ICS 号：xxxx-xxxx
中国标准文献分类号

世 界 超 高 清 视 频 产 业 联 盟 标 准

T/UWA xxxx-xxxx

基于流媒体的多屏互动技术规范

Technical specification of multiscreen collaboration based on streaming media

（ 征求意见稿 ）

xxxx-xx-xx 发布（Calibri/黑体四号） xxxx-xx-xx 实施

世界超高清视频产业联盟

目　　次

前　言

本文件按照 GB/T 1.1—2020《标准化工作导则 第 1 部分：标准化文件的结构和起草规则》给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本文件由世界超高清视频产业联盟提出并归口。

本文件主要起草单位：。

本文件主要起草人：。

基于流媒体的多屏互动技术要求

1 范围

本规范旨在定义同一局域网环境下，流媒体资源多屏互动场景中多设备之间的相互协同机制、协议和接口规范。本规范中，基于流媒体的多屏互动场景仅包含针对选定媒体资源进行跨设备内容传输和播控。

本规范适用于同一局域网环境中跨设备多屏互动相关的应用软件设计和开发。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

rfc6762 Multicast DNS (mDNS)
<https://www.rfc-editor.org/info/rfc6762>
rfc6763 DNS-Based Service Discovery
<https://www.rfc-editor.org/info/rfc6763>
rfc2326 Real Time Streaming Protocol
<https://www.rfc-editor.org/info/rfc2326>
rfc8216 HTTP Live Streaming
<https://www.rfc-editor.org/info/rfc8216>
rfc5869 HMAC-based Extract-and-Expand Key Derivation Function(HKDF)
<https://www.rfc-editor.org/info/rfc5869>
rfc2104 HMAC: Keyed-Hashing for Message Authentication
<https://www.rfc-editor.org/info/rfc2104>

3 术语和定义

下列术语以及定义适用于本文件。

3.1 内容 Content

要投播的一系列的媒体类资源，如视频、音乐、图片等。

3.2 内容服务端 Content Storage and Management Server

投播内容的存储和业务管理系统的服务器。

3.3 发送端 Source

多屏互动的发起设备，具备将当前设备中的资源通过本规范投射到另一个设备中的能力，包括发起设备发现，发起设备鉴权，发起媒体资源播控等。

3.4 接收端 Sink

多屏互动的接收设备，具备包括将接收发送端通过本规范投射过来的资源，并对相应资源进行流媒体拉取和播放的能力，包括允许被发现，接收设备鉴权，接收媒体资源播控，媒体资源渲染等。

3.5 渲染 Rendering

视音频媒体资源内容播放、图片资源显示等行为。

3.6 播控 Play and Control

发送端通过协议控制接收端进行内容获取、渲染及播控，并同步接收端播放状态。

3.7 可信设备 Trusted Device

通过可信设备认证的发送端和接收端设备

3.8 加密 Encrypt

为了产生加密的信息，即隐藏数据的信息内容，由密码算法对数据进行（可逆）变换。

3.9 解密 Decrypt

与加密过程相对应的逆过程。即由密码算法对已加密数据进行逆变换。

3.10 可信通道 Trusted Channel

通过可信设备认证协议产生密钥，能够对数据加密、发送、接收、解密处理的数据传输通道。

3.11 DRM 服务端 DRM server

向DRM客户端提供许可证服务的实体。

4 缩略语

下列缩略语适用于本文件。

API　应用程序接口（Application Programming Interface）
ChinaDRM 中国数字版权管理 （China Digital Rights Management）
DRM　数字版权管理（Digital Rights Management）
ECDH　椭圆曲线DH密钥协商协议（Elliptic Curve Diffie-Hellman key agreement）
HTTP　超文本传输协议（Hyper Text Transfer Protocol）
HLS　基于HTTP的实时流媒体协议（Http Live Streaming）
LAN　局域网（Local Area Network）
mDNS　多播域名系统（Multicast DNS）
ISO　国际标准化组织（International Organization for Standardization）
JSON　JS对象简谱（JavaScript Object Notation）
RTSP　实时流协议（Real Time Streaming Protocol）
SPEKE　简单秘钥协商（Simple Password Exponential Key Exchange）
URL　统一资源定位符（Uniform Resource Locator）
WLAN　无线局域网络（Wireless Local Area Network）
NearLink 中国原生的新一代近距离无线联接技术

5 总体框架

5.1 概述

随着电子产品形态的繁荣发展，各类产品通过发挥自身优势满足消费者的不同需求，智能手机、笔记本等类设备方便携带，易操作，内容丰富；智能电视、音箱等类设备更专注于视频、音乐、图片等多媒体类内容渲染显示，超大屏幕给消费者带来沉浸式享受，共同观影等。

消费者在家居、办公等场景时需要把智能手机上的内容流转到智能电视进行播放，实现多设备协同，充分发挥环境中设备优势，获取最佳体验。

本规范定义了多屏互动软件的基础架构及交互操作规范，涵盖设备发现和控制、连接认证鉴权、能力协商、媒体格式、媒体播控协议、网络传输及用户界面规格，实现发送端设备媒体内容协同到接收端设备播控的功能。

5.2 协议架构

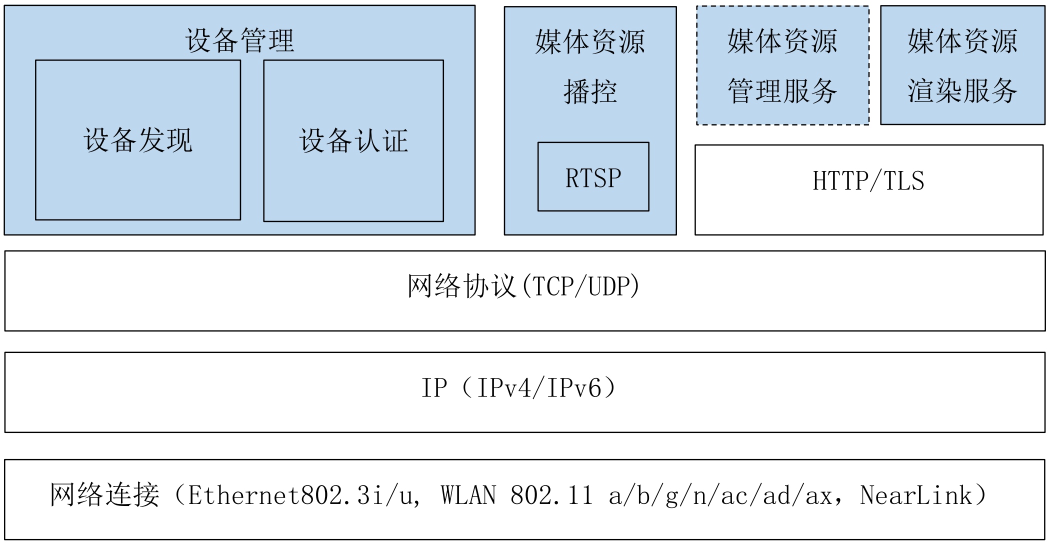


图 1 基于流媒体的多屏互动基础架构

如图 1 基础架构所示，本规范主要定义应用层协议，主要包含四个部分：设备管理、媒体资源播控、媒体资源管理和渲染。

设备管理主要负责通过设备发现协议和设备可信认证协议完成双端设备发现和可信认证管理。设备发现通过设备发现协议进行服务发布和订阅完成设备能力交互；设备认证基于设备可信算法进行双端交互完成可信认证，生成秘钥，建立可信通道。

媒体资源播控服务（控制模块）负责将内容从发送端发送到接收端进行播控。发送端将从媒体资源管理服务获取的内容链接发送到接收端；接收端负责通过该内容链接访问资源服务器，通过 HTTP 协议申请资源，接收内容进行播放；接收端播控服务把播放进度、状态等信息同步给发送端，发送端可以通过播控协议对接收端播放进行控制，如暂停/继续、倍速等。媒体资源播控基于实时流媒体协议 RTSP 实现。

媒体资源管理（媒体服务器）负责提供视音频等资源存储和认证授权管理，包括业务系统和 DRM 业务管理系统；业务系统完成内容管理、业务鉴权、内容分发等功能；针对需要 DRM 保护的内容，DRM 业务管理系统提供内容加密、密钥管理及内容授权功能。本规范不详细规定媒体资源管理的具体行为和协议内容，而由业务实现确定。

媒体资源渲染（渲染模块）负责内容获取、渲染和播放，通过 HTTP TLS1.2/1.3 协议从媒体资源管理存储服务器获取要播放的资源，直播场景下采用 RTP 协议从媒体资源管理存储服务器获取资源，根据

对应的资源类型加载对应的渲染器，进行渲染，主要包括音视频播放器、图片查看器。

5.3　**模型定义**

本要求定义发送端和接收端基础模型，一个物理设备可承载一个或多个发送端或接收端实例。

5.4　应用场景

5.4.1　一对一本地资源协同

该场景如下图所示：

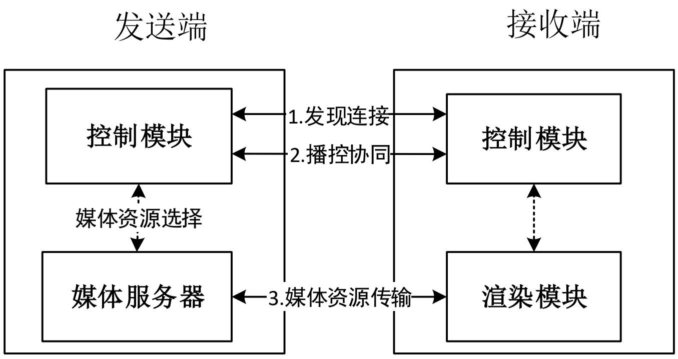


图 2　一对一本地资源协同

该场景仅涉及两个设备，发送端和接收端。发送端部署控制模块和媒体服务器，接收端部署控制模块和渲染模块。发送端控制模块与媒体服务器之间的资源选择过程本规范不定义，如可以由用户选择本地某个资源文件等方式实现。两端的控制模块之间首先实现发现连接，然后进行播控。接收端的控制模块将控制信息传递给渲染模块，两模块间的信息传递属于内部实现，本协议不定义。最后，渲染模块从媒体服务器中获取指定的媒体资源内容。

5.4.2　一对一在线资源协同

该场景如下图所示：

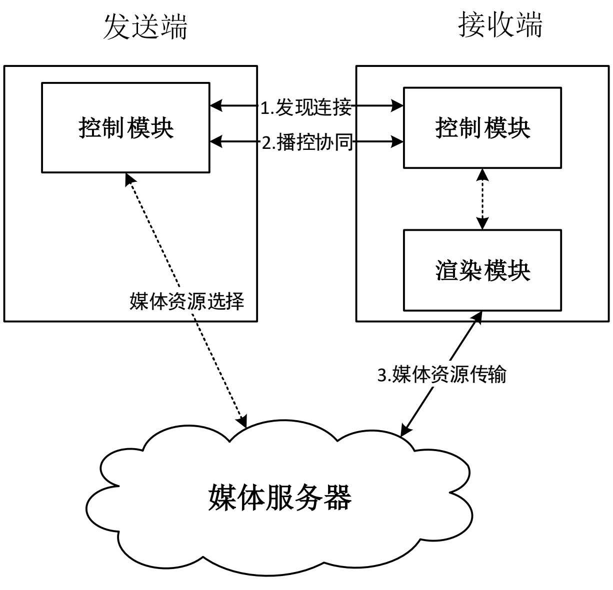


图 3　一对一在线资源协同

该场景仅涉及三个设备，发送端、接收端和媒体服务器。发送端仅部署控制模块，接收端部署控制

模块和渲染模块，媒体服务器为单独另外设备。发送端控制模块与媒体服务器之间的资源选择过程本规范不定义，如可以由发送端内其他模块如视频应用与媒体服务器交互后通过应用接口传递给控制器。两端的控制模块之间首先实现发现连接认证，然后进行播控协同。接收端的控制模块将控制信息传递给渲染模块，两模块间的信息传递属于内部实现，本协议不定义。最后，渲染模块从在线的媒体服务器中获取制定的媒体资源内容。

5.4.3　一对多本地资源协同

该场景如下图所示：

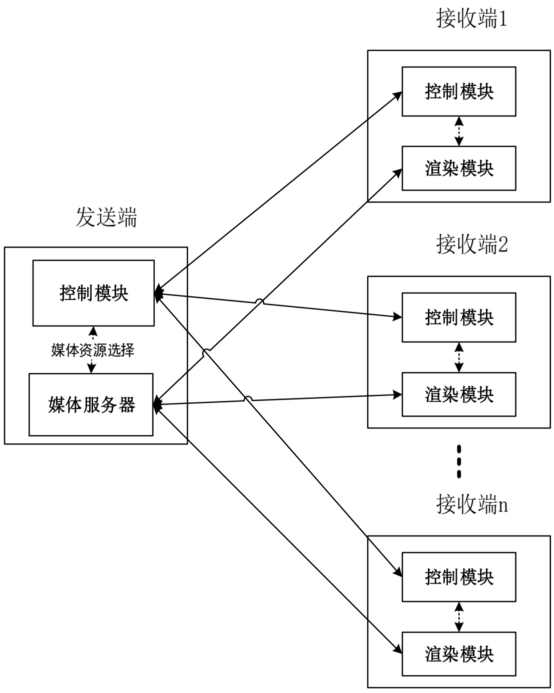


图 4　一对多本地资源协同

该场景涉及一个发送端和多个接收端。发送端仅部署控制模块和媒体服务器，每个接收端部署控制模块和渲染模块。发送端控制模块与媒体服务器之间的资源选择过程本规范不定义。发送端与每个接收端之间的协同过程与 5.4.1 章节定义一致。发送端的控制模块和媒体服务器需具备启动多个连接的能力。

5.4.4　一对多在线资源协同

该场景如下图所示：

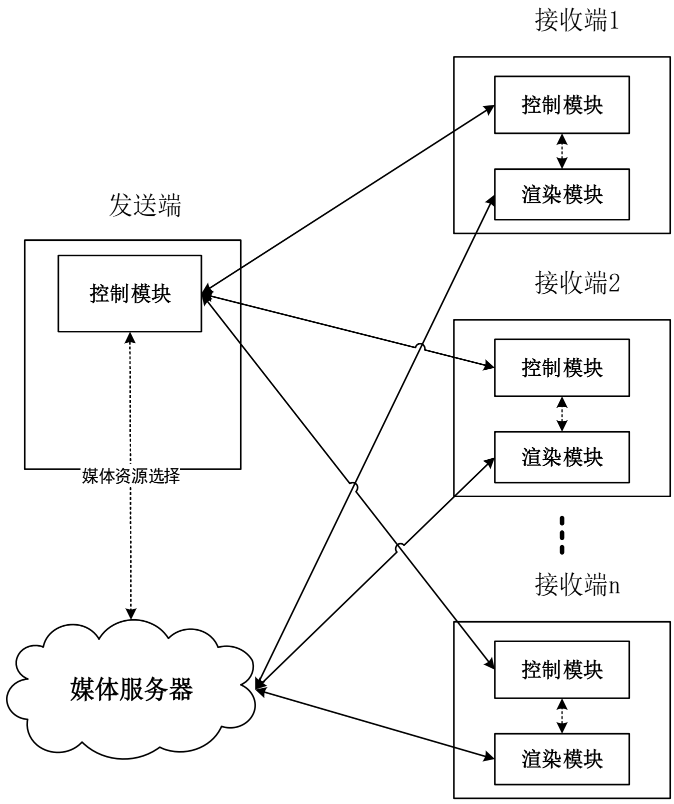


图 5　一对多在线资源协同

该场景包含一个发送端、一个媒体服务器和多个接收端。发送端仅部署控制模块，每个接收端部署一个控制模块和渲染模块，媒体服务器为单独另外设备。控制模块与媒体服务器之间的资源选择过程本规范不定义。发送端、媒体服务器与每个接收端之间的协同过程与 5.4.2 章节定义一致。控制模块和媒体服务器需具备启动多个连接的能力。

5.5　业务流程

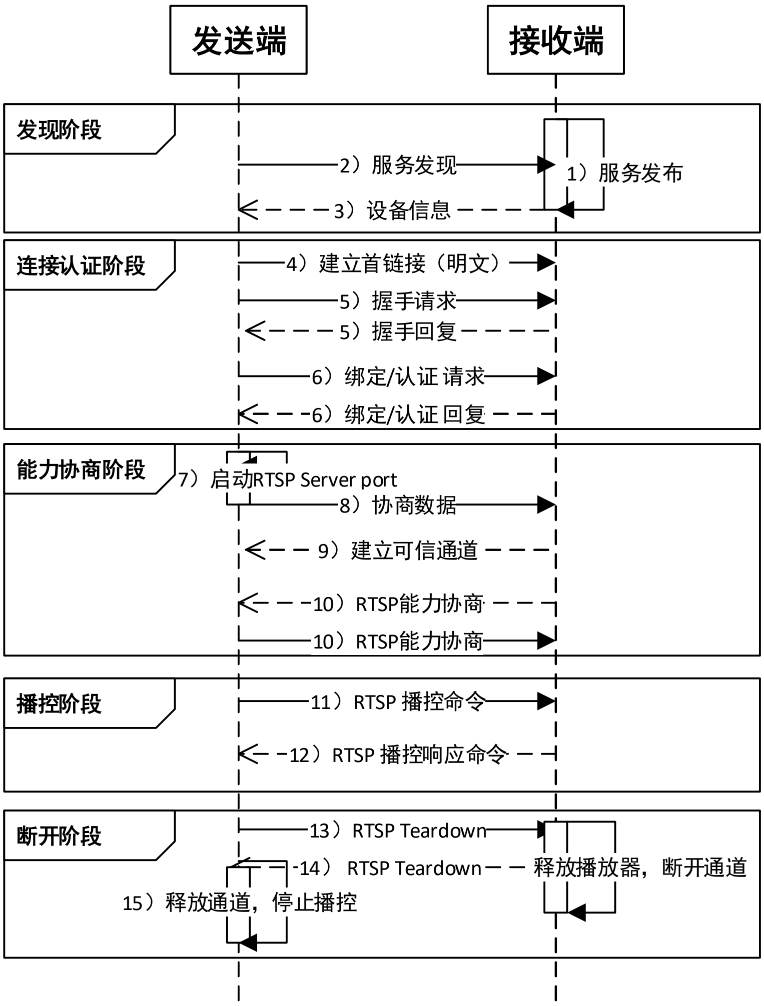


图 6 业务流程

流媒体的多屏协同主要包含发现阶段、连接认证阶段、能力协商阶段、播控阶段及断开阶段，各阶段依次流动。各阶段流程概括介绍如下：

发现阶段：

1） 接收端由用户打开或默认设备开启后，进行服务发布，等待被发现；

2） 发送端进行服务发现；

3） 接收端设备回复本端支持的服务、能力以及服务端口等信息；

连接认证阶段：

4）发送端设备发起连接，通过发现阶段发现的接收端设备 IP 和服务端口建立首链接通路；

5）发送端发送握手报文，报文主要包括本机 ID，鉴权模式等信息，接收端接收到握手报文后结合本端能力进行回复；

6）双端如有任一端不存在信任关系是发起设备绑定流程，如双端均存有对端设备的可信凭证可进入认证流程，完成后双端均生成会话秘钥，具体协商可参看 6.2 章节；

能力协商阶段：

7）发送端建立协商可信通道的服务端，生成 RTSP 服务端口，等待连接；

8）发送服务端口通过会话秘钥加密后发送到接收端；

9）接收端连接到 RTSP 端口，建立可信通道，通道数据均使用会话加解密；

10）双端通过 RTSP 进行进行播放能力协商并完成对应响应，详细参考 7.2 章节；

播控阶段：

11）发送端通过播控可信通道进行播放内容播控命令发送；

12）接收端进行播放命令响应及播控内容信息、进度、状态变更响应，回复给发送端；

断开阶段：

13）发送端发送 RTSP teardown 方法发起断开；

14）接收端，接收端进行本端播放服务回收，并发送 teardown 到发送端；

15）发送端进行本端服务停止，恢复到未连接状态；

双端均可以发送 RTSP teardown 方法发起断开并开始卸载本端的服务能力，等待一段时间，建议不超过 1s，若未等回复，超时后则可直接拆除 RTSP 链路，恢复到未连接状态。

下面章节针对各阶段进行详细交互和报文定义进行展开。

5.5.1 本地资源推送流程

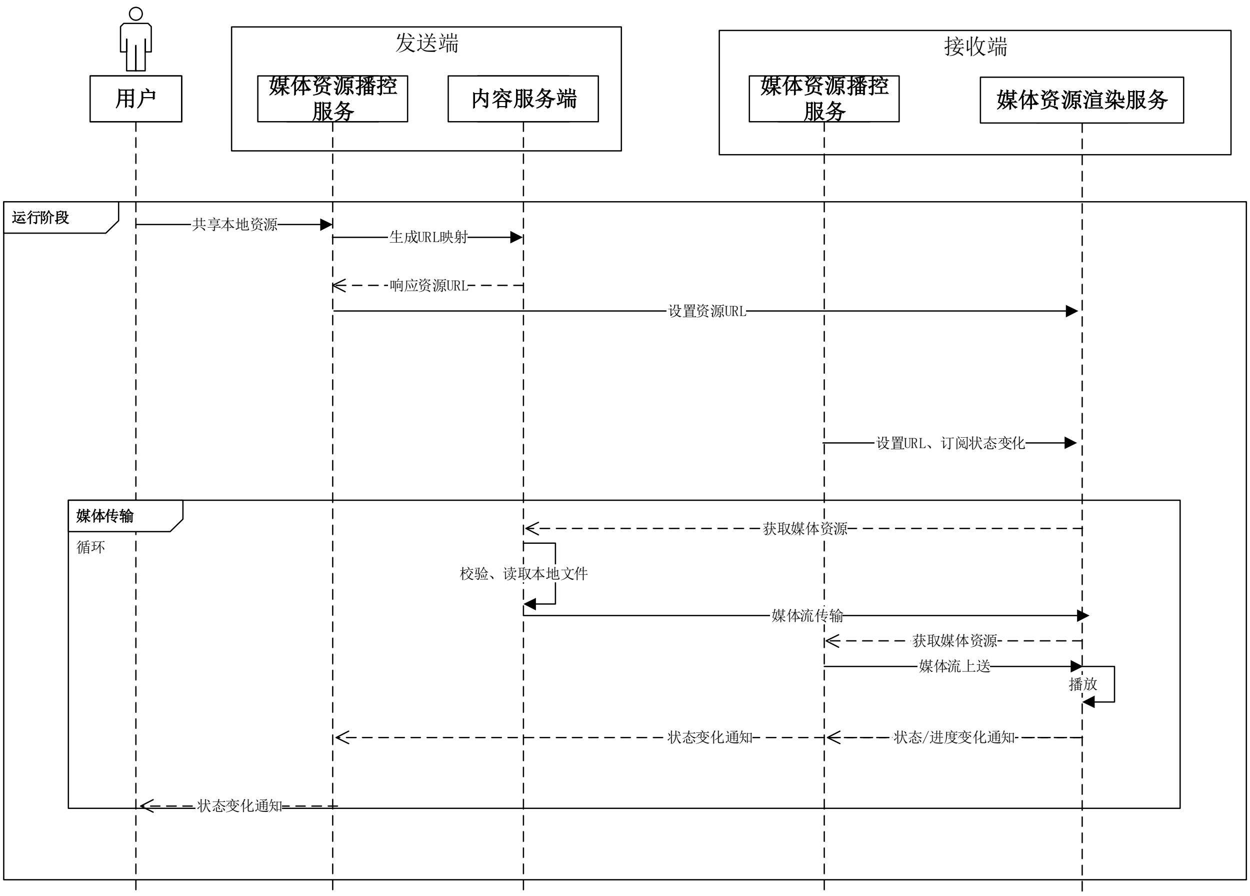


图 本地资源播放流程

1、 用户在媒体应用中，选择待推送的本地资源文件，将本地资源文件及相关媒体信息传递给发送端控制模块。

2、 发送端控制模块调用媒体服务共享模块，将本地文件路径生成 URL 映射，媒体服务共享模块保存该 URL 和本地文件路径的映射。

3、 发送端控制模块将生成的 URL 映射、标题、作者等相关媒体信息通过加密通道传输至接收端控制模块

4、 接收端控制模块通过加密通道接收到报文后，解析媒体信息，校验合法性通过后，设置给媒体渲染器。

5、 接收端媒体渲染服务播放文件，计算读取的内容偏移量及长度，封装成 HTTP Request 请求，在

Content-Range 字段中，携带文件读取起始位置、读取长度，通过加密通道向发送端媒体服务模块请求媒体资源。

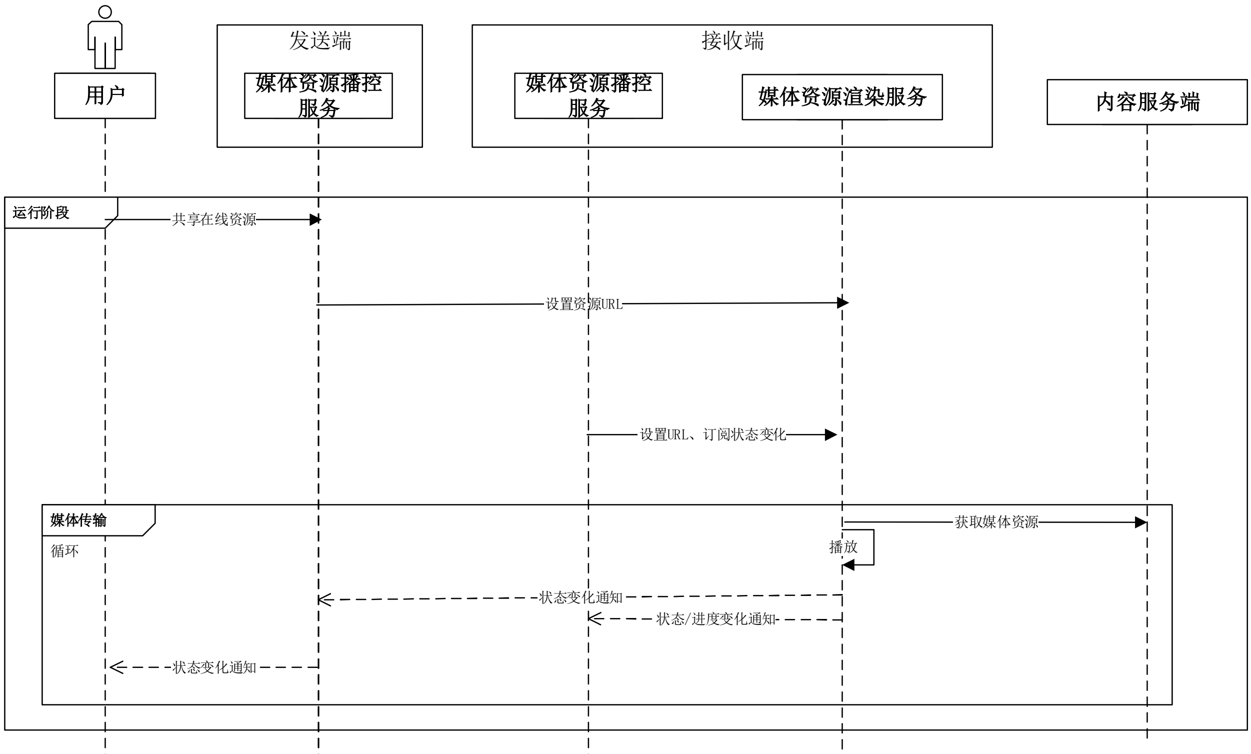
6、发送端媒体服务模块收到接收端发来的请求后，校验请求的合法性。合法性校验通过后，将请求的 URL 重新解析成本地文件路径，并按照 Request 请求，从该文件指定位置读取指定长度内容后，将返回信息封装成 HTTP Response 报文并通过加密通道传输，在 Content-Length 字段中返回本次请求的读取位置和读取长度，并在 HTTP Response Body 中携带读取的文件内容。

7、接收端通过加密通道接收到 Response 报文后，将文件内容传递给接收端媒体渲染器。

8、接收端媒体渲染器正常播放时，将播放进度及状态变化生成回调行为事件封装成 RTSP 报文，并通过加密通道将事件传递给发送端媒体控制器。

9、发送端媒体控制器将事件传递给媒体应用，媒体应用进行相应的响应，绘制界面，通知用户。

5.5.2　在线资源推送流程



1、用户在媒体应用中，选择待推送的在线资源文件，将在线资源文件及相关媒体信息传递给发送端控制模块；

2、发送端控制模块将 URL、标题、作者等相关媒体信息通过加密通道传输至接收端控制模块；

3、接收端控制模块通过加密通道接收到报文后，解析媒体信息，校验合法性通过后，设置给媒体渲染器；

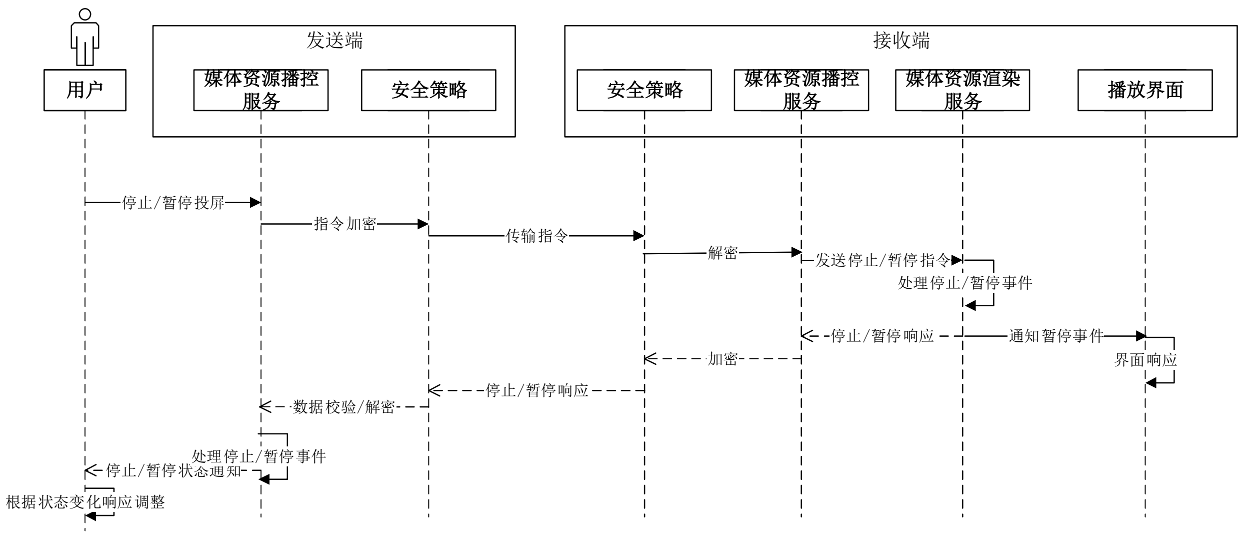
4、接收端媒体渲染器播放文件，计算读取的内容偏移量及长度，封装成 HTTP Request 请求，在 Content-Range 字段中，携带文件读取起始位置、读取长度，向网络媒体服务模块请求媒体资源；

5、网络侧媒体服务模块收到接收端发来的请求后，校验请求的合法性。合法性校验通过后，并按照 Request 请求，将返回信息封装成 HTTP Response 报文，在 Content-Length 字段中返回本次请求的读取位置和读取长度，并在 HTTP Response Body 中携带读取的文件内容；

6、接收端接收到 Response 报文后，将文件内容传递给接收端媒体渲染器；

7、 接收端媒体渲染器正常播放时，将播放进度及状态变化生成回调行为事件封装成 RTSP 报文，并通过加密通道将事件传递给发送端媒体控制器；
8、 发送端媒体控制器将事件传递给媒体应用，媒体应用进行相应的响应，绘制界面，通知用户。

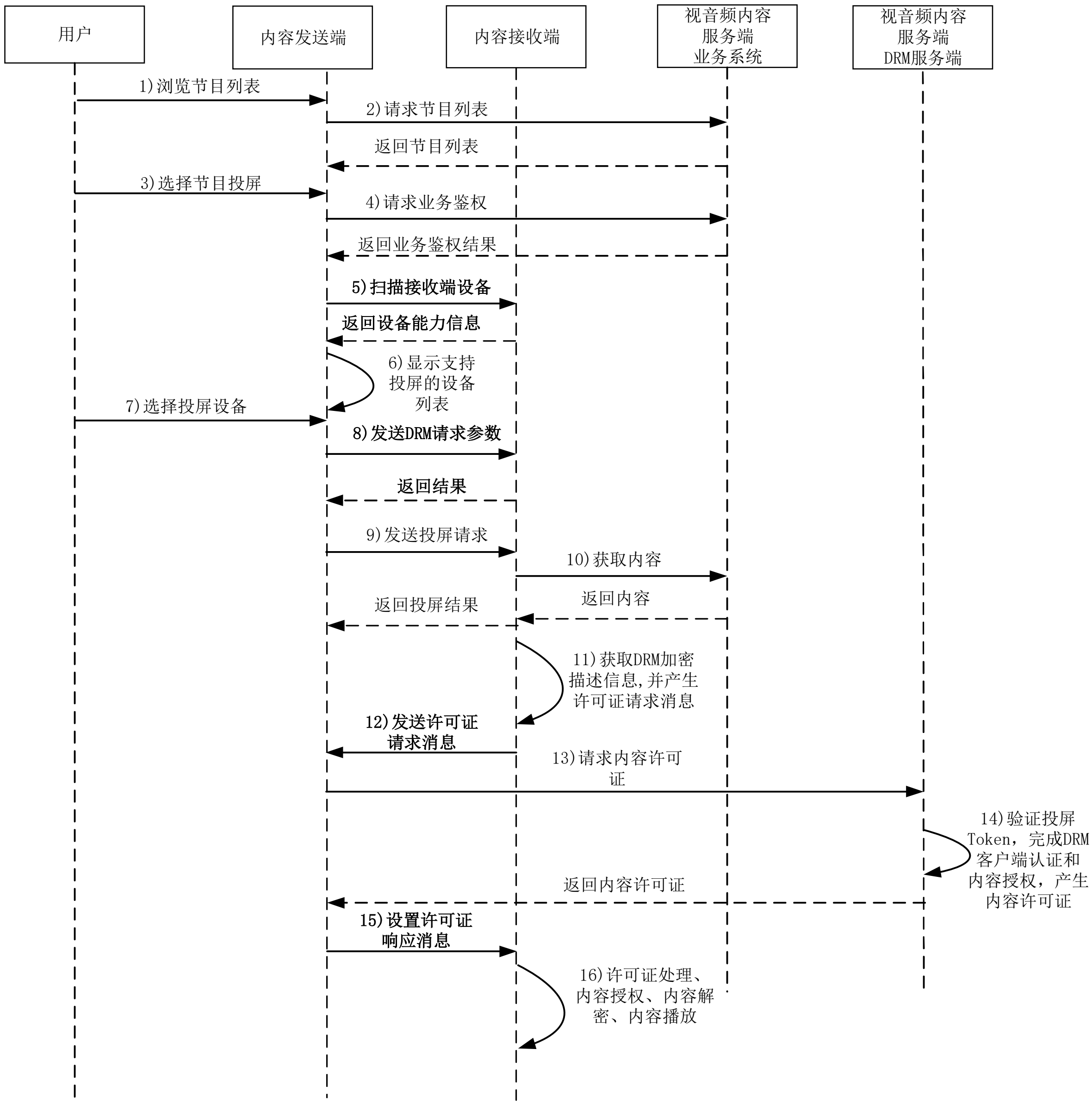
5.5.3 发送端播控操作流程



1、用户在媒体应用中，对当前媒体渲染器进行响应的控制，例如暂停操作。
2、 发送端控制模块用户操作封装成 RTSP 报文，通过加密通道传输至接收端控制模块
3、 接收端控制模块通过加密通道接收到报文后，解析操作命令，校验合法性通过后，设置给媒体渲染器。
4、 接收端进行响应，将状态变化生成回调行为事件封装成 RTSP 报文，并通过加密通道将事件传递给发送端媒体控制器。
5、 发送端媒体控制器接收到接收端发来的 RTSP 报文后，校验回调行为事件合法性，更新发送端控制模块状态
6、将事件传递给媒体应用，媒体应用进行相应的响应，绘制界面，通知用户。

5.5.4 DRM 资源投屏流程

Cast+ DRM 内容保护流程如下图所示。



流程如下：

1) 用户在内容发送端浏览节目列表；

2) 内容发送端从视音频内容服务端业务系统请求节目列表;视音频内容服务端业务系统返回节目列表。节目列表中包括业务鉴权参数、内容 DRM 信息。

3) 用户选择节目投屏。

4) 内容发送端根据业务鉴权参数，和视音频内容服务端交互，完成业务鉴权，获取内容播放链接、投屏认证令牌等业务鉴权结果。

5) 内容发送端通过投屏协议接口扫描接收端设备，内容接收端返回设备信息。设备信息包含设备 DRM 能力信息，含 DRM 类型、安全级别等。

6) 内容发送端在界面显示支持投屏的设备列表。

7) 用户选择投屏设备。

8) 内容发送端根据内容DRM信息、投屏设备的DRM能力信息，判断内容要求的DRM类型、安全级别是否可以被投屏设备支持，如支持，将包括DRM类型drmSchema、投屏认证令牌的DRM请求参数信息通过投屏协议扩展接口设置到内容接收端；如不支持，返回错误，停止投屏。
9) 内容发送端通过投屏协议接口将投屏请求消息发送到内容接收端。
10) 内容接收端使用内容播放链接从视音频内容服务端业务系统获取内容，返回投屏结果。
11) 内容接收端解析内容中的DRM加密描述信息，将DRM加密描述信息和投屏认证令牌发送给DRM客户端获取许可证请求消息。
12) 内容接收端将许可证请求消息通过投屏协议扩展接口发送到内容发送端请求内容许可证。
13) 内容发送端根据内容DRM信息中的licenseURL将许可证请求消息转发到视音频内容服务端的DRM服务端。
14) 视音频内容服务端的DRM服务端验证许可证请求消息中的投屏认证令牌，完成DRM客户端认证和内容授权，产生许可证响应消息，并返回内容发送端。
15) 内容发送端将许可证响应消息通过投屏协议扩展接口设置到内容接收端。
16) 内容接收端完成内容许可证处理、内容授权、解密及播放，并返回投屏结果给内容发送端。

如许可证需要更新，可以重复上述步骤11)-15)更新许可证。

内容接收端应支持状态反馈机制，将DRM状态反馈至内容发送端。

具体可参考《视音频内容分发数字版权管理 超分发技术规范 第1部分：跨终端DRM内容保护》

6 设备管理

6.1 设备发现

6.1.1 发现流程

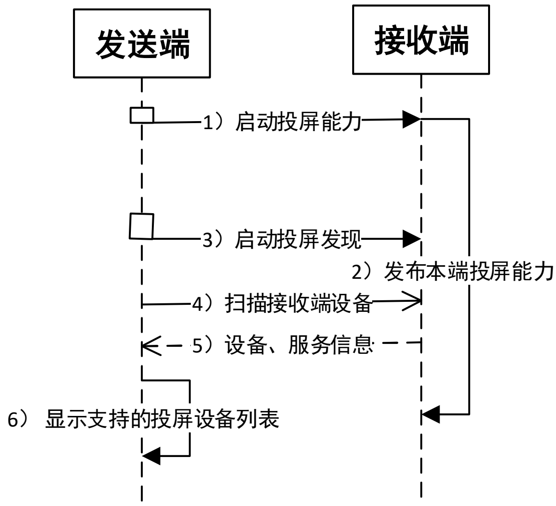


图7 投屏服务发现流程

流程如下：

1) 用户启动接收端投屏业务开关，使能接收端设备管理服务；
2) 设备管理服务通过设备发现协议发布当前设备名称、设备类型、投屏服务类型等能力；
3) 用户启动发送端设备进行服务发现，使能发送端设备管理服务。
4) 设备管理服务通过发现协议进行投屏服务查询。
5) 通过发现协议获取接收端设备信息。
6) 发送端在界面显示支持投屏的设备列表。

6.1.2 设备发现协议报文

本标准发现遵循 mDNS 协议进行发现，本节描述投屏设备发现过程中在 mDNS 的协议配置的报文。

发现报文配置：

名称	类型	描述	可选/必选
服务名称	string	厂商定义，源端设备可以显示为设备名称，采用 UTF-8 编码方式，长度不超过 32bytes。	必选
服务类型	string	_cast-remote._tcp.local	必选
端口	Int16	接收端发布服务前通过套接字服务端（SocketServer）随机生成的等待连接认证端口。	必选

TXT record 配置：

遵照 mDNS 协议，TXT 记录以键值对的方式对支持的能力进行描述，键值均以 UTF-8 进行编码。

键	值类型	描述	可选/必选
deviceid	string	设备标识，可随机生成，考虑用户体验，建议落盘存储，最小长度不低于 32bytes，最大长度不超过 64 bytes。	必选
devicetype	Int32	设备类型。	必选
features	Int32	特性,以 bit 进行定义的。	必选
deviceExtraInfo	String	设备扩展信息	可选

DeviceType 值的定义：

类型	值	描述
智能手机	0x0001	智能手机
智能 PAD	0x0002	智能 PAD
个人电脑	0x0003	个人电脑
智能电视	0x0004	智慧屏电视
机顶盒	0x0005	机顶盒
OTT 盒子	0x0006	网络盒子
Dongle	0x0007	投屏器
智能音箱	0x0008	智能音箱
投影仪	0x0009	投影仪
电子白板	0x000A	电子白板
智能显示器	0x000B	智能显示器
智能座舱	0x000C	智能座舱

deviceExtraInfo 的定义：

JSON 键	值类型	描述	可选/必选
Device			
CodecCap	String	播放器支持的解码能力，默认支持 H.264，H.265，不用单独配置。9.1.6 编解码格式中可选项，可配置。比如支持 H.266 ，可作为值配置，多个值使用分号分割。	可选
ScreenResolution	String	屏幕支持的最高分辨率，如 1080P 可配置：1920x1080	可选

JSON 键	值类型	描述	可选/必选
ScreenFPS	Int16	屏幕最高刷新率，如：60， 90 等 fps	可选
AudioEffect	Int8	1-DOLBY 杜比音效 2-HISTEN 音效 3-Audio Vivid	可选
HDRCap	Int8	0x0001-HDR Vivid 0x0002-HDR10 0x0004-HDR10+ 0x0008-DOLBY VISION	可选
AdsEnc	Int8	支持多链接无缝衔接播放能力，可以优化支持广告播放。0：不支持 1：支持	可选
DRMCap	String	DRM 能力，以 JSON 格式配置。	可选

features 域的定义：

BIT	名称	描述
0	Stream_Video	是否支持视频播放 0：不支持 1：支持
1	Stream_Audio	是否支持音频播放 0：不支持 1：支持
2	Stream_Photo	是否支持图片查看 0：不支持 1：支持
3	Screen_Mirror	是否支持镜像显示能力 0：不支持 1：支持
4	Screen_4K	是否屏幕分辨率支持 4K 且具备 4K 资源播放和渲染能力 0：不支持 1：支持-
5	Screen_8K	是否屏幕分辨率支持 8K 且具备 8K 资源播放和渲染能力 0：不支持 1：支持
6	Network available	是否正常访问网络 0：不支持 1：支持
7~31	reserved	保留，默认为 0

drmCapabilites 的定义：

JSON 键	值类型	描述
drmSchema	Int8	DRM 类型， 1-ChinaDRM 2-Widevine 3-Playready 4-FairPlay
securityLevel	String	DRM 客户端安全级别，如采用 GY/T 277-2019 标准，软件安全级别固定为“SW”，硬件安全级别固定为“HW”，增强硬件安全级别固定为“EH”。

6.2 连接认证鉴权

6.2.1 基本要求

发送端和接收端是多屏协同连接认证鉴权主体；在连接首次建立时，双端需完成双向鉴权，即发送端对接收端进行认证鉴权，同时接收端对发送端也进行认证鉴权。根据双端是否互为可信设备，认证鉴权可分为绑定流程或认证流程。

绑定流程用于双端设备中任一端没有对端的可信凭证状态时，基于用户双端带外共享、6 位数字的认证码完成绑定流程，协商产生会话密钥的过程；同时若用户选择建立长期信任关系，则将对端的设备信息（如设备 ID 等）与对端公钥进行绑定，记录绑定关系及凭证。绑定过程中，用户需要在双端设备上均有确认操作，防止误连与恶意连接。

认证流程用于双端均已基于绑定流程完成了互信关系并保存了所需要凭证，通过绑定关系中对应的公钥完成对对端设备的认证鉴权，并生成会话密钥的过程。认证过程中，用户不必须在双端设备上均有确认操作。

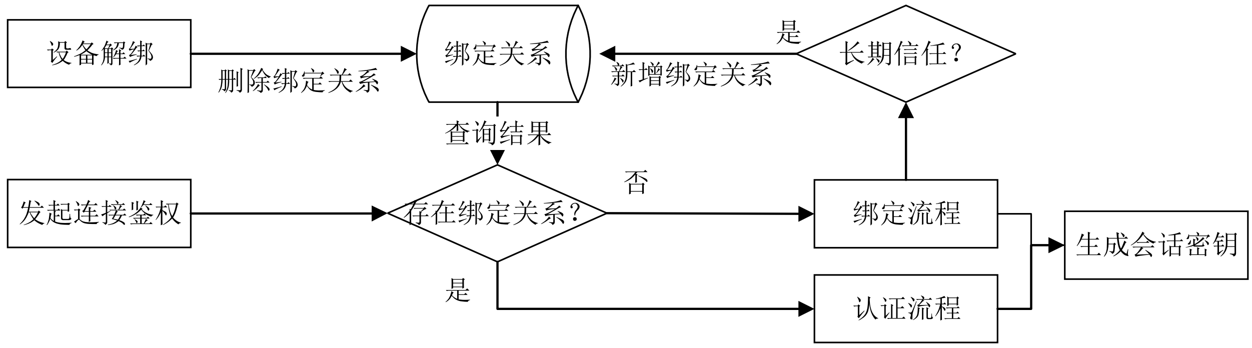


图 7：设备可信关系增删与连接鉴权总体流程

6.2.2　设备绑定流程

首次认证的双端设备，需要进行设备绑定流程。采用标准 SPEKE 协议[ISO/IEC 11770-4:2017]。其中，认证码应由发送端或接收端采用安全的随机数生成，并通过带外传输至对端设备：如可以采用接收端设备生成并显示认证码给用户，由用户手动输入至发送端设备中。绑定流程首先采用 SPEKE 密钥协商出对称密钥，并根据该对称密钥加密交换得到对端的公钥，并将对端设备在发现交互中的 ID 与公钥进行绑定。SPEKE 协议由 TCP 报文承载，所有报文均遵从 JSON 格式，具体流程如下：

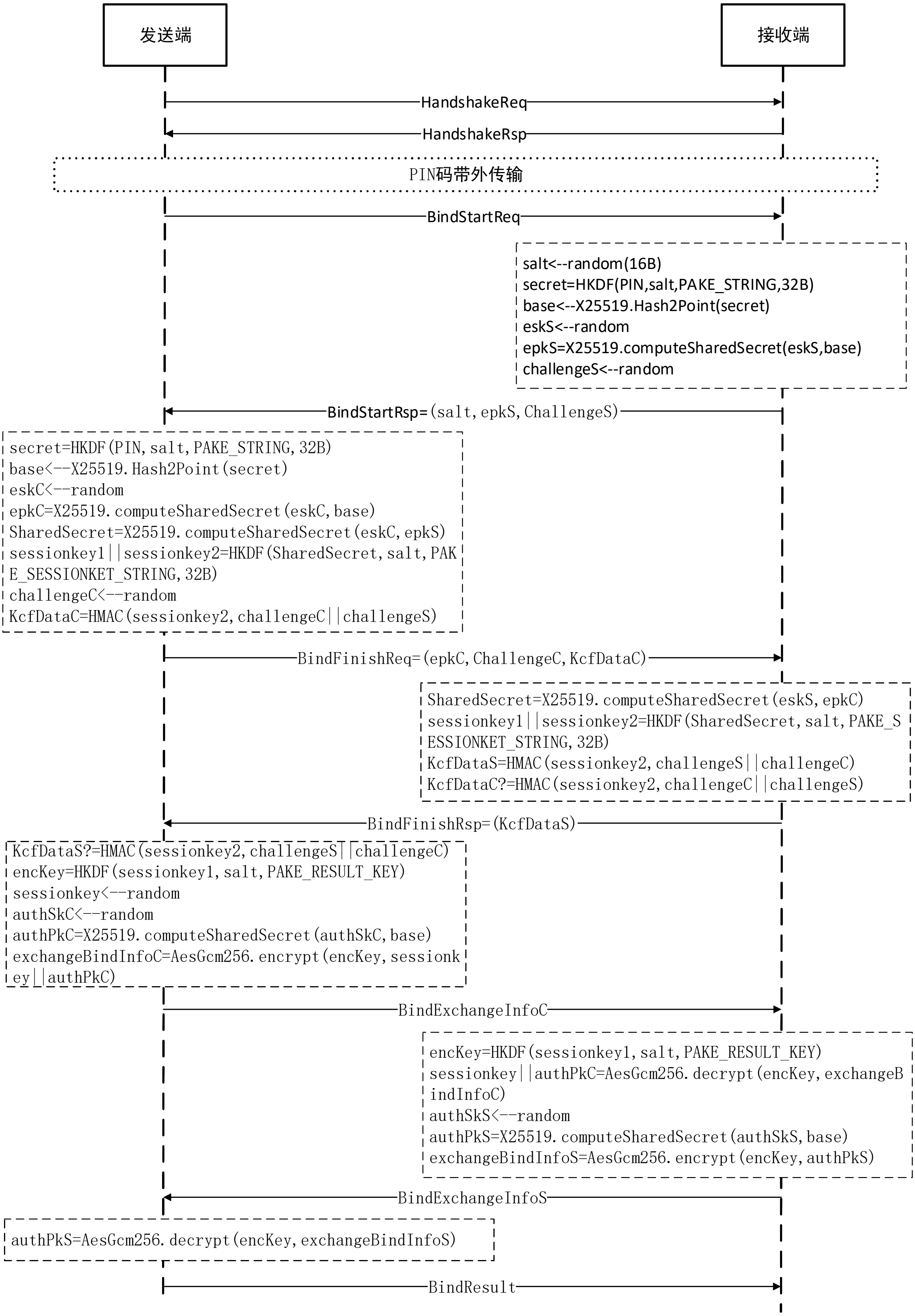

发送端
接收端
HandshakeReq
HandshakeRsp
PIN码带外传输
BindStartReq
salt<--random(16B)
secret=HKDF(PIN,salt,PAKE_STRING,32B)
base<--X25519.Hash2Point(secret)
eskS<--random
epkS=X25519.computeSharedSecret(eskS,base)
challengeS<--random
BindStartRsp=(salt, epkS, ChallengeS)
secret=HKDF(PIN, salt, PAKE_STRING, 32B)
base<--X25519. Hash2Point(secret)
eskC<--random
epkC=X25519. computeSharedSecret(eskC, base)
SharedSecret=X25519. computeSharedSecret(eskC, epkS)
sessionkey1||sessionkey2=HKDF(SharedSecret, salt, PAKE_SESSIONKET_STRING, 32B)
challengeC<--random
KcfDataC=HMAC(sessionkey2, challengeC||challengeS)
BindFinishReq=(epkC, ChallengeC, KcfDataC)
SharedSecret=X25519. computeSharedSecret(eskS, epkC)
sessionkey1||sessionkey2=HKDF(SharedSecret, salt, PAKE_SESSIONKET_STRING, 32B)
KcfDataS=HMAC(sessionkey2, challengeS||challengeC)
KcfDataC?=HMAC(sessionkey2, challengeC||challengeS)
BindFinishRsp=(KcfDataS)
KcfDataS?=HMAC(sessionkey2, challengeS||challengeC)
encKey=HKDF(sessionkey1, salt, PAKE_RESULT_KEY)
sessionkey<--random
authSkC<--random
authPkC=X25519. computeSharedSecret(authSkC, base)
exchangeBindInfoC=AesGcm256. encrypt(encKey, sessionkey||authPkC)
BindExchangeInfoC
encKey=HKDF(sessionkey1, salt, PAKE_RESULT_KEY)
sessionkey||authPkC=AesGcm256. decrypt(encKey, exchangeBindInfoC)
authSkS<--random
authPkS=X25519. computeSharedSecret(authSkS, base)
exchangeBindInfoS=AesGcm256. encrypt(encKey, authPkS)
BindExchangeInfoS
authPkS=AesGcm256. decrypt(encKey, exchangeBindInfoS)
BindResult

图 8：绑定协议交互

1、HandshakeReq：握手请求消息

名称	值类型	描述
Version	String	支持协议版本号， 当前为 1.0
OperType	Int32	1，代表握手命令
Deviceid	string	隐私要求比较高的，可随机生成，考虑用户体验，随机生成后建议落盘存储，最大长度不超过 64bytes。
DeviceName	String	建议长度不超过 32bytes
sequenceNumber	Int32	命令的序列号
isGenericTrusted	Boolean	是不是支持通用认证码模式可信
isPwdTrusted	Boolean	是不是支持密码模式可信
authVersion	String	绑定或认证流程使用的鉴权算法集版本号，默认为 1.0

2、HandshakeRsp：握手请求回复

名称	值类型	描述
Version	String	支持协议版本号， 当前为 1.0
OperType	Int32	1，代表握手命令
handshakeResult	Int32	握手结果 HANDSHAKE_SUCCESS = 5 DEVICE_BUSY = 4 HANDSHAKE_FAILED = 255
authVersion	String	绑定或认证流程使用的鉴权算法集版本号
sequenceNumber	Int32	命令的序列号，来自对应的 HandshakeReq
isGenericTrusted	Boolean	HandShakeReq 中为 true 且本端识别对端为 PIN 码可信时为 true，否则为 false。
isPwdTrusted	Boolean	HandShakeReq 中为 true 且本端识别对端为密码可信时为 true，否则为 false。
isConfirmed	Boolean	是否已确认
allowedAlways	Boolean	保留可信凭证，由业务或者用户参与后确定，0：不保留凭证，仅本次可信；1：保留可行凭证。
authVersion	String	绑定或认证流程使用的鉴权算法集版本号，默认为 1.0

handshakeResult 为 DEVICE_BUSY 代表接收端设备正在忙碌，不支持抢断，握手失败，发送端应停止连接。为 HANDSHAKE_SUCCESS，说明双端是就绪状态，可以发起后续流程。
isPwdTrusted 和 isGenericTrusted 均为 True 时，优选密码模式。

3、BindStartReq：绑定流程开始-请求消息

名称	值类型	描述
Version	String	支持协议版本号， 当前为 1.0
OperType	Int32	2，代表绑定命令

4、BindStartRsp：绑定流程开始-回复消息

名称	值类型	描述
Version	String	支持协议版本号， 当前为 1.0
OperType	Int32	2，代表绑定命令
Salt	byte[16]	16 字节随机数作为盐值。
epkS	string	接收端的 DH 协商公钥值。
challengeS	byte[16]	16 字节随机数作为接收端挑战值。

接收端调用随机数生成16字节盐值salt，随机生成ECDH私钥eskS，并根据salt和认证码计算得到epkS。
接收端生成的16字节随机数，作为挑战值challengeS。

5、BindFinishReq：绑定流程结束-请求消息

名称	值类型	描述
Version	String	支持协议版本号， 当前为1.0
OperType	Int32	3，代表绑定结束命令
epkC	string	发送端的DH协商公钥值。
challengeC	byte[16]	16字节随机数作为发送端挑战值。
KcfDataC	byte[32]	发送端认证mac结果。

发送端随机生成ECDH私钥eskC，并根据盐值和认证码计算得到epkC。
发送端根据eskC和epkS计算得到会话密钥，并派生出Sessionkey1和Sessionkey2，分别用于加密和HMAC计算。
发送端生成随机数challengeC。
发送端根据Sessionkey2计算challengeC||challengeS的HMAC值，即KcfDataC。

6、BindFinishRsp：绑定流程结束-回复消息

名称	值类型	描述
Version	String	支持协议版本号， 当前为1.0
OperType	Int32	3，代表绑定结束命令
KcfDataS	byte[32]	接收端认证mac结果。

接收端根据eskS和epkC计算得到会话密钥，并同样派生出Sessionkey1和Sessionkey2，用于加密和HMAC计算。
接收端根据Sessionkey2计算challengeC||challengeS的HMAC值，即KcfDataC’并，并比对是否等于对端传来的KcfDataC，否则鉴别失败，结束流程。
接收端根据Sessionkey2计算challengeS||challengeC的HMAC值，即KcfDataS。
接收端根据Sessionkey1派生出加密密钥DataEncKey。

7、BindExchangeInfoC：交换密钥-请求消息

名称	值类型	描述
Version	String	支持协议版本号， 当前为1.0。
OperType	Int32	4，代表交换秘钥命令。
encSessionKey	Byte[64]	加密的sessionkey，加密密钥为前述产生，sessionkey用于业务数据加密。
encPkC	string	[可选]加密的发送端认证公钥，绑定deviceId，用于长期信任后续的免PIN码认证。

发送端同样根据Sessionkey1派生出加密密钥DataEncKey。
发送端生成16字节随机数作为业务加密密钥Sessionkey，并采用DataEncKey加密Sessionkey得到encSessionKey，加密模式为AES-GCM-128，IV值为随机生成的16字节。
【可选】若为长期信任关系，则发送端生成公私钥对(skC,pkC)，，并采用DataEncKey加密pkC得到encPkC，加密模式为AES-GCM-128，IV值为随机生成的16字节。

8、BindExchangeInfoS：交换密钥-回复消息

名称	值类型	描述
Version	String	支持协议版本号， 当前为1.0

名称	值类型	描述
OperType	Int32	4，代表交换秘钥命令。
encResult	Byte[64]	加密的密钥交换结果。至少需要 16 字节 IV+32 字节校验值+1 位结果。
encPkC	String	[可选]加密的接收端认证公钥，绑定 deviceId，用于长期信任后续的免 PIN 码认证。

接收端解密 encSessionKey，得到 sessionkey，若 gcm 模式解密校验失败则 Result=false，否则将 sessionkey 作为业务加密密钥。

【可选】若为长期信任关系，则接收端生成公私钥对(authSkS, authPkS)，并采用 DataEncKey 加密 pkS 得到 encPkS，加密模式为 AES-GCM-128，IV 值为随机生成的 16 字节。同时解密 encPkC 得到 authPkC，若 gcm 模式解密校验失败则 Result=false；否则 Result=true。

9、ExchangeBindFinish：绑定确认消息

名称	值类型	描述
Version	String	支持协议版本号， 当前为 1.0
OperType	Int32	5，代表绑定结束命令。
encBindResult	Byte[64]	加密的密钥交换结果。至少需要 16 字节 IV+32 字节校验值+1 位结果。

若为长期信任关系，发送端解密 encPkS 得到 AuthPkS，解密失败则 bindResult=false，否则 bindResult=true 并将 pkS 与接收端 deviceIdS 绑定，并将 (versionIndex，AuthPkS，deviceIdS，authSkC) 四元组存盘记录。返回 bindResult 的加密结果 encBindResult。

接收端收到后查看 bindresult=true，则将 authpkC 与 Client 端 deviceId 绑定。并将 (versionIndex，AuthPkC，deviceIdC，authSkS) 四元组存盘记录。

6.2.3 设备认证

当双端具有可信关系时，执行设备认证过程，此时双端互相已持有对端的公钥。双端基于本端的私钥和对端的公钥，基于 ECDH 算法计算出一个共享的秘密，作为 SPEKE 协议中的 secret，完成设备认证。认证过程得到的对称密钥，作为后续通信的会话密钥使用。

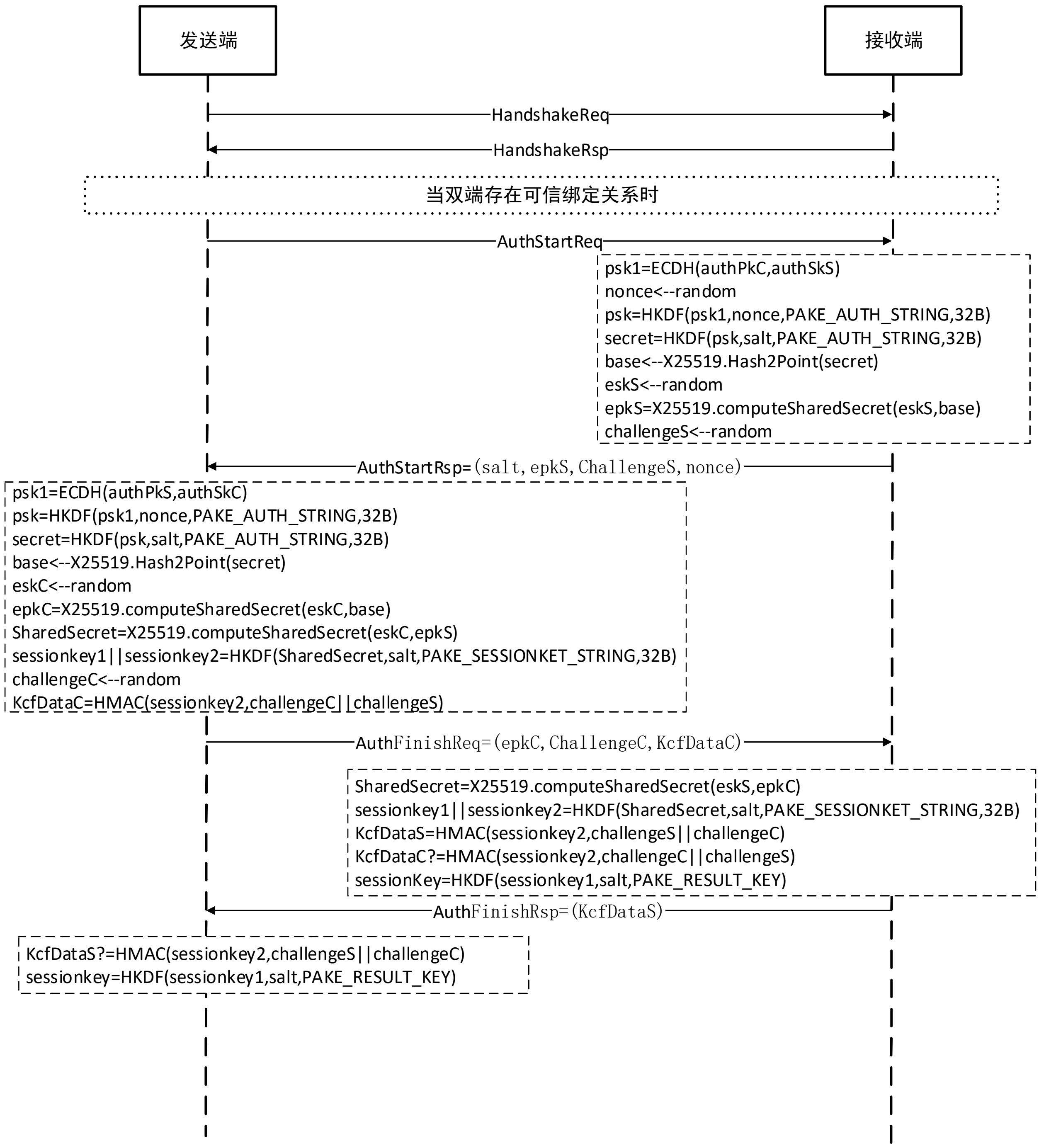


图 9：认证协议交互

1、AuthStartReq：认证流程开始-请求消息

名称	值类型	描述
Version	String	支持协议版本号， 当前为 1.0
OperType	Int32	6，代表鉴权命令
protocolIndex	Int8	选用的鉴别协议序号，后 7 位默认为 0x01。
isAlwaysTrust	Boolean	是否永久信任，作为 flag 放到 protocolIndex 中，占用最高的 1 个 bit 位。

发送端根据双端的 protocolList 交集，选取安全的协议；若双端均为 0x0000 0001，即默认值，则进行以下交互。

2、AuthStartRsp：认证流程开始-回复消息

名称	值类型	描述
Version	String	支持协议版本号， 当前为 1.0
OperType	Int32	6，代表鉴权命令
challengeS	byte[16]	16 字节随机数作为 Server 端挑战值。
nonce	byte[16]	16 字节随机数作为 nonce 值。
salt	byte[16]	16 字节随机数作为盐值。
epkS	string	接收端的 DH 协商公钥值。

接收端调用随机数生成：

16 字节 salt 值、16 字节 challengS 值、16 字节 nonce 值。并查询本地绑定关系得到对端的认证公钥 authPkC 和对应本端私钥 authSkS，计算得 psk。并基于 psk，调用 Hash2Point 函数生成椭圆曲线基点 base。调用随机数生成私钥 eskS，并基于 base 生成公钥 epkS。

3、AuthFinishReq：认证流程结束-请求消息

名称	值类型	描述
Version	String	支持协议版本号， 当前为 1.0
OperType	Int32	7，代表认证完成命令
epkC	string	发送端的 DH 协商公钥值。
challengeC	byte[16]	16 字节随机数作为 Client 端挑战值。
KcfDataC	byte[32]	发送端认证 mac 结果。

并查询本地绑定关系得到对端的认证公钥 authPkS 和对应本端私钥 authSkC，计算得 psk。并基于 psk，调用 Hash2Point 函数生成椭圆曲线基点 base。

发送端随机生成 ECDH 私钥 eskC，并根据盐值和 PIN 码计算得到 epkC。

发送端根据 eskC 和 epkS 计算得到会话密钥，并派生出 Sessionkey1 和 Sessionkey2，分别用于加密和 HMAC 计算。

发送端生成随机数 challengeC。

发送端根据 Sessionkey2 计算 challengeC||challengeS 的 HMAC 值，即 KcfDataC。

4、AuthFinishRsp：认证流程结束-回复消息

名称	值类型	描述
Version	String	支持协议版本号， 当前为 1.0
OperType	Int32	7，代表认证完成命令
KcfDataS	byte[32]	接收端认证 mac 结果。

接收端根据 eskS 和 epkC 计算得到会话密钥，并同样派生出 Sessionkey1 和 Sessionkey2，用于加密和 HMAC 计算。

接收端根据 Sessionkey2 计算 challengeC||challengeS 的 HMAC 值，即 KcfDataC’并，并比对是否等于对端传来的 KcfDataC，否则鉴别失败，结束流程。

接收端根据 Sessionkey2 计算 challengeS||challengeC 的 HMAC 值，即 KcfDataS。

接收端根据 Sessionkey1 及随机派生的 salt 值派生出加密密钥 sessionKey。

发送端同样根据 Sessionkey1 及收到的 salt 值派生出加密密钥 sessionKey。

6.2.4 算法选择

绑定和认证流程使用的验证算法如下，定义为 Ver1.0 为基础版本，可根据业务要求选择其他加密算法代替定位新的加密算法套，但必须支持最基础的版本。

1、HKDF 算法
 密钥推导算法，选择 SHA-256 作为 Hash 函数。
2、HMAC 算法
 HMAC（散列消息鉴别码）中，选择 SHA-256 作为 Hash 函数。
3、ECDH 算法选择
 默认协议版本选择 X.25519 作为 ECDH 算法，用于绑定和认证流程中的密钥协商。
4、Hash2Point 算法
 选择 Elligator2 方案作为 Hash2Point 算法，参考 IETF 标准 https://datatracker.ietf.org/doc/draft-irtf-cfrg-hash-to-curve/，用于将 psk 值转换为椭圆曲线基点。
5、对称加密
 AesGCM128

7 能力协商和播控命令定义

控制通道是建立连接后首个基于TCP协议需要建立的连接，负责双端控制信令交互，所有的信令格式遵从RTSP规范。本规范使用的RTSP协议格式，遵循RTSP V1.0标准[RFC2326]，并在其基础上进一步扩展。

7.1 RTSP 方法

本规范使用了 RTSP 中定义的 ANNOUNCE、OPTIONS、SET_PARAMETER、GET_PARAMETER、TEARDOWN 方法，进行双端状态、播控及响应信令交互。

7.1.1 RTSP ANNOUNCE

该方法用于协商多屏互动时进行双端加密算法，接收端设备连接发送端 RTSP 通道成功后，开始发送。

1） 消息头定义：

ANNOUNCE * RTSP/1.0
CSeq: *实际报文序号*
Content-Type: text/parameters
Content-Length: *实际消息体报文长度*

2） 消息体定义如下：

encrypt_description: encrypt_list=*algStr*

algStr 定义如下：

类型	值定义	描述
String	aes128ctr/ aes128ctr, aes128gcm	控制 RTSP 信令优选 AES128GCM 加密算法，视频内容优先使用 AES128CTR 加密算法。

具体示例如下：encrypt_description: encrypt_list=aes128ctr, aes128gcm

3）响应定义：

同 RTSP1.0,本规范不做要求。

7.1.2 RTSP OPTIONS

该方法用于请求获取对端支持的方法。

1） 消息头定义：

```
OPTIONS * RTSP/1.0
Date: yyyy-MM-dd HH:mm:ss
CSeq: seq
```

2） 消息体定义：

无

3）响应定义：

同 RTSP1.0,本规范不做要求。

7.1.3 RTSP GET_PARAMETER

该方法用于请求获取接收端播放能力参数。没有消息体的 GET_PARAMETER 也可用于测试对端是否在线（类似于"ping"）。参数值携带在消息中。

1） 消息头定义：

```
GET_PARAMETER rtsp://localhost/hisight1.1 RTSP/1.0
Date: 具体日期，格式如 yyyy-MM-dd HH:mm:ss
CSeq: 具体序列号
Content-Type: text/parameters
Content-Length: 实际消息体长度
```

2） 消息体定义：

```
his_player_controller_capability
```

3） 响应定义：

同 RTSP1.0,携带相应参数值。具体参数定义见 7.3 和 7.4。

参考播放示例（发送端->接收端）：

```
GET_PARAMETER rtsp://localhost/hisight1.1 RTSP/1.0
Date: yyyy-MM-dd HH:mm:ss
CSeq: seq
Content-Type: text/parameters
Content-Length: 消息体长度

his_player_controller_capability
```

保活（keep alive）报文：

```
GET_PARAMETER rtsp://localhost/hisight1.1 RTSP/1.0
Date: yyyy-MM-dd HH:mm:ss
CSeq: seq
```

7.1.4 RTSP SET_PARAMETER

该方法用来设置对端的参数值。本规范通过该方法实现发送端控制模块对接收端的播控操作。具体播控操作携带在消息体中，见 xxx 章节。

1）消息头定义：

SET_PARAMETER rtsp://localhost/hisight1.1 RTSP/1.0
Date: 具体日期，格式如 *yyyy-MM-dd HH:mm:ss*
CSeq: 具体序列号
Content-Type: text/parameters
Content-Length: 实际消息体长度

2）消息体定义：消息体定义：

his_execute_method: *method*
module_id: *module_id*
event: *action*
param: *param*

参数值按需携带在消息体中。具体定义如表 2 所示。

表 2 GET_PARAMETER/ SET_PARAMETER 参数值定义

参数名称	必选/可选	参数描述	取值
his_version	可选	协议版本号	
his_execute_method	可选	触发方法执行	当前有三种取值： • SETUP - 控制模块启动播控 • RENDER_READY– 渲染模块通知播放就绪。 • SEND_EVENT_CHANGE – 控制模块进行播放控制
module_id	可选	事件分发的模块，应用于 SEND_EVENT_CHANGE 方法	媒体播控模块标识符 1009
event	可选	需要分发的事件	控制事件 100 控制回调事件 101 本地资源流通道创建事件 102 本地资源流通道销毁事件 103
param	可选	分发的事件详细内容	JSON 格式，由模块解析。 具体的赋值可参考 8.3 播控的定义

- 该方法用来发送播控指令。

报文定义：

SET_PARAMETER rtsp://localhost/hisight1.1 RTSP/1.0
Date: *yyyy-MM-dd HH:mm:ss*
CSeq: *seq*
Content-Type: text/parameters
Content-Length: *消息体长度*

his_execute_method: SEND_EVENT_CHANGE
module_id: 1003(媒体播控模块标识符)
event: *action*
param: *param*

参考播放示例（发送端->接收端）：

SET_PARAMETER rtsp://localhost/hisight1.1 RTSP/1.0
Date: *yyyy-MM-dd HH:mm:ss*
CSeq: seq
Content-Type: text/parameters
Content-Length: *消息体长度*

his_execute_method: SEND_EVENT_CHANGE
module_id: 1009
event: 100
param:{"ACTION":"play","DATA":{"CURRENT_INDEX":*index*,"PROGRESS_INTERVAL":*interval*,"LIST":[{"MEDIA_TYPE":"VIDEO","MEDIA_ID":"*media_id*","MEDIA_URL":"*media_url*","SOURCE_TYPE":"VIDEO","ALBUM_TITLE":"*title*","START_POSITION":*position*,"MEDIA_NAME":"name","VIDEO_TRACK_NAME":"track name"}

参考播放响应示例（接收端->发送端）
SET_PARAMETER rtsp://localhost/hisight1.1 RTSP/1.0
Date: *yyyy-MM-dd HH:mm:ss*
Cseq: seq
Content-Length: *消息体长度*

his_execute_method: SEND_EVENT_CHANGE
module_id: 1009
event: 101
param:{"CALLBACK_ACTION":"onPositionChanged","DATA":{"POSITION":*position*,"BUFFER_POSITION":*buffer_position*,"DURATION":*duration*}}

- RTSP SETUP

报文定义：
SET_PARAMETER rtsp://localhost/hisight1.1 RTSP/1.0
Date: *yyyy-MM-dd HH:mm:ss*
CSeq: *seq*
Content-Type: text/parameters

Content-Length: *body length*

his_execute_method: SETUP

7.1.5 RTSP TEARDOWN

断开连接。请求会停止业务，释放相关资源。

报文定义：

```
TEARDOWN rtsp://localhost/hisight1.1 RTSP/1.0
CSeq: seq
```

7.2 消息场景流程

本节定义了基于流媒体的多屏互动设备之间交换的 RTSP 消息，实现设备参数协商、会话建立和会话管理的功能。

表 3 多屏互动技术中 RTSP 消息

阶段	发起者	数据体	行为	描述
Announce1	接收端	7.1.1	7.3.1	请求加密算法列表
Announce2	发送端	7.1.1	7.3.1	返回协商后加密算法列表
M1	发送端	7.1.2	7.3.2	宣称本端支持方法(可选)
M2	接收端	7.1.2	7.3.2	宣称本端支持方法(可选)
M3	发送端	7.1.3	7.3.2	请求对端参数
M4	发送端	7.1.4	7.3.2	发送端发送经过协商后的参数
M5	发送端	7.1.5	7.3.3	会话建立
TEARDOWN	发送端/ 接收端	7.1.6	7.3.4	断开本次连接

7.2.1 协商加密算法阶段

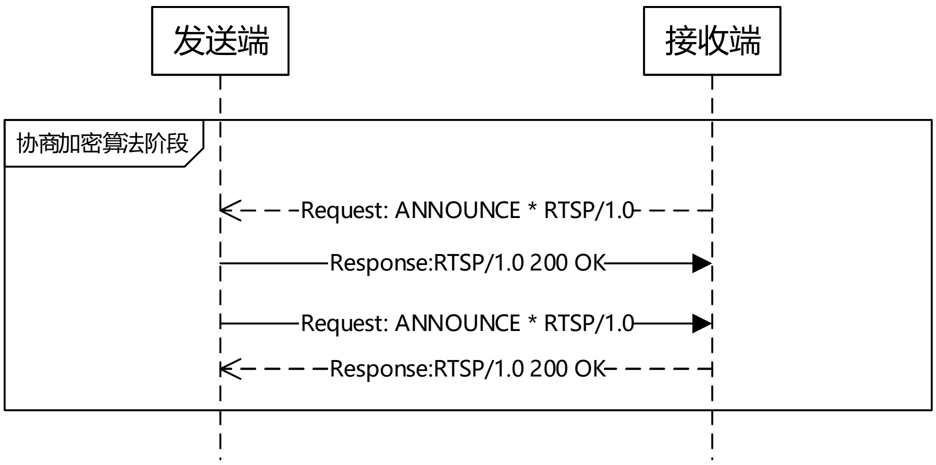


图 1 协商加密算法阶段

- Announce1 阶段：
  接收端主动发起，携带本端加密算法列表发送至对端，当前版本建议支持 AES-CTR-128 和 AES-GCM-128，至少支持 AES-CTR-128；
- Announce2 阶段：
  发送端接收到之后，与本端支持的加密算法列表进行协商，流媒体加密算法选择 AES-CTR-128，播控信令优选 AES-GCM-128，返回协商后加密算法列表给接收端；

7.2.2 连接协商阶段

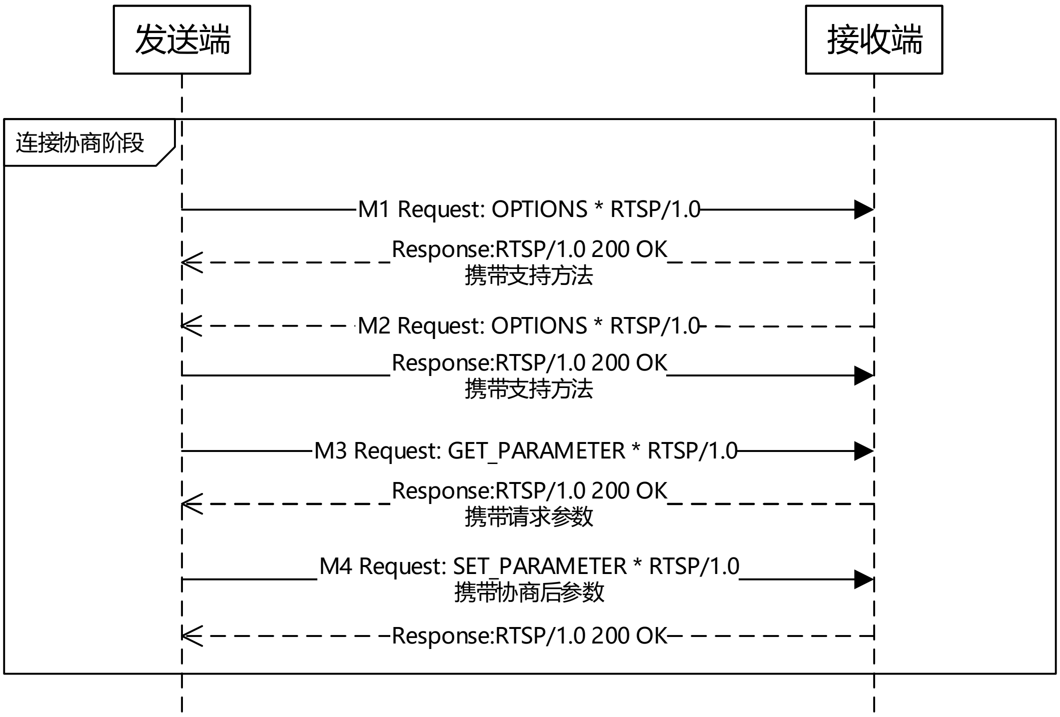


图 2 连接协商阶段

- M1&M2 阶段：
  该阶段双端皆可发起，请求对端设备支持的方法；至少支持“SETUP, TEARDOWN, GET_PARAMETER, SET_PARAMETER”；
- M3 阶段：
  发送端发起，向接收端请求媒体参数；接收端接收到请求之后，按照请求参数列表进行对应填充，并回复响应报文携带填充后的报文给发送端；
- M4 阶段：
  发送端发起，接收到接收端 M3 阶段的响应报文后，逐个参数进行协商，向接收端发送协商之后的参数列表；接收端接收该请求，并将协商后参数进行保存；双端协商完毕。

7.2.3 播控阶段

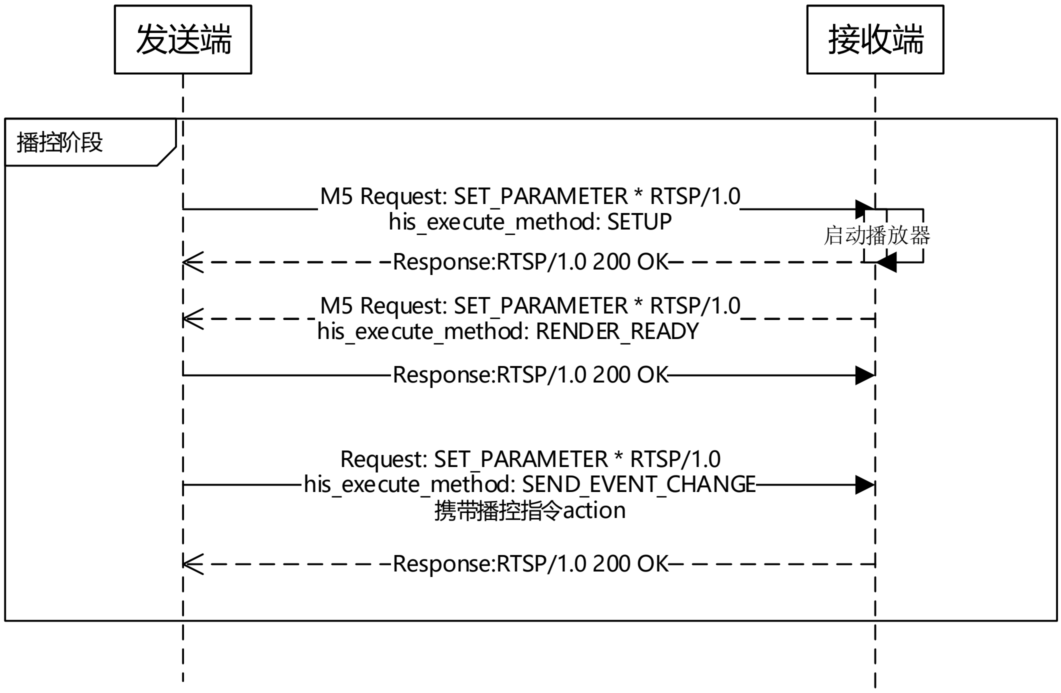


图 3 播控阶段

M5	发送端	7.1.5	7.3.3	会话建立
播控命令	发送端/接收端	7.14	7.3.3	发送端发送播控命令，接收端进行响应。
TEARDOWN	发送端/ 接收端	7.1.6	7.3.4	断开本次连接

- M5 阶段：
  发送端发起，会话建立；

7.2.4 断开连接

断开连接从发送端或者接收端都可以出发，发送 TEARDOWN 消息后，资源销毁，该会话结束。

7.3 设备能力参数

在双端交互过程中，会携带接收端相关能力，传给发送端，包括但不限于是否支持4K、是否支持DRM及支持的类型、接收端初始音量、接收端DRM能力，以JSON格式封装传递。

关键字	必选/可选	值类型	说明
SUPPORT_RESOLUTION_4K	可选	int	接收端是否支持 4K 解码播放及渲染能力 0：不支持 1： 支持
MEDIA_VOLUME	必选	int	接收端初始音量，0~100
DRM_CAPABILITY_PROPERTIES	必选	JSON String	DRM_TYPE_CLEARKEY; DRM_TYPE_WIDEVINE; DRM_TYPE_PLAYREADY; DRM_TYPE_CHINADRM; 接收端通过获取到的接收端 DRM 能力

			以 JSON String 格式填充信息。
DECODE_CAPABILITY	可选	String	H264 H265 H266 VP8 VP9
SOUND_EFFECT	可选	Int	DOLBY 杜比音效 HISTEN 音效

7.4 播控质量参数

用于发送端对接收端播放内容的用户体验收集，用户体验主要包括播放成功、起播时间、重新缓冲以及视觉质量等因素。

关键字	必选/可选	值类型	说明
PLAY_SUCCESS	必选	int	接收端是否播放成功，0：未播放成功 1：播放成功
START_PLAY	必选	int	起播时长，从接收到链接到首帧播放耗时
CACHE_TIME	必选	int	重新缓冲

7.5 播控链路保活

通过发送端发送GET_PARAMETE方法到接收端并在预定的时间内（建议不低于30s，不高于60s）接收端接收端端设备的响应，确保接收端设备是保活状态。每次探测间隔为120s，探测失败可以重试一次，如果仍失败认为对端设备异常，退出播放状态。

8 媒体资源播控

8.1 概述

作为控制模块，主要完成媒体资源推送及媒体资源的播放控制。发送端控制模块接收用户触发的控制动作，比如播放、暂停、停止等，并以密文的形式发送给接收端控制模块，并由其下发给媒体渲染模块进行响应，媒体渲染模块进行相应操作后并将状态响应同步给接收端控制模块。接收端控制模块通过控制通道反馈给发送端控制模块，最终呈现给用户。

8.2 播放操作及响应定义

发送端控制模块向接收端的媒体资源播控服务发出播控请求，接收端进行请求命令解析后，调用媒体资源渲染服务执行相关操作，获取操作执行结果，该动作、结果和错误封装在RTSP报文中，通过RTSP加密报文响应，发送端接收到响应后，更新播放状态。

所有的播控命令均以JSON格式组成，分为ACTION、CALLBACK_ACTION和DATA，ACTION表示请求的动作，CALLBACK_ACTION表示响应的回调，DATA表示输入/输出值。

8.2.1 播放命令定义

ACTION	DATA	类型	描述	可选/必选
play	CURRENT_INDEX	int	当前音频/视频/图片资源在列表中的索引	必选

ACTION	DATA	类型	描述	可选/必选
	PROGRESS_INTERVAL	int	刷新间隔时间，单位毫秒。配置后由接收端设备按照配置值进行主动同步播放进度。 未配置时，默认每 60 秒同步一次。	可选
	LIST	List<PlayInfo>	播放信息列表实体类 PlayInfo 定义 参考 8.2.1.1	必选

8.2.1.1 播放信息（PlayInfo）的全部字段信息

字段	类型	描述
KEY_MEDIA_ID	String	媒体资源 ID
KEY_MEDIA_NAME	String	媒体资源名称
KEY_MEDIA_URL	String	媒体资源 URL
KEY_MEDIA_TYPE	String	媒体资源类型
KEY_MEDIA_SIZE	Long	媒体文件大小，单位字节
KEY_START_POSITION	Int	媒体资源开始播放的起始位置
KEY_DURATION	Int	媒体资源总时长
KEY_CLOSING_CREDITS_POSITION	Int	媒体资源片尾位置
KEY_ALBUM_COVER_URL	String	专辑封面 URL
KEY_ALBUM_TITLE	String	专辑名称
KEY_MEDIA_ARTIST	String	媒体资源作者名称
KEY_LRC_URL	String	歌词 URL
KEY_LRC_CONTENT	String	歌词内容
KEY_APP_ICON_URL	String	北向应用 icon 的 URL
KEY_APP_NAME	String	北向应用名称
KEY_DRM_TYPE	Int	DMR 证书类型，比如 Widevine DMR、ChinaDRM 等
KEY_VIDEO_TRACK_NAME	String	视频轨名称
KEY_AUDIO_TRACK_NAME	String	音频轨名称

8.2.2 播放命令的响应定义

8.2.2.1 播放命令的响应一

接口调用成功的响应

CALLBACK_ACTION	DATA	类型	描述
onMediaItemChanged	playInfo	PlayInfo	返回当前播放的媒体资源

注意：onMediaItemChanged 返回的 PlayInfo 仅包含以下部分数据

DATA	类型	描述
MEDIA_ID	string	媒体资源 ID
MEDIA_NAME	string	媒体资源名称

DATA	类型	描述
MEDIA_ARTIST	string	媒体资源作者名称
APP_NAME	string	北向应用名称
MEDIA_TYPE	string	媒体资源类型
ALBUM_TITLE	string	专辑名称

8.2.2.2　播放命令的响应二

资源播放成功的响应

CALLBACK_ACTION	DATA	类型	描述
onPlayerStatusChanged	PLAYBACK_STATE	int	播放状态，1/2/3/4 分别表示 初始化中/缓冲中/准备就绪/列表播放完毕
	IS_PLAY_WHEN_READY	boolean	是否正在播放中

CALLBACK_ACTION	DATA	类型	描述
onPositionChanged	POSITION	int	播放位置，单位毫秒
	BUFFER_POSITION	int	缓冲位置，单位毫秒
	DURATION	int	媒体资源总时长，单位毫秒

8.2.3　暂停命令定义

ACTION	DATA	类型	描述	可选/必选
Pause	NA	NA	NA	NA

8.2.4　继续命令定义

ACTION	DATA	类型	描述	可选/必选
Resume	NA	NA	NA	NA

8.2.5　停止命令定义

ACTION	DATA	类型	描述	可选/必选
Stop	NA	NA	NA	NA

8.2.6　暂停/继续/停止命令的响应定义

CALLBACK_ACTION	DATA	类型	描述
onPlayerStatusChanged	PLAYBACK_STATE	int	播放状态，1/2/3/4 分别表示 初始化中/缓冲中/准备就绪/列表播放完毕
	IS_PLAY_WHEN_READY	boolean	是否正在播放中

8.2.7　新增媒体项命令定义

ACTION	DATA	类型	描述	可选/必选
addMediaItem	playInfo	PlayInfo	需要新增的媒体资源	必选

注意：输入的 PlayInfo 应包含全部的字段数据

8.2.8　新增媒体项命令的响应定义

无

8.2.9　播放下一条命令的定义

ACTION	DATA	类型	描述	可选/必选
next	NA	NA	NA	NA

8.2.10　播放前一条命令的定义

ACTION	DATA	类型	描述	可选/必选
previous	NA	NA	NA	NA

8.2.11　播放下一条/上一条命令的响应定义

CALLBACK_ACTION	DATA	类型	描述
onMediaItemChanged	playInfo	PlayInfo	返回当前播放的媒体资源

注意：onMediaItemChanged 返回的 PlayInfo 仅包含以下部分数据

DATA	类型	描述
MEDIA_ID	string	媒体资源 ID
MEDIA_NAME	string	媒体资源名称
MEDIA_ARTIST	string	媒体资源作者名称
APP_NAME	string	北向应用名称
MEDIA_TYPE	string	媒体资源类型
ALBUM_TITLE	string	专辑名称

8.2.12　指定位置播放命令的定义

ACTION	DATA	类型	描述	可选/必选
seek	POSITION	int	指定的播放位置，单位毫秒	必选

8.2.13　快进命令的定义

ACTION	DATA	类型	描述	可选/必选
fastForward	DELTA	int	快进时长，单位毫秒	必选

8.2.14　快退命令的定义

ACTION	DATA	类型	描述	可选/必选
fastRewind	DELTA	int	快退时长，单位毫秒	必选

8.2.15　指定位置/快进/快退播放命令的响应定义

CALLBACK_ACTION	DATA	类型	描述
onPositionChanged	POSITION	int	播放位置，单位毫秒
	BUFFER_POSITION	int	缓冲位置，单位毫秒

CALLBACK_ACTION	DATA	类型	描述
	DURATION	int	媒体资源总时长，单位毫秒

8.2.16 设置重播模式命令的定义

ACTION	DATA	类型	描述	可选/必选
setRepeatMode	MODE	int	循环模式 0/1/2/3 分别为 关闭/单曲循环/列表循环/随机播放	必选

8.2.17 设置重播模式命令的响应定义

CALLBACK_ACTION	DATA	类型	描述
onRepeatModeChanged	REPEAT_MODE	int	循环模式，0/1/2/3 分别为 关闭/单曲循环/列表循环/随机播放

8.2.18 设置倍速命令的定义

ACTION	DATA	类型	描述	可选/必选
setSpeed	SPEED	float	播放速度，取值 0.25/0.5/0.75/1.0/1.25/1.5/1.75/2.0 之一	必选

8.2.19 设置倍速命令的响应定义

CALLBACK_ACTION	DATA	类型	描述
onPlaySpeedChanged	SPEED	float	播放速度，取值 0.25/0.5/0.75/1.0/1.25/1.5/1.75/2.0 之一

8.2.20 设置音量命令的定义

ACTION	DATA	类型	描述	可选/必选
setVolume	VOLUME	int	音量，取值范围是 0—100	必选

8.2.21 设置静音命令的定义

ACTION	DATA	类型	描述	可选/必选
setMute	MUTE	boolean	是否静音，true 则静音，false 则恢复上一次的音量	必选

8.2.22 设置音量/静音命令的响应定义

CALLBACK_ACTION	DATA	类型	描述
onVolumeChanged	VOLUME	int	音量，取值范围是 0—100，静音则值为 0

8.2.23 设置音频轨道信息命令的定义

ACTION	DATA	类型	描述	可选/必选
setAudioTrackInfo	trackInfo	TrackInfo	音频轨道信息，如立体声、环绕声等。 TrackInfo 定义参考 8.2.23.1	必选

8.2.23.1 轨道信息（TrackInfo）的全部字段信息

DATA	类型	描述
KEY_RENDERER_INDEX	Int	渲染器索引
KEY_GROUP_INDEX	Int	组索引
KEY_TRACK_INDEX	Int	轨迹索引

DATA	类型	描述
KEY_TRACK_NAME	String	轨迹名称
KEY_SELECTED	Boolean	是否被选中
KEY_WIDTH	Int	宽度
KEY_HEIGHT	Int	高度
KEY_BITRATE	Int	比特率
KEY_CHANNEL_COUNT	Int	通道数目
KEY_LANGUAGE	String	语言

8.2.24 设置视频轨信息命令的定义

ACTION	DATA	类型	描述	可选/必选
setVideoTrackInfo	trackInfo	TrackInfo	视频轨道信息，如标清、高清等。TrackInfo 定义参考 8.2.23.1	必选

8.2.25 设置音/视频轨信息命令的响应定义

CALLBACK_ACTION	DATA	类型	描述
onTrackChanged	trackInfoSet	TrackInfoSet	可用的音频轨、视频轨信息实体的集合，可以从中选择进行设置。TrackInfoSet 定义参考 8.2.25.1

8.2.25.1 轨道信息集（TrackInfoSet）的全部字段信息

DATA	类型	描述
KEY_AUDIO_TRACKINFO_LIST	List<TrackInfo>	音频轨列表
KEY_AUDIO_HAS_SELECTION_OVERRIDE	Boolean	是否使用默认音频轨
KEY_IS_AUDIO_SUPPORT_AUTO	Boolean	是否支持自动音频轨
KEY_VIDEO_TRACKINFO_LIST	List<TrackInfo>	视频轨列表
KEY_VIDEO_HAS_SELECTION_OVERRIDE	Boolean	是否使用默认视频轨
KEY_IS_VIDEO_SUPPORT_AUTO	Boolean	是否支持自动视频轨

8.2.26 DRM onKeyRequest 的定义

CALLBACK_ACTION	DATA	类型	描述
onKeyRequest	MEDIA_ID	string	媒体资源 ID，与 PlayInfo 中的 MediaId 一致
	REQUEST_KEY	byte[]	版权信息密钥

8.2.27 DRM provideKeyResponse 的定义

ACTION	DATA	类型	描述	可选/必选
provideKeyResponse	MEDIA_ID	string	媒体资源 ID，与 PlayInfo 中的 MediaId 一致	必选
	RESPONSE_KEY	byte[]	来自许可证服务器响应的版权数据	必选

8.2.28 播放器错误的定义

CALLBACK_ACTION	DATA	类型	描述
onPlayerError	ERROR_CODE	int	错误码

CALLBACK_ACTION	DATA	类型	描述
	ERROR_MSG	string	错误信息

9 规格定义

本章定义接收端播放器需要支持的播放内容格式，涉及视频、音频、图片格式。

9.1.1 视频支持的格式

支持的媒体类型		可选性
视频	MP4	必选
	FMP4	可选
	WebM	必选
	Matroska	必选
	AVI	必选
	MOV	必选
	WMV	必选
	MKV	必选
	VOB	必选
	FLV	必选
	M3U8	必选
	MPEG	必选
	3g2	可选
	3gp	可选
	rm	可选
	RMVB	可选

9.1.2 音频支持的格式

支持的媒体类型		可选性
音频	MP3	必选
	WAV	必选
	M4A	必选
	MIDI	必选
	AIFF/AU	必选
	ADTS（AAC)	必选
	FLAC	必选
	AMR	必选
	Ogg	必选
	WMA	可选
	RA	可选

9.1.3 图片支持的格式

支持的媒体类型		可选性
图片	png	必选
	jpeg	必选
	bmp	必选
	webp	可选
	gif	可选
	svg	可选

9.1.4 流媒体通信协议类型

流媒体通信协议类型		可选性
流媒体类型	HLS	必选
	DASH	必选
	RTP over UDP	可选
	RTP, interleaved RTSP over TCP	可选

9.1.5 DRM 支持的类型

支持的媒体类型		可选性
DRM	CHINADRM	可选
	WIDEVINE	可选
	PLAYREADY	可选
	CLEARKEY	可选

9.1.6 编解码格式

支持的媒体编解码格式		可选性
视频编码格式	H.264	必选
	H.265	必选
	H.266	可选
	VP8	可选
	VP9	可选
	AVS/AVS+	可选
	AVS2	可选
	AVS3	可选