ECI客户端**的管道。
reqIccPipeOpenReq	A	H→C	0x4	来自另一个**ECI客户端**的打开管道的请求。
reqIccPipeCancel	A	C→H	0x5	**ECI客户端**取消管道。
reqIccPipeClose	A	H→C	0x6	**ECI主机**通知**ECI客户端**，对应方的管道已关闭。
reqIccPipeMsgSend	A	C→H	0x7	**ECI客户端**向管道的对应方发送一条消息。
reqIccPipeMsgRecv	A	H→C	0x8	**ECI客户端**收到来自管道对应方的一条消息。

9.9.2.2 getIccMaxClients消息

C→H uintgetIccMaxClients()

- 获取**ECI主机**可支持的最大**ECI客户端**数量。

属性定义：

- 表示**ECI主机**可支持的最大**ECI客户端**数量的无符号整数。

9.9.2.3 reqIccSystemReady消息

H→C reqIccSystemReady()

- ECI主机通知ECI客户端所有其他ECI客户端已被初始化。

语义：

- 该消息在系统初始化时提供，以向注册到此API的所有**ECI客户端**指示，它有可能开始读取客户端信息注册表，并尝试打开至其他**ECI客户端**的管道。
- 结果中的ConnId字段反映了**ECI客户端**与潜在对应方的**入口/出口连接**的最新状态。这些可能会发生变化。
- 不需要结果消息。

9.9.2.4 getIccClientInfo消息

C→H ClientInfogetIccClientInfo(ushort**clientId)**

- **ECI客户端**读取系统中另一个**ECI客户端**的身份和连接状态。

参数定义：

clientId:ushort	用于设置管道的客户端ID。该标识符在系统的生命周期中不会改变。它在重新初始化时发生变化。

属性定义：

- connectionID是一个动态属性。
- ClientInfo是一种结构，提供指定**ECI客户端**的身份以及与该**ECI客户端**的任何入口/出口连接。它在下面定义。

ClientInfo的类型定义：

```
#define MaxConnId 32

typedef struct ClientInfo {
ECI_Operator_IdoperatorId;
ECI_Platform_Operation_IdplatformOperationId;
ECI_Vendor_Id vendorId;
  union {
ECI_Client_Series_IdclientSeriesId;
ECI_Client_IdclientId;
  } client;
ushortconnId[MaxConnId];
}
```

字段定义：

operatorId:ECI_Operator_Id	**ECI客户端**的运营商标识。
platformOperationId: ECI_Platform_Operation_Id	**ECI客户端**的平台操作ID。
client: union	ECI_Client_Series_Id或ECI_Client_Id。clientSeriesId和clientId的类型字段定义这是一个clientSeriesId还是一个clientId。
VendorId: ECI_Vendor_Id	**ECI客户端**的供应商ID。
clientSeriesId: ECI_Client_Sesies_Id	**ECI客户端**的客户端系列ID。
clientId: ECI_Client_Id	**ECI客户端**的客户端ID。
connId: ushort[MaxConnId]	连接ID数组；值0xFFFF表示一个空数组条目。空数组条目都在数组的末尾。

9.9.2.5 reqIccPipeOpen消息

C→H reqIccPipeOpen(ushort**clientId,** byte**protocolId[16]) →**
H→C resIccPipeOpen(ushort**clientId)**

- 该消息使**ECI客户端**能够请求**ECI主机**打开一个至另一个**ECI客户端**的管道。

请求参数定义：

clientId:ushort	请求管道的客户端的ID。
protocolId[16]:字节	要使用的消息协议的ID。这应是一个UUID [IETF RFC 4122]，其数组中的八位字节按网络顺序。

结果参数定义：

clientId:ushort	请求打开管道的客户端的ID。

先决条件响应：

- 打开管道或返回一个错误代码。表9.9.2.5-1列出了相关的错误代码。

表9.9.2.5-1 – reqIccPipeOpen错误代码

名称	描述
ErrIccPipeOpenReject	参见表9.9.2.11-1。
ErrIccPipeOpenNoConn	
ErrIccPipeOpenProtocol	
ErrIccPipeOpenNotReady	

9.9.2.6 reqIccPipeOpenReq消息

H→C reqIccPipeOpenReq(ushort**clientId,** byte**protocolId**[16]**) →**
C→H resIccPipeOpen(ushort**clientId)**

- 该消息使**ECI客户端**能够接收来自另一个**ECI客户端**的传入请求，以通过**ECI主机**打开管道。

请求参数定义：

clientId:ushort	请求管道的客户端的ID。
protocolId[16]**:** 字节	要使用的消息协议的ID。这应是一个UUID [IETF RFC 4122]，八位字节按网络顺序，如同字节。

结果参数定义：

clientId: ushort	请求管道的客户端的ID。

语义：

- clientId的响应值应与请求值相同。

先决条件响应：

- **ECI客户端**可能会拒绝该管道。错误代码与打开管道的错误代码相同，并且透明地传送给请求者。它们列在表9.9.2.5-1中。

9.9.2.7 reqIccPipeCancel消息

C→H reqIccPipeCancel(ushort**clientId) →**
H→C resIccPipeCancel(ushort**clientId)**

- 该消息使**ECI客户端**能够指示**ECI主机**，它想终止管道。

请求参数定义：

clientId: ushort	被取消管道的客户端的ID。

结果参数定义：

clientId:ushort	被取消管道的客户端的ID。

语义：

- clientId的响应值应与请求值相同。

先决条件响应：

- 管道终止：请求管道取消的**ECI客户端**将不会收到来自管道的更多消息。

详细的语义：

- 如果管道未打开，则这在无错误情况下进行处理。

9.9.2.8 reqIccPipeClose消息

H→C reqIccPipeClose(ushort**clientId,** uint **reason) →**
C→H resIccPipeClose(ushort**clientId)**

- 该消息使**ECI主机**能够通知**ECI客户端**，对方部分的管道已关闭。

请求参数定义：

clientId:ushort	已关闭管道的客户端的ID。
reason: uint	关闭管道的原因。值在表9.9.2.11-1中列出。

表9.9.2.8-1 – reqIccPipeClose原因值

名称	值	描述
IccPipeCloseCancel	0×01	对方部分使用reqIccPipeCancel消息来关闭管道。
IccPipeCloseStop	0×02	因对应方**ECI主机**终止而使管道被**ECI主机**关闭。**ECI客户端**有可能随后被重新初始化。
RFU	其他	保留以供未来使用。

结果参数定义：

clientId：ushort	已关闭的管道客户端的ID。

先决条件请求：

- 没有更多的消息将通过管道来发送。

先决条件响应：

- **ECI客户端**不会尝试通过（关闭的）管道来发送新消息。

9.9.2.9 reqIccPipeMsgSend消息

C→H reqIccPipeMsgSend(ushort**clientId,** uint**msgId,** uint**dataLen,** byte **data[])→**
H→C resIccPipeMsgSend(ushort**clientId)**

- 该消息使**ECI客户端**能够向管道的对应方发送消息。表9.9.2.11-1列出了相关的错误代码。

请求参数定义：

clientId:ushort	消息发送到的客户端的ID。
msgId: uint	消息的ID。所有负值和零都保留；所有正值都是特定于应用程序的（意思是在发送方和接收方情形下定义）。
dataLen: uint	数据参数的长度，以字节数为单位。这不得超过32 768。
data[]: 字节	消息的数据字段。

结果参数定义：

clientId：ushort	管道客户端的ID。

先决条件请求：

- 下一条reqIccMsgSend消息只有在收到相同管道的前一条resIccMsgSend消息后才会发送。

表9.9.2.9-1 – reqIccPipeMsgSend错误代码

名称	描述
ErrIccPipeClosed	参见表9.9.2.11-1。

9.9.2.10 reqIccPipeMsgRecv消息

H-→C reqIccPipeMsgRecv(ushort**clientId,** uint**msgId,** uint**dataLen,** byte **data[])→**
C→H resIccPipeMsgRecv(ushort**clientId)**

- 该消息使**ECI客户端**能够从管道的对应端接收消息。

请求参数定义：

clientId: ushort	从中接收消息的客户端的ID。
msgId:uint	消息的ID。保留所有的负值和零；所有正值都特定于应用程序（依据发送方和接收方的情形来定义含义）。
dataLen: uint	数据参数的长度，以字节数为单位。它不得超过32 768。
data: byte[]	消息的数据字段。

结果参数定义：

clientId:ushort	管道客户端的ID。

先决条件请求：

- 下一条reqIccMsgRecv消息只有在收到相同管道的前一条resIccMsgRecv消息后才会发送。

9.9.2.11 客户端间通信的错误代码

表9.9.2.11-1列出了客户端间通信API的错误代码。

表9.9.2.11-1 – 客户端间通信的错误代码

名称	值	描述
ErrIccPipeOpenReject	-256	对应端拒绝了管道。
ErrIccPipeOpenNoConn	-257	由于没有建立与ECI客户端的**入口/出口连接**，对应端拒绝了管道。
ErrIccPipeOpenProtocol	-258	对应端拒绝为管道提议的协议。
ErrIccPipeOpenNotReady	-259	对方端不处于准备接受管道的状态。稍后重新尝试建立管道是合适的。
ErrIccPipeClosed	-260	管道关闭。

10 强制的和可选的ECI主机功能

10.1 引言

ECI技术规范支持一系列用于媒质消费的**CPE**技术解决方案。决于**CPE制造商**决定在其设备中实现的前端、核心和后端功能。对于设备的前端和后端功能，**制造商**很可能只实现符合其硬件/协议栈的**ECI** API。为了给**用户**提供灵活性，表10.2-1列出了不同类别**CPE**的所有强制的（m）、可选的（o）和有条件的（c）API。

10.2 用于不同类型CPE设备的强制的和可选的ECI功能列表

表10.2-1给出了不同类型**CPE**设备的强制的和可选的**ECI**功能列表。若干API的实现是有条件的，这取决于**CPE**设备中某些硬件/软件组件的可用性。

表10.2-1 – 强制的和可选的ECI功能列表

API	条款	主机	条件（如适用）	解扰客户端	微服务器	微客户端
主机接口发现	9.4.2	M		M	M	M
MMI	9.4.3	M		O	O	O
IP	9.4.4	C	如果支持IP链接	O	O	O
HTTP(S)	9.4.4.6	M		O	O	O
文件系统	9.4.5	M		O	O	O
计时器和时钟	9.4.6	M		O	O	O
电源管理	9.4.7	M		O	O	O
国家和语言设置	9.4.8	M		O	O	O
高级安全通用	9.5.2.2	M		M	M	M
高级安全解密	9.5.2.3	M		M	n.a.	M
高级安全出口	9.5.2.4	C	用于记录或网关	O	n.a.	O
高级安全加密	9.5.2.5	C	用于记录或网关	n.a.	M	na
智能卡	9.5.3	C	用于支持的SC阅读器	O	O	O
数据轮播	9.5.4	C	用于广播网络	O	O	O
解密（参见注释）	9.6.2	M		M	n.a.	M
出口连接	9.7.2.3	C	用于记录或网关	O	n.a.	O
入口连接	9.7.2.4	C	用于记录或网关	n.a.	M	n.a.
重加密（参见注释）	9.7.2.5	C	用于记录或网关	n.a.	M	n.a.
微客户端解密	9.7.2.6	M		O	n.a.	M
国家和语言设置	9.4.8	M		O	O	O
标准URI	9.8.2.3	M		M	M	M
客户URI	9.8.2.4	M		M	M	M
基本URI	9.8.2.5	M		M	M	M
输出控制	9.8.2.6	M		M	M	M
水印	9.8.2.7	C	用于具有广播或多播功能的设备	O	n.a.	O
父控制	9.8.2.8	M		M/O	M/O	M/O
内容属性同步	9.8.2.9	M		M	M	M
父认证	9.8.2.10	M		O	n.a.	O
父认证委托	9.8.2.11	M		O	n.a.	O
客户端间通信	9.9.2	M		O	O	O

注： 可专门为**微服务器**和解密客户端指定时隙。

时隙本身在技术上是相同的，但所需的**AS**资源和相关的解扰功能是不同的。

发现API不提供允许**ECI主机**检测**ECI客户端**可解密或加密文件与/或传输流格式媒质数据的机制。这种信令由setDcrMhMatch消息的decryptId参数的mhType字段提供（参见第9.6.2.2.2节）。对于重加密，这种发现由setEncrModes消息的EciEncrModes参数提供（参见第9.7.2.5.3节）。

- 一个仅使用**ECI**的设备应提供至少2个VM实例和AS时隙。
- 支持PVR功能的**ECI主机**应至少支持一个用于**微服务器**的附加容器（**VM**实例）和**AS时隙**。如果此类**ECI主机**也提供存储内容的播放功能，则它应支持至少一个用于**微客户端**的附加容器（**VM**实例）和**AS时隙**，以便对重加密的内容进行解码。
- 支持网络网关功能的**ECI主机**应至少支持一个用于**微服务器**的附加容器（**VM**实例）和**AS时隙**。

附件A

ECI主机的密码函数

（本附件是本建议书不可分割的组成部分。）

A.1 散列函数

本建议书中的散列函数全部基于[NIST FIPS 197]中定义的SHA256。

第5.2节中的散列函数等同[NIST FIPS 197]中定义的SHA-256()。

c函数asHash(uchar *data, uintdatalength, resultLength, uchar *result)使用起始于dataLength长度数据的八位字节作为*dataIn*八位字节串，计算八位字节串resultOut作为resultLength/8八位字节串，并将之存为结果，根据：

resultOut = BS2OSP(truncate(SHA-256(OS2BSP(*dataIn*)),resultLength)))

resultLength应该是8的倍数。截断应该是将比特串（参数1）左截断为长度（参数2）比特的函数。

BS2OSP和OS2BSP是按照[ITU-T J.1014]第7节中的定义将比特串转换为八位字节串的函数，反之亦然。

A2 非对称加密

非对称加密和解密操作在[ITU-T J.1014]的第12.4节中进行定义。

A.3 对称加密

除非提供了有关AES应用的特定应用程序参考，否则本建议书中定义的AES加密应如[NIST FIPS 197]中所定义。

除非提供了带AES的CBC特定应用参考，AES的CBC应用应如[NIST Block 2001]中所定义。如果没有定义，则应使用初始化向量0。

除非提供了带AES的CTR特定应用参考，否则AES的CTR应用应如[NIST Block 2001]中所定义。如果没有定义，则应使用初始化向量0。

A.4 随机数生成

本建议书中定义的随机数生成须符合[ITU-T J.1014]附件A中定义的规范。

附件B

互操作性参数

（本附件是本建议书不可分割的组成部分）

B.1 引言

本附件定义了与**CPE**中的资源需求有关的参数。遵守这些要求有助于**ECI客户端**之间的互操作性，网络和**CPE**提供的**ECI**安全服务。

B.2 撤销列表长度

CPE应保留足够的NV存储空间来存储每个可撤销项目以下长度的**撤销列表**，如表B.2-1所示。**ECI TA**应确保发布的**ECI TA** RL遵守这些限制。

表B.2-1 – 撤销列表最大长度

撤销列表	最大number_of_ids
制造商RL	500
主机RL	500
供应商RL	500
ECI客户端RL	500
运营商RL	500
平台操作RL	500

B.3 ECI客户端图像大小

ECI主机应该为其支持的每个**ECI客户端**时隙提供至少500KB的**ECI客户端图像**存储空间。

B.4 广播轮播配置参数

ECI定义从广播轮播下载的所有项目的最大采集时间tCdownloadScenario，以便允许合适的**ECI主机**设计。tCdownloadScenario参数反映了实际的下载时间；因此轮播重复率应至少为此的三倍，从而确保**ECI主机**在这些限制内进行下载。广播公司应提供足够的带宽来支持所需的重复率。

ECI还出于缓冲区分配目的定义了最大模块大小。

表B.4-1定义了tCdownloadScenario以及**ECI主机**应设计处理的最大模块大小。

表B.4-1 – ECI轮播的最大下载场景周期和模块大小

表类型	tCdownloadScenario	最大模块大小
ECI客户端图像	5分钟	500KB
ECI客户端撤销数据	5分钟	每桶100KB
平台操作证书链	10秒	50KB
平台操作撤销数据	5分钟	每桶100KB
ECI主机撤销数据	5分钟	每桶100KB
AS设置数据	2分钟	每桶20KB

附件C

ECI主机API概述

（本附件是本建议书不可分割的组成部分。）

表C-1定义了第9.3.1节中定义的**MsgApiTag**的值。

表C-1：ECI API的编号方案

API	条款	MsgApiTag值	最高的API版本	弃用的API版本
主机接口发现	9.4.2	0x0001	0x0000	无
MMI	9.4.3	0x0002	0x0000	无
IP	9.4.4	0x0003	0x0000	无
HTTP(S)	9.4.4.6	0x0004	0x0000	无
文件系统	9.4.5	0x0005	0x0000	无
计时器和时钟	9.4.6	0x0006	0x0000	无
电源管理	9.4.7	0x0007	0x0000	无
国家和语言设置	9.4.8	0x0008	0x0000	无
高级安全通用	9.5.2.2	0x0009	0x0000	无
高级安全解密	9.5.2.3	0x000A	0x0000	无
高级安全出口	9.5.2.4	0x000B	0x0000	无
高级安全加密	9.5.2.5	0x000C	0x0000	无
智能卡	9.5.3	0x000D	0x0000	无
数据轮播	9.5.4	0x000E	0x0000	无
解密	9.6.2	0x000F	0x0000	无
出口连接	9.7.2.3	0x0010	0x0000	无
入口连接	9.7.2.4	0x0011	0x0000	无
重加密	9.7.2.5	0x0012	0x0000	无
微客户端解密	9.7.2.6	0x0013	0x0000	无
标准URI	9.8.2.3	0x0014	0x0000	无
客户URI	9.8.2.4	0x0015	0x0000	无
基本URI	9.8.2.5	0x0016	0x0000	无
输出控制	9.8.2.6	0x0017	0x0000	无
水印	9.8.2.7	0x0018	0x0000	无
父控制	9.8.2.8	0x0019	0x0000	无
内容属性同步	9.8.2.9	0x0020	0x0000	无
父认证	9.8.2.10	0x0021	0x0000	无
父认证委托	9.8.2.11	0x0022	0x0000	无
客户端间通信	9.9.2	0x0023	0x0000	无

附件D

内容属性定义的前向兼容性

（本附件是本建议书不可分割的组成部分。）

内容属性必须使用硬件或低级固件以高度可靠的方式来实现，并可能在生产SOC后变得复杂、昂贵或者不可能更改或更新。尽管存在此类升级限制，本节解释了为此类内容属性创建演进路径的方法。

未来可能需要新的内容属性与/或现有内容属性的扩展功能。这可能包括扩展表示内容属性值的位数。旧**ECI主机**中的内容属性实施方案不知道新的功能，更新它通常是不可行的。**ECI主机**中内容属性的定义使与新内容属性功能相关的最大前向兼容性得以实现。

ECI主机将为所有输入值定义一个行为，并忽略不用于创建定义之行为的任何字段扩展。即未来内容属性的每个值都将在所有不执行所有扩展的**ECI主机**上单独定义一个行为，包括符合第一个内容属性版本的**ECI主机**。使用该原则，可以分配新的内容属性值，并全面了解先前版本的**ECI主机**实施方案会产生哪些行为。如果新的内容属性应该有两个（或更多个）不同的选项用于旧**ECI主机**的后向兼容解释，则可以在新的内容属性定义中分配两个（或更多个）保留值，它们具有相同的新内容属性语义，但每个值带有一个合适的（但不同的）后向兼容解释。

字段扩展的一个示例是为输出控制API中的新输出类型X实例一个待定义的新输出控制字段。这被分配给在版本1中保留的位5。它可以使用与OcIP字段相同的语义。**ECI客户端**的任何先前实施方案都将分配该字段0。旧**ECI主机**的解释如下所示：

- 如果OcAnyOther == 0b0，则允许OutputX；
- 如果OcAnyOther == 0b1，则不允许OutputX。

当OcX == 0b0时，这完全对应于新**ECI主机**实施方案中的语义。但是，当OcX == 0b1时，输出许可将与之前OcX==0b0时的配置相反，从而允许新**ECI主机**和新**ECI客户端**组合中的新功能。请注意，取决于OcAnyOther的字段值的反向解释可确保任何未定义字段的值0具有其自然的含义：对OcAnyOther == 0b0允许的最大值（允许其他输出）以及对OcAnyOther= 0b1允许的最小值（不允许其他输出）。

反之亦然，重要的是不使用最新内容属性定义的**ECI客户端**不会无意中解决它们不知道的新内容属性功能，或者更糟的是，出于私人目的，使用这些可能未分配的值，这基于以下事实，即这些值已在所有的**ECI主机**中定义了行为。这种不适当的用法通常会给未来将这些数值纳入**ECI**定义之目的造成严重障碍。因此，本规范明确禁止**ECI客户端**应用未分配的内容属性值。

具体而言：对具有多个值的字段，各保留值都将在**ECI主机**中拥有已定义的行为，但**ECI客户端**不得使用保留值。

内容属性定义中任何未分配的子字段都应拥有一个在**ECI主机**中定义的已定义行为，它对应其中一个已定义的内容属性值。通常，**ECI主机**应忽略这样的子字段，即**ECI主机**仅根据定义的字段来解释内容属性值。通常，**ECI客户端**应将值0分配给这样的子字段。未分配子字段等于零策略的任何偏差都将由内容属性定义的一个版本进行预定义。

符合对应内容属性定义的**ECI主机**将忽略任何字段扩展，分配值的**ECI客户端**将把值0赋予此类字段扩展。

附录I

按字母顺序列出所有可用的API消息

（本附录非本建议书不可分割的组成部分。）

附录I中列出的API消息摘自本建议书第9节的下列表格，并列在表I-1中。

表I-1 – 给出不同API消息的表列表

API	条款	API类别
主机接口发现API	9.4.2.1-1	
用户接口API	9.4.3.1-1	
IP套接字API	9.4.4.3.1-1	
UDP套接字API	9.4.4.4.1-1	
TCP套接字API	9.4.4.5.1-1	
HTTP获取API	9.4.4.6.1-1	
文件打开/关闭API	9.4.5.2.1-1	通用的API
文件访问API	9.4.5.3.1-1	
文件目录服务API	9.4.5.4.1-1	
计时器API	9.4.6.2.1-1	
时钟API	9.4.6.3.1-1	
电源转换API	9.4.7.2-1	
从待机状态唤醒API	9.4.7.3-1	
国家/语言设置API	9.4.8.1-1	
高级安全通用API	9.5.2.2.1-1	
高级安全解密API	9.5.2.3.1-1	
高级安全出口API	9.5.2.4.1-1	
高级安全加密API	9.5.2.5.1-1	ECI特定的API
智能卡会话管理API	9.5.3.6.1-1	
智能卡通信API	9.5.3.6.1-1	
数据轮播采集API	9.5.4.1-1	
媒质句柄解密会话API	9.6.2.2.1-1	
出口连接API	9.7.2.3.1-1	
入口连接API	9.7.2.4.1-1	
重加密API	9.7.2.5.1-1	
解密API	9.7.2.6.1-1	
使用权利和父控制API	9.8.2.1-1	
客户端间通信API	9.9.2.1-1	

表I-2按字母顺序列出了所有API消息。

表I-2：按字母顺序排列的所有API消息列表

编号	消息	API	条款	类型	方向	描述
1	callAsNextKeySession	高级安全通用	9.5.2.2.3	S	C→H	更改为会话的下一个随机密钥。
2	callCardGetProp	**智能卡**	9.5.3.6.5	S	H→C	获取卡通信属性/参数。
3	callCardSessionPrio	**智能卡**	9.5.3.5.3	S	C→H	设置**智能卡**会话优先级。
4	callCardSetProp	**智能卡**	9.5.3.6.4	S	H→C	设置卡通信参数。
5	callFileDataLog	文件系统	9.4.5.3.6	S	C→H	在缓冲文件末尾添加数据。
6	callLocaltime	时钟	9.4.6.3.3	S	C→H	将时间整数值转换为本地时间。
7	getApis	接口发现	9.4.2.2	S	C→H	获取可用的主机API。
8	getApiVersions	接口发现	9.4.2.3	S	C→H	获取主机**API**的可用版本。
9	getAsClientRnd	高级安全通用	9.5.2.2.13	S	C→H	为**ECI客户端**应用程序获取一个新的随机数。
10	getAsSC	高级安全通用	9.5.2.2.14	S	C→H	获取会话中内容的当前加扰控制字段状态。
11	getAsSessionLimitCounter	高级安全通用	9.5.2.2.10	S	C→H	获取会话的当前限制计数器值。
12	getAsSessionRk	高级安全通用	9.5.2.2.9	S	C→H	获取会话的随机密钥值。
13	getAsSlotRk	高级安全通用	9.5.2.2.8	S	C→H	获取**AS时隙**的随机密钥值。
14	getCardConnStatus	**智能卡**	9.5.3.5.4	S	H→C	提供卡连接状态的状态。
15	**getChipsetId**	高级安全通用	**9.5.2.2.16**	S	C→H	**密钥阶梯块**的ChipsetID值
16	getDcrMarkMeta	内容属性	9.8.2.7.4	S	H→C	阅读标记系统属性。
17	getDcrMarkSyst	内容属性	9.8.2.7.2	S	H→C	获取支持的标记系统。
18	getDcrTsSource	解密TS源控制	9.6.2.3.6.2	S	C→H	**ECI客户端**获取TS的源。
19	getEncrStdUri	内容属性	9.8.2.3.2	S	C→H	获取需重加密之内容的标准URI。
20	getEncrBasicUri	内容属性	9.8.2.5.2	S	C→H	获取需重加密之内容的基本URI。
21	getEncrCustUri	内容属性	9.8.2.4.2	S	C→H	获取需重加密之内容的自定义URI。
22	getEncrOutputCtrl	内容属性	9.8.2.6.2	S	C→H	获取需重加密之内容的输出控制限制。
23	getEncrParCtrl	内容属性	9.8.2.8.2	S	C→H	获取需解密之内容的父控制条件。
24	getIccClientInfo	客户端间通信	9.9.2.4	S	C→H	**ECI客户端**读取系统中另一个**ECI客户端**的身份和连接状态。
25	getIccMaxClients	客户端间通信	9.9.2.2	S	C→H	**ECI客户端**读取**ECI主机**可支持的最大**ECI客户端**数量。
26	**getImageTargetId**	高级安全通用	**9.5.2.2.17**	S	C→H	获取CPE的ECI_Image_Target_Id值

表I-2：按字母顺序排列的所有API消息列表

编号	消息	API	条款	类型	方向	描述
27	getPwrStatus	电源管理	9.4.7.2.2	S	C→H	获取当前值电源状态。
28	getTime	时钟	9.4.6.3.2	S	C→H	以整数值读取本地系统时钟。
29	reqAsAStartDecryptSession	高级安全解密	9.5.2.3.2	A	C→H	在**ECI客户端**的**AS时隙**中启动一个解密会话。
30	reqAsAuthDecrSlotConfig	高级安全解密	9.5.2.3.4	A	H→C	使用认证机制（解密模式）认证时隙配置。
31	reqAsAuthEncrSlotConfig	高级安全加密	9.5.2.5.5	A	C→H	使用认证机制（加密模式）认证时隙配置和加密参数。
32	reqAsClientChalResp	高级安全通用	9.5.2.2.7	A	C→H	在数据上应用**ECI客户端**认证密钥并返回结果。
33	reqAsComputeAkClient	高级安全通用	9.5.2.2.6	A	C→H	计算**ECI客户端**应用程序的身份认证密钥。
34	reqAsComputeEncrCw	高级安全加密	9.5.2.5.4	A	C→H	计算加密控制字。
35	reqAsEventCpChange	高级安全加密	9.5.2.5.8	A	H→C	加密会话中关于导入内容中内容属性变化的事件消息。
36	reqAsEventSC	高级安全通用	9.5.2.2.15	A	H→C	会话中关于加扰控制字段变化的事件消息。
37	reqAsEventSessionLimit	高级安全通用	9.5.2.2.12	A	H→C	达到剩余单位的限制值时，将事件发送给**ECI客户端**。
38	reqAsExportConnEnd	高级安全出口	9.5.2.4.3	A	C→H	终止现有的出口会话。
39	reqAsExportConnSetup	高级安全出口	9.5.2.4.2	A	C→H	设置从解密到加密会话的出口连接。
40	reqAsInitSlot	高级安全通用	9.5.2.2.2	A	C→H	初始化**AS时隙**。
41	reqAsLdUssk	高级安全加密	9.5.2.5.6	A	C→H	加载**微服务器**密钥。
42	reqAsLoadSlotLk	高级安全通用	9.5.2.2.5	A	C→H	计算顶级链接密钥（LK1）。
43	reqAsMInikLk1	高级安全加密	9.5.2.5.7	A	C→H	计算不对称的**微客户端**初始化消息。
44	reqAsStartEncryptSession	高级安全加密	9.5.2.5.3	A	C→H	启动一个加密会话。
45	reqAsStopSession	高级安全通用	9.5.2.2.4	A	C→H	停止一个会话。
46	reqCardCmdRes	**智能卡**	9.5.3.6.2	A	C→H	发送卡命令，获取卡回应。
47	reqCardReInit	**智能卡**	9.5.3.6.3	A	C→H	用最新的初始化首选项设置，重新启动卡（热启动或冷启动），并重新执行。
48	reqCCardConClose	**智能卡**	9.5.3.5.6	A	H→C	通知**ECI客户端**卡会话已关闭。

表I-2：按字母顺序排列的所有API消息列表

编号	消息	API	条款	类型	方向	描述
49	reqCCardConOpen	**智能卡**	9.5.3.5.5	A	H→C	通知**ECI客户端**卡会话已打开。
50	reqCCountry	国家	9.4.8.2.2	A	H→C	**ECI主机**请求实际的**ECI客户端**首选国家/地区设置。
51	reqCLanguage	语言	9.4.8.2.4	A	H→C	**ECI主机**请求实际的**ECI客户端**首选语言设置。
52	reqCpChange	内容属性	9.8.2.9.2	A	H→C	**ECI主机**指示需要重加密之内容的内容属性即将发生变化。
53	reqDCAcqModule	数据轮播获取	9.5.4.3	A	C→H	**ECI客户端**请求**ECI主机**使用模块过滤器参数和各种各样模式将特定的ECI数据轮播模块获取到一个文件中。
54	reqDCAcqGroupInfo	数据轮播获取	9.5.4.2	A	C→H	**ECI客户端**请求**ECI主机**读取指定ECI数据轮播之DSI消息中的GroupInfoIndication结构。
55	reqDcrFileQuit	解密媒质文件	9.6.2.4.4.4	A	C→H	**ECI客户端**取消与**ECI主机**的解扰会话。
56	reqDcrFileData	通过文件过滤器请求数据	9.6.2.4.5.2.4	A	C→H	**ECI客户端**请求**ECI主机**通过文件过滤器来获取数据。
57	reqDcrFileStop	解密媒质文件	9.6.2.4.4.3	A	H→C	**ECI主机**请求**ECI客户端**停止解扰**媒质句柄**。
58	reqDcrFileFilter	请求文件过滤器	9.6.2.4.5.2.3	A	C→H	**ECI客户端**请求**ECI主机**为安全数据获取设置一个数据过滤器。
59	reqDcrFileKeyComp	请求密钥计算	9.6.2.4.6.3	A	H→C	启动**ECI客户端**任何要求的计算或其他活动，以创建一个带有Key-ID的控制字。
60	reqDcrFileStart	解密媒质文件	9.6.2.4.4.2	A	H→C	请求**ECI客户端**解扰或返回文件或流的解扰状态。
61	reqDcrIpServer	重加密	9.7.2.6.5	A	C→H	**微客户端**请求**ECI主机**提供**微服务器**的IP地址，以便做与**媒质句柄**会话相关的进一步通信。
62	reqDcrMhBcAlloc	媒质句柄解密	9.6.2.2.5	A	C→H	**ECI客户端**为其自己的广播网络访问请求**媒质句柄**会话。
63	reqDcrMhCancel	媒质句柄解密	9.6.2.2.6	A	C→H	**ECI客户端**取消与**ECI主机**的媒质会话。
64	reqDcrMhClose	媒质句柄解密	9.6.2.2.4	A	H→C	**ECI主机**使用**ECI客户端**关闭媒质会话。
65	reqDcrMhOpen	媒质句柄解密	9.6.2.2.3	A	H→C	**ECI主机**请求**ECI客户端**使用一个**媒质句柄**来打开指定类型的媒质会话。
66	reqDcrMsgRecv	重加密	9.7.2.6.7	A	H→C	**ECI主机**向**微客户端**提供来自**媒质句柄**会话之**微服务器**的消息。
67	reqDcrMsgSend	重加密	9.7.2.6.6	A	C→H	**微客户端**请求**ECI主机**向**媒质句柄**会话之**微服务器**发送一条消息。
68	reqDcrTargetCred	重加密	9.7.2.6.4	A	H→C	**ECI主机**请求**ECI客户端**为通常用于目标认证的**微服务器**连接提供初始化数据。

表I-2：按字母顺序排列的所有API消息列表

编号	消息	API	条款	类型	方向	描述
69	reqDcrTargets	重加密	9.7.2.6.3	A	H→C	**ECI主机**请求**微客户端**提供它可以提供解密服务的加密目标。
70	reqDcrTsData	重加密	9.7.2.6.8	A	C→H	**微服务器**向**ECI主机**提供数据，以便将其转发给**媒质句柄**的**目标微客户端**，以进行解密，包括与ECM相关的同步信息。
71	reqDcrTsDescrquit	TS内容解密	9.6.2.3.4.4	A	C→H	**ECI客户端**请求**ECI主机**终止**媒质句柄会话**的解扰。
72	reqDcrTsData	**微客户端**解密	6.7.2.6.7	A	H→C	**ECI主机**为**微客户端**提供（不远的）将来需要的数据，以便解密**媒质句柄**上的内容。
73	reqDcrTsDescrStop	TS内容解密	9.6.2.3.4.3	A	H→C	**ECI主机**请求**ECI客户端**停止**媒质句柄会话**的解扰。
75	reqDcrTsDescrStart	TS内容解密	9.6.2.3.4.2	A	H→C	请求**ECI客户端**对TS中程序的解扰状态进行解扰或返回。
76	reqDcrTsRelocate	解密TS源控制	9.6.2.3.6.3	A	C→H	**ECI客户端**重新安置TS的源。
77	reqDcrTsSection	解密TS数据获取	9.6.2.3.5.5	A	H→C	将获取的部分转交给**ECI客户端**。
78	reqDcrTsSelectCancel	解密TS源控制	9.6.2.3.6.6	A	C→H	**ECI客户端**取消其之前的程序选择。
79	reqDcrTsSelectPmt	解密TS源控制	9.6.2.3.6.5	A	C→H	**ECI客户端**通过PMT在TS中选择节目。
80	reqDcrTsSelectPrg	解密TS源控制	9.6.2.3.6.4	A	C→H	**ECI客户端**通过程序编号在TS中选择程序。
81	reqDcrTsTable	解密TS数据获取	9.6.2.3.5.6	A	C→H	**ECI客户端**获取流中的表格。
82	reqEncrConnDrop	重加密	9.7.2.5.5	A	H→C	**ECI主机**请求**ECI客户端**丢弃关于之前预认证之重加密连接的任何信息。
83	reqEncrConnSetup	重加密	9.7.2.5.4	A	H→C	**ECI主机**请求**ECI客户端**创建重加密目标连接，并预认证重加密**目标**，以便在设置**媒质句柄**会话中做后续参考。
84	reqEncrFileData	重加密	9.7.2.5.18	A	C→H	**微服务器**为**ECI主机**提供一条消息，以便将其转发给**媒质句柄**的目标**微客户端**进行解密，包括KeyID相关的同步信息。
85	reqEncrIpServer	重加密	9.7.2.5.13	A	H→C	**ECI主机**请求**微服务器**的IP服务器地址，以便允许**微客户端**创建IP连接。

表I-2：按字母顺序排列的所有API消息列表

编号	消息	API	条款	类型	方向	描述
86	reqEncrMhCancel	重加密	9.7.2.5.9	A	C→H	**ECI客户端**终止与指定的出口**ECI客户端**的**入口连接**。
87	reqEncrMhClose	重加密	9.7.2.5.8	A	H→C	**ECI主机**关闭与**ECI客户端**的重**加密会话**。
88	reqEncrMhOpen	重加密	9.7.2.5.7	A	H→C	**ECI主机**请求**ECI客户端**打开**媒质句柄**会话，以便为已建立的重加密连接对来自**入口连接**的内容进行重加密。
89	reqEncrMhQuit	重加密	9.7.2.5.12	A	C→H	**ECI客户端**通知**ECI主机媒质句柄**重加密操作已终止。
90	reqEncrMhStart	重加密	9.7.2.5.10	A	H→C	**ECI主机**请求**ECI客户端**为**媒质句柄**会话启动重加密操作。
91	reqEncrMhStop	重加密	9.7.2.5.11	A	H→C	**ECI主机**请求**ECI客户端**为**媒质句柄**会话停止重加密操作。
92	reqEncrMsgRecv	重加密	9.7.2.5.18	A	H→C	**ECI主机**向**微服务器**提供一条来自某个**媒质句柄**会话**目标**的消息。
93	reqEncrMsgSend	重加密	9.7.2.5.14	A	C→H	**微服务器**请求**ECI主机**将消息转发给**媒质句柄**会话的**目标**。
94	reqEncrTargets	重加密	9.7.2.5.3	A	H→C	**ECI主机**请求**ECI客户端**提供它可以认证的**目标**节点。
95	reqEncrTsData	重加密	9.7.2.5.16	A	C→H	**微服务器**向**ECI主机**提供数据，以便将其转发给**媒质句柄**的**目标微客户端**进行解密，包括与ECM相关的同步信息。
96	reqEncrTsEcm	重加密	9.7.2.5.17	A	C→H	**微服务器**发出ECM部分，**微客户端**需要在下一个加密期间进行解密。
97	reqExpConnCancel	出口连接	9.7.2.3.5	A	C→H	**ECI客户端**通过入口**ECI客户端**终止初始化的**出口连接**。
98	reqExpConnDrop	出口连接	9.7.2.3.4	A	H→C	**ECI主机**取消先前已初始化的、出口**ECI客户端**到入口**ECI客户端**的连接。
99	reqExpConnNodes	出口连接	9.7.2.3.2	A	H→C	**ECI主机**从**ECI客户端**请求出口选项节点。
100	reqExpConnSetup	出口连接	9.7.2.3.3	A	H→C	**ECI主机**请求**ECI客户端**根据**入口链**将**出口连接**初始化为入口**ECI客户端**。
101	reqExpMhCancel	出口连接	9.7.2.3.8	A	C→H	**ECI客户端**取消出口会话。
102	reqExpMhClose	出口连接	9.7.2.3.7	A	H→C	**ECI主机**关闭出口会话。
103	reqExpMhOpen	出口连接	9.7.2.3.6	A	H→C	**ECI主机**请求**ECI客户端**根据先前已初始化的**出口连接**创建一个出口会话。

表I-2：按字母顺序排列的所有API消息列表

编号	消息	API	条款	类型	方向	描述
104	reqFileClose	文件系统	9.4.5.2.3	A	C→H	关闭一个打开的文件。
105	reqFileCreate	文件系统	9.4.5.4.3	A	C→H	创建一个新的文件。
106	reqFileDelete	文件系统	9.4.5.4.4	A	C→H	删除一个文件。
107	reqFileDir	文件系统	9.4.5.4.5	A	C→H	列出**ECI客户端**文件系统中可用文件的文件名。
108	reqFileOpen	文件系统	9.4.5.2.2	A	C→H	打开一个**ECI客户端**私人文件。
109	reqFileRead	文件系统	9.4.5.3.3	A	C→H	从当前文件位置开始读取连续的字节。
110	reqFileRemoveData	文件系统	9.4.5.3.5	A	C→H	从当前位置的文件中删除数据。
111	reqFileSeek	文件系统	9.4.5.3.4	A	C→H	重新定位当前文件位置。
112	reqFileStat	文件系统	9.4.5.4.2	A	C→H	返回文件的大小和修改时间。
113	reqFileWrite	文件系统	9.4.5.3.2	A	C→H	从当前文件位置开始连续写入字节。
114	reqHCardConClose	**智能卡**	9.5.3.5.7	A	C→H	通知**ECI主机ECI客户端**希望终止与所连接卡的会话。
115	reqHCountry	国家	9.4.8.2.1	A	C→H	请求实际的**ECI主机**首选的国家/地区设置。
116	reqHLanguage	语言	9.4.8.2.3	A	C→H	请求实际的**ECI主机**首选的语言设置。
117	reqHttpGetData	HTTP获取	9.4.4.6.3	A	C→H	对URL执行HTTP获取请求，并将结果作为数据传递给客户端。
118	reqHttpGetFile	HTTP获取	9.4.4.6.3	A	C→H	对URL执行HTTP获取请求，并将结果存储在文件中。
119	reqIccPipeCancel	客户端间通信	9.9.2.7	A	C→H	**ECI客户端**取消管道。
120	reqIccPipeClose	客户端间通信	9.9.2.8	A	H→C	**ECI主机**通知**ECI客户端**与对应方的管道已关闭。
121	reqIccPipeMsgRecv	客户端间通信	9.9.2.10	A	H→C	**ECI客户端**收到来自其管道对应方的消息。
122	reqIccPipeMsgSend	客户端间通信	9.9.2.9	A	C→H	**ECI客户端**向其管道的对应方发送一条消息。
123	reqIccPipeOpen	客户端间通信	9.9.2.5	A	C→H	请求打开一条到另一个**ECI客户端**的管道。
124	reqIccPipeOpenReq	客户端间通信	9.9.2.6	A	H→C	来自另一个**ECI客户端**的打开一条管道的请求。
125	reqIccSystemReady	客户端间通信	9.9.2.3	A	H→C	**ECI主机**通知**ECI客户端**所有的**ECI客户端**都已被初始化。
126	reqImpConnCancel	入口连接	9.7.2.4.6	A	C→H	**ECI客户端**终止与指定的出口**ECI客户端**的**入口连接**。
127	reqImpConnChain	入口连接	9.7.2.4.3	A	H→C	**ECI主机**请求入口**ECI客户端**为特定的入口节点提供输入链。
128	reqImpConnChainRenew	入口连接	9.7.2.4.3	A	C→H	**ECI客户端**请求**ECI主机**使用更新的**入口链**来重新初始化连接。

表I-2：按字母顺序排列的所有API消息列表

编号	消息	API	条款	类型	方向	描述
129	reqImpConnDrop	入口连接	9.7.2.4.5	A	H→C	**ECI主机**使用指定的出口**ECI客户端**来丢弃入口连接。
130	reqImpConnNodes	入口连接	9.7.2.4.2	A	H→C	**ECI主机**请求入口**ECI客户端**提供其入口节点。
131	reqImpConnSetup	入口连接	9.7.2.4.4	A	H→C	**ECI主机**请求入口**ECI客户端**通过一个入口节点来初始化与特定出口**ECI客户端**的**入口连接**。
132	reqIpAddrinfo	IP套接字	9.4.4.3.4	A	C→H	获取（远程）**ECI主机**的地址。
133	reqIpClose	IP套接字	9.4.4.3.3	A	C→H	关闭**ECI** IP套接字。
134	reqIpSocket	IP套接字	9.4.4.3.2	A	C→H	打开**ECI** IP套接字。
135	reqIpTcpAccept	TCP/IP套接字	9.4.4.5.5	A	C→H	TCP服务器对等端接受来自TPC客户端对等端的连接
136	reqIpTcpConnect	TCP/IP套接字	9.4.4.5.2	A	C→H	TCP客户端连接到TCP服务器对等端。
137	reqIpTcpRecv	TCP/IP套接字	9.4.4.5.4	A	C→H	从所连接的对等端接收数据。
138	reqIpTcpSend	TCP/IP套接字	9.4.4.5.3	A	C→H	将数据发送给所连接的对等端。
139	reqIpUdpRecvMsg	TCP/IP套接字	9.4.4.4.3	A	C→H	从对等端UDP端口接收消息。
140	reqIpUdpSendMsg	TCP/IP套接字	9.4.4.4.2	A	C→H	将消息发送给对等端UDP端口。
141	reqParAuthChk	内容属性	9.8.2.10.3	A	C→H	请求**ECI主机**代表**ECI客户端**执行父认证。
142	reqParAuthChkCan	内容属性	9.8.2.10.4	A	C→H	取消先前向主机提出的父认证请求。
143	reqParAuthCid	内容属性	9.8.2.10.5	A	H→C	为需要解码的（未来）内容科目请求父pin码授权。这可能会触发一个父身份认证对话。
144	reqParAuthDel	内容属性	9.8.2.11.2	A	H→C	**ECI主机**将父身份认证委托给一个**ECI客户端**。
145	reqParAuthDelCan	内容属性	9.8.2.11.3	A	H→C	**ECI主机**取消先前向**ECI客户端**提出的父认证请求。
146	reqPwrChange	电源管理	9.4.7.2.4	A	H→C	电源状态变化的通知。
147	reqTimerCancel	计时器	9.4.6.2.3	A	C→H	取消先前设置的计时器事件。
148	reqTimerEvent	计时器	9.4.6.2.2	A	C→H	将来设置一个计时器事件。
149	reqUiClientQuery	用户接口	9.4.3.4.8	A	H→C	**ECI客户端**从浏览器中的HTML应用程序接收请求，并提供（动态）响应。
150	reqUiContainerMount	用户接口	9.4.3.4.2	A	C→H	用HTML资源装载UI应用程序容器以支持UI会话。
151	reqUiSessionCancel	用户接口	9.4.3.4.7	A	H→C	**ECI主机**取消**用户**接口会话。
152	reqUiSessionClose	用户接口	9.4.3.4.6	A	C→H	**ECI客户端**结束**用户**接口会话。
153	reqUiSessionCommence	用户接口	9.4.3.4.4	A	H→C	**ECI主机**建议**ECI客户端**打开UI会话。
154	reqUiSessionOpen	用户接口	9.4.3.4.5	A	C→H	**ECI客户端**请求打开与**用户**的**用户**接口会话，并在屏幕上呈现内容。

表I-2：按字母顺序排列的所有API消息列表

编号	消息	API	条款	类型	方向	描述
155	reqPwrWakeupEvent	电源管理	9.4.7.3	A	H→C	提示唤醒计时器到期。
156	setApiVersion	接口发现	9.4.2.4	S	C→H	设置要使用之主机API的版本。
157	setAsPermitCPChange	高级安全加密	9.5.2.4	S	C→H	启用/禁用导入的内容属性CP更改对控制字选择发挥作用，用于在一个加密会话中进行加密。
158	setAsSC	高级安全加密	9.5.2.4	S	C→H	设置加密会话之加密内容的加扰控制字段。
159	setAsSessionLimitEvent	高级安全通用	9.5.2.5.11	S	C→H	设置向**ECI客户端**发送reqAsEventSessionLimit消息的限值。
160	setCardMatch	**智能卡**	9.5.3.5.2	S	C→H	设置**ECI客户端**的卡识别说明符列表。
161	setCpSync	内容属性	9.8.2	S	C→H	**ECI客户端**提示当前的内容属性组是一致的，并可用于即将被控制字解扰的内容。
162	setDcrBasicUri	内容属性	9.8.2.5.1	S	C→H	为需要解扰的内容设置基本的URI。
163	setDcrCustUri	内容属性	9.8.2.4.1	S	C→H	为需要解扰的内容设置自定义URI。
164	setDcrMarkBasic	内容属性	9.8.2.7.5	S	C→H	为需要解扰的内容设置基本的标记有效载荷。
165	setDcrMarkExt	内容属性	9.8.2.7.6	S	C→H	为需要解扰的内容设置扩展的标记有效载荷。
166	setDcrMarkMeta	水印	9.8.2.7.3	S	C→H	设置标记系统控制值。
167	setDcrMhMatch	媒质句柄解密	9.6.2.2.2	S	C→H	向**ECI主机**发送信号，在该主机下可识别用于解扰内容的**ECI客户端**。
168	setDcrModes	重加密	9.7.2.6.1	S	C→H	**微客户端**通知**ECI主机**它支持的模式（加密模式、数据格式模式和同步模式）。
170	setDcrOutputCtl	内容属性	9.8.2.6.1	S	C→H	为需要解扰的内容设置输出控制限制。
171	setDcrParCtl	内容属性	9.8.2.8.1	S	C→H	为需要解扰的内容设置父控制条件。
172	setDcrStdUri	内容属性	9.8.2.8.1	S	C→H	为需要解扰的内容设置标准URI。
173	setDcrTsSectionAcq	解密TS数据获取	9.6.2.3.5.4	S	C→H	设置区段采集的过滤器。
176	setDcrTsSectionAcqDefault	解密TS数据获取	9.6.2.3.5.3	S	C→H	设置区段采集的默认过滤器。
177	setEncrModes	重加密	9.7.2.5.2	S	C→H	**微服务器**通知**ECI主机**它支持的模式（加密模式、数据格式模式和同步模式）。
178	setPwrInfo	电源管理	9.4.7.2.3	S	C→H	请求事件通知以更改电源状态。

表I-2：按字母顺序排列的所有API消息列表

编号	消息	API	条款	类型	方向	描述
179	setUiClientAttention	用户接口	9.4.3.4.3	S	C→H	**ECI客户端**表示希望启动一个UI会话，而无关某个**媒质句柄**。
180	setPwrWakeup	电源管理	9.4.7.3	S	C→H	设置**ECI客户端**的唤醒时间。

附录II

有待进一步发展的领域

（本附录非本建议书不可分割的组成部分。）

已经确定，本建议书需要做进一步的开发和验证，才能满足[ITU-T J.1010]中规定的要求，并且[ITU-T J.1010]需要进行更新，以反映MovieLabs增强型内容保护（ECP）规范[b-ECP]的要求。[ITU-T J.1011]、ITU-T J.1012、[ITU-T J.1013]、[ITU-T J.1014]、[ITU-T J.1015]和 [b-ITU-T J.1015.1]建议书未来应予更新，以反映对[ITU-T J.1010]的那些更新。

国际电联的许多成员国以及各行各业的利益攸关方 – 包括设备和电子组件的制造商、受版权保护的内容的所有者和被许可者、过顶（OTT）业务和线性电视业务的提供商以及全球各地基于有条件访问系统（CAS）和数字版权管理（DRM）解决方案的提供商都对嵌入式通用接口（ECI）不能完全满足ECP要求以及更广泛的行业内容保护要求表示了担忧。

更具体地说，其对ITU-T第9研究组（SG9）会议（2020年4月16-23日）的文稿引起了人们的关注。来自以色列、澳大利亚、ITU-T部门成员三星公司以及SG9准成员Sky Group和MovieLabs的文稿提议在ECI建议书中纳入一系列更改，但未就此达成共识。这些项目在[b-SG9 Report 17 Ann.1]中进行了清点。

这些建议包括：

1) 通过缩小ECI范围来简化ECI系统；
2) 取消DRM；
3) 取消对内容的重加密；
4) 取消软件管理；
5) 增加用于安全存储和加密操作的API；
6) 允许供应商特定的密钥阶梯；
7) 采用ITU-T J.1207 TEE要求；
8) 包括VM的TEE实施方案；
9) 升级密码算法的强度，例如，使用SHA-384；
10) 使用标准证书，例如，ITU-T X.509；
11) 重新考虑客户端之间的通信；
12) 与ETSI进行其他联络；
13) 进行额外的同行评审；
14) 探索信托机构模型的替代方案；
15) 进一步定义ECI合规性和稳健性规则的技术方面问题；
16) 增加有关多样性的要求，例如，地址空间随机化；
17) 增加有关运行时完整性检查的要求。

这些建议反映出内容保护及其违背之可能带来的威胁正在不断演变。ECI最初是在批准本ITU-T建议书之前近十年来构思的。对像ECI这样的系统，需要定期根据攻击技术和行业保护要求的最新状况进行评估。

存在其他机制以实现互操作性。特别是对DRM用例，大多数互联网视频服务已经部署了其他解决方案，以提供互操作性并满足其需求。

由于许多成员国将国际电联标准视为对其市场和行业发展有影响力的指导来源，因此进一步的澄清很重要。关注清单确保ECI在其国内市场中的实施，这可涉及对本ITU T建议书含义的全面理解，并确保在考虑立法、法规或有关要求消费者数字电视设备可互操作的市场需求时能虑及这些问题。它还确保技术设备制造商（它们可能更喜欢使用一组独特的要求或其他标准来设计产品）在为不同市场开发产品时可以考虑到这些问题。

参考书目

[b-ITU-T J.1015.1] ITU-T J.1015.1建议书（2020年），用于可交换有条件访问/数字版权管理（CA/DRM）解决方案的嵌入式通用接口（ECI）；高级安全系统-密钥阶梯块：控制字使用规则信息和相关数据1的验证。

[b-ITU-T J-Suppl.7] ITU-T 系列J建议书 – 增补7（2020年），用于可交换有条件访问/数字版权管理（CA/DRM）解决方案的嵌入式通用接口（ECI）；ECI实施导则（EG）。

[b-SG9 Report 17 Ann.1] ITU-T SG9会议报告，SG9-R17-附件1（2020年），2020年4月16-23日召开的SG9全虚拟会议第17号报告附件1。
https://www.itu.int/md/T17-SG09-R-0017/en

[b-ETSI GS ECI 001-1] ETSI GS ECI 001-1: "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 1: Architecture, Definitions and Overview".

[b-ETSI GS ECI 001-2] ETSI GS ECI 001-2: "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 2: Use cases and requirements".

[b-ETSI GS ECI 001-3] ETSI GS ECI 001-3 V1.1.1 (2017-07): "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 3: CA/DRM Container, Loader, Interfaces, Revocation"

[b-ETSI GS ECI 001-4] ETSI GS ECI 001-4: "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 4: The Virtual Machine".

[b-ETSI GS ECI 001-5-1] ETSI GS ECI 001-5-1: "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions Part 5: The Advanced Security System Sub-part 1: ECI specific functionalities".

[b-ETSI GS ECI 001-5-2] ETSI GS ECI 001-5-2: "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 5: The Advanced Security System; Sub-part 2: Key Ladder Block".

[b-ETSI TS 102 034] ETSI TS 102 034 (V1.4.1): "Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks".

[b-Richardson] Richardson, S. Ruby: "RESTfull Web services", L. o'Reilly, 2007.

[b-DASH-IF V3] Dash Industry Forum (2015): "Guidelines for Implementation: Dash-IF Interoperability Points version 3.0".

[b-DASH-IF ID] Dash Industry Forum: "Identifiers for protection".
http://dashif.org/identifiers/protection/.

[b-CA Browser] CA Browser Forum: "Baseline Requirements: Certificate Policy for the Issuance and Management of Publicly-Trusted Certificates".
https://cabforum.org/.

[b-NIST SP 800-52r2] NIST SP 800-52 rev2 (August 2019): "Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations".

[b-CI Plus] CI Plus Specification V1.3.1 (2011-09).
Available at http://www.ci-plus.com.

[b-DLNA] DLNA Networked Device Interoperability Guidelines, Digital Living Network Alliance.
http://www.dlna.org/guidelines

[b-HbbTV] Hybrid Broadcast Broadband Television (HbbTV®) Operator Applications.

[b-ETSI GS ECI 001-6] ETSI GS ECI 001-6: "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 6: Trust Environment".

[b-ETSI GS ECI 002] ETSI GS ECI 002: "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; System validation".

[b-IETF RFC 8259] IETF RFC 8259 (2017), *The JavaScript Object Notation (JSON) Data Interchange Format*.

[b-IANA] IANA "Media Types" database.
http://www.iana.org/assignments/media-types/media-types.xhtml

[b-HDCP2.3] Digital Content Protection LLC, "*High Bandwidth Digital Content Protection System, Mapping HDCP to HDMI*" revision 2.3., Feb 28, 2018
https://www.digital-cp.com/sites/default/files/HDCP%20on%20HDMI%20Specification%20Rev2_3.pdf

[b-Illgner] Klaus Illgner, Christoph Schaaf, MarnixVlot: "Embedded Common Interface (ECI) for Digital Broadcasting Applications: Security and Interoperability combined", Broadband Journal of the SCTE, Vol. 38, No. 3, August 2016.

[b-Menezes] Menezes, A., van Oorschot, P. and Vanstone, S: "Handbook of Applied Cryptography", CRC Press, 1996.

[b-ECP] MovieLabs Specification for Enhanced Content Protection – Version 1.2
https://movielabs.com/ngvideo/MovieLabs_ECP_Spec_v1.2.pdf

虽然本条款中包含的任何超链接在发布时均有效，但不能保证其长期有效性。

ITU-T建议书系列

系列A	ITU-T工作的组织
系列D	资费及结算原则和国际电信/ICT的经济和政策问题
系列E	综合网络运行、电话业务、业务运行和人为因素
系列F	非话电信业务
系列G	传输系统和媒介、数字系统和网络
系列H	视听及多媒体系统
系列I	综合业务数字网
系列J	**有线网络和电视、声音节目及其他多媒体信号的传输**
系列K	干扰的防护
系列L	环境与ICT、气候变化、电子废物、节能；线缆和外部设备的其他组件的建设、安装和保护
系列M	电信管理，包括TMN和网络维护
系列N	维护：国际声音节目和电视传输电路
系列O	测量设备的技术规范
系列P	电话传输质量、电话设施及本地线路网络
系列Q	交换和信令
系列R	电报传输
系列S	电报业务终端设备
系列T	远程信息处理业务的终端设备
系列U	电报交换
系列V	电话网上的数据通信
系列X	数据网、开放系统通信和安全性
系列Y	全球信息基础设施、互联网协议问题、下一代网络、物联网和智慧城市
系列Z	用于电信系统的语言和一般软件问题

瑞士印刷
2020年，日内瓦