ICS：33.030
CCS：M21

UHD World Association
世界超高清视频产业联盟

世 界 超 高 清 视 频 产 业 联 盟 标 准

T/UWA 024-2023

基于流媒体的多屏互动技术要求

Technical requirements of multiscreen interaction based on streaming media

2023-12-31 发布　　2023-12-31 实施

世界超高清视频产业联盟 发布

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：华为技术有限公司、中央广播电视总台、康佳集团股份有限公司、深圳TCL新技术有限公司、深圳市腾讯计算机系统有限公司、夏普电子研发（南京）有限公司、海信视像科技股份有限公司、北京奇艺世纪科技有限公司、北京字节跳动网络技术有限公司、深圳创维-RGB电子有限公司、优酷信息技术（北京）有限公司、咪咕文化科技有限公司、四川新视创伟超高清科技有限公司、OPPO广东移动通信有限公司、京东方科技集团股份有限公司、杭州当虹科技股份有限公司、烽火通信科技股份有限公司、北京中视广信科技有限公司、深圳市酷开网络科技股份有限公司、深圳市洲明科技股份有限公司、利亚德光电股份有限公司、北京流金岁月传媒科技股份有限公司、西安诺瓦星云科技股份有限公司、上海数字电视国家工程研究中心有限公司。

本文件主要起草人：刁月磊、王浩、黄一宏、高伟、梁刚、马鸿、顾军、聂自非、佘桂海、韦泽垠、刘文富、吴志栩、唐景松、孙吉超、肖成创、赵腾飞、徐遥令、陈灿、郭佩佩、金晶、毕蕾、沈显超、邹双泽、杨益红、王琳、李汤锁、陈明武、张华、裘昊、徐煜烨、孙剑、吴旭、王雷、白莹杰、白建军、刘莉、曾泽君、周凯旋、葛敏锋、殷惠清。

基于流媒体的多屏互动技术要求

1 范围

本文件规定了多个设备接入同一局域网环境时，实现基于流媒体的多屏互动的相关技术要求。

本文件适用于接入同一局域网环境下多个设备实现多屏互动的软件设计、开发和应用。

2 规范性引用文件

本文件没有规范性引用文件。

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

要投播的视频、音乐、图片等媒体类资源文件。

3.2

媒体服务 multimedia Service

提供投播内容的存储和业务鉴权、内容分发管理的服务。

3.3

发送端 source device

多屏互动的发起设备。

3.4

接收端 sink device

多屏互动的接收设备。

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

发送端控制接收端进行内容获取、渲染、播放及控制。

3.7

加密 encipherment;encryption

对数据进行密码变换以产生密文的过程。

[来源：GB/T 36322-2018,3.5]

3.8

解密 decipherment;decryption

与加密过程对应的逆过程。

[来源：GB/T 25069-2022,3.305]

3.9

安全信道 secure channel

为所交换消息提供保密性及真实性的通信信道。

[来源：ISO/IEC 24745:2011,2.30]

4 缩略语

下列缩略语适用于本文件。

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）

ISO 国际标准化组织（International Organization for Standardization）

JSON JS对象简谱（JavaScript Object Notation）

LAN 局域网（Local Area Network）

mDNS 多播域名系统（Multicast DNS）

NearLink 星闪 中国原生的新一代近距离无线联接技术

RTSP 实时流协议（Real Time Streaming Protocol）

SPEKE 简单密钥协商（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）

5 总体框架

5.1 概述

本文件定义了多屏互动的基础架构及交互操作规范，涵盖设备发现、连接认证鉴权、能力协商、媒体播控协议及支持媒体格式规格，实现发送端设备多媒体内容协同到接收端设备播控的功能。

5.2 协议架构

基于流媒体的多屏互动基础架构如图1所示，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服务。

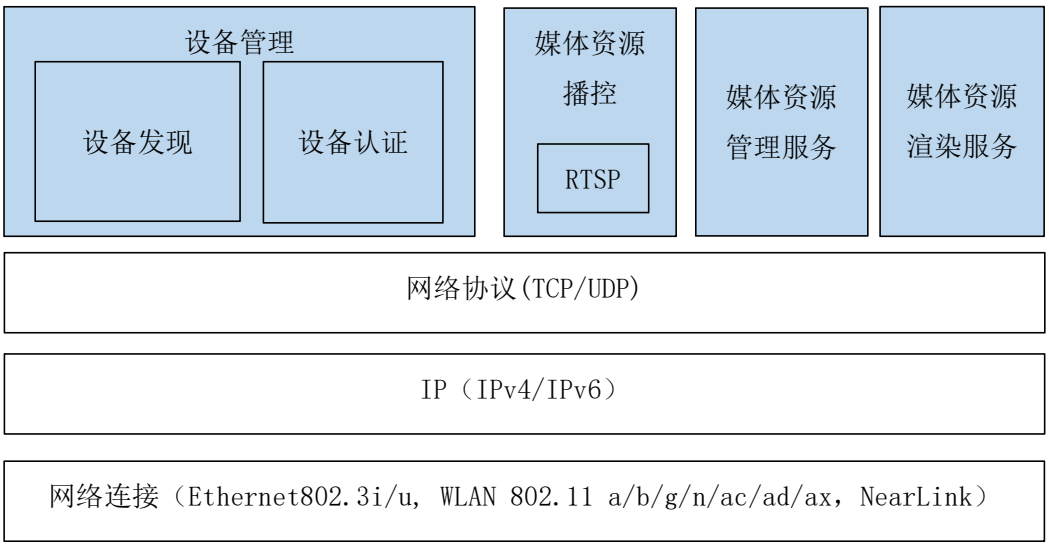

图 1　基于流媒体的多屏互动基础架构

设备管理应由设备发现和设备认证组成。设备发现应支持服务发布、订阅及能力交互。设备认证应支持双端可信认证，密钥生成和管理。

媒体资源播控（控制模块）应支持媒体资源相关的播控命令封装、播放状态等信息的封装和解封装，接收端应具备渲染服务管理能力。

媒体资源管理（媒体服务）应提供内容存储和业务管理系统,宜支持 HTTP、HLS 协议进行资源访问。本文件不详细规定具体行为和协议内容，由内容服务提供者确定。

媒体资源渲染（渲染服务）应支持访问媒体资源管理服务获取内容、根据资源类型加载对应的渲染器，进行渲染和播控。

5.3　模型定义

本文件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

一对一本地资源协同流程如图 2 所示。

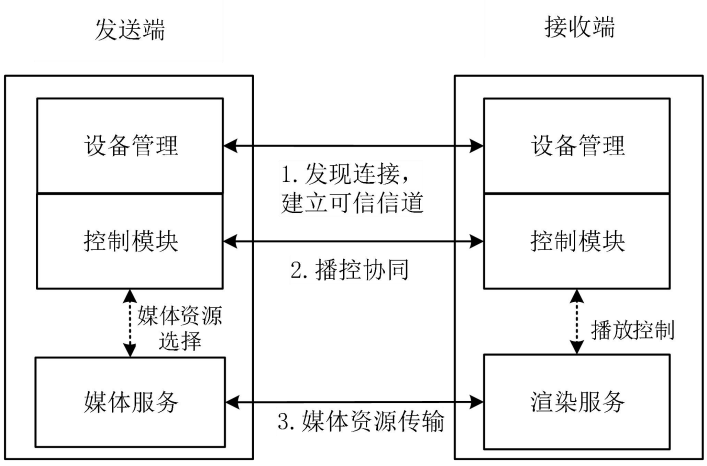

图 2　一对一本地资源协同

发送端应部署设备管理、控制模块和媒体服务，接收端应部署设备管理、控制模块和渲染服务。首先，设备管理发现连接，建立安全信道，再通过控制模块进行播控，最后，渲染服务从媒体服务器中获取指定的媒体资源内容。发送端控制模块与媒体服务之间的资源选择过程本文件不定义，可由用户选择本地某个资源文件等方式实现。接收端的控制模块将播控信息传递给渲染服务，两模块间的信息传递属于内部实现，本文件不定义。

5.4.2 一对一在线资源协同

一对一在线资源协同流程如图 3 所示。

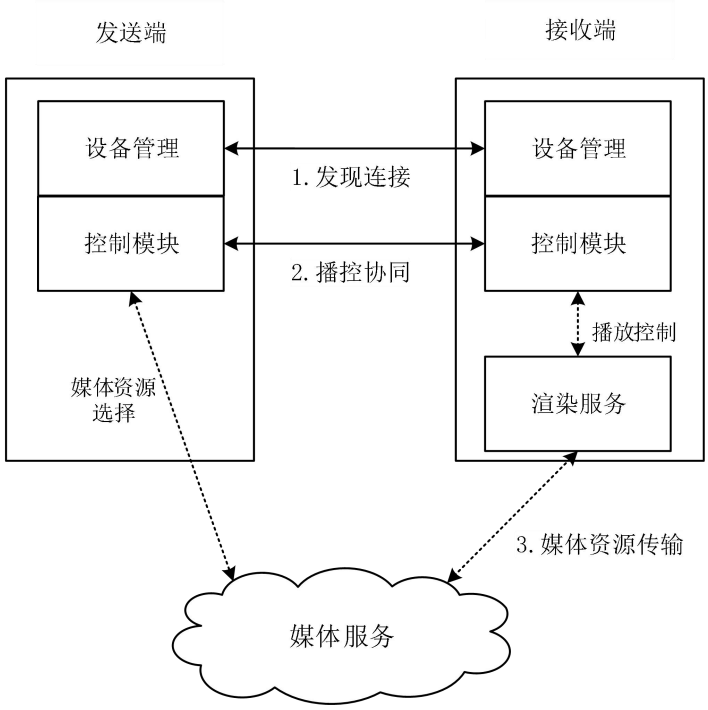

图 3 一对一在线资源协同

发送端应部署设备管理和控制模块，接收端应部署设备管理、控制模块和渲染服务，媒体服务部署在网络可达的设备上。首先，设备管理发现连接，建立安全信道，再通过控制模块进行播控，最后，渲染服务从在线的媒体服务获取媒体资源内容。发送端控制模块与媒体服务之间的资源选择过程本文件不定义，可由发送端内其他模块如视频应用与媒体服务交互后通过应用接口传递给控制模块。

5.4.3 一对多本地资源协同

一对多本地资源协同流程如图 4 所示。

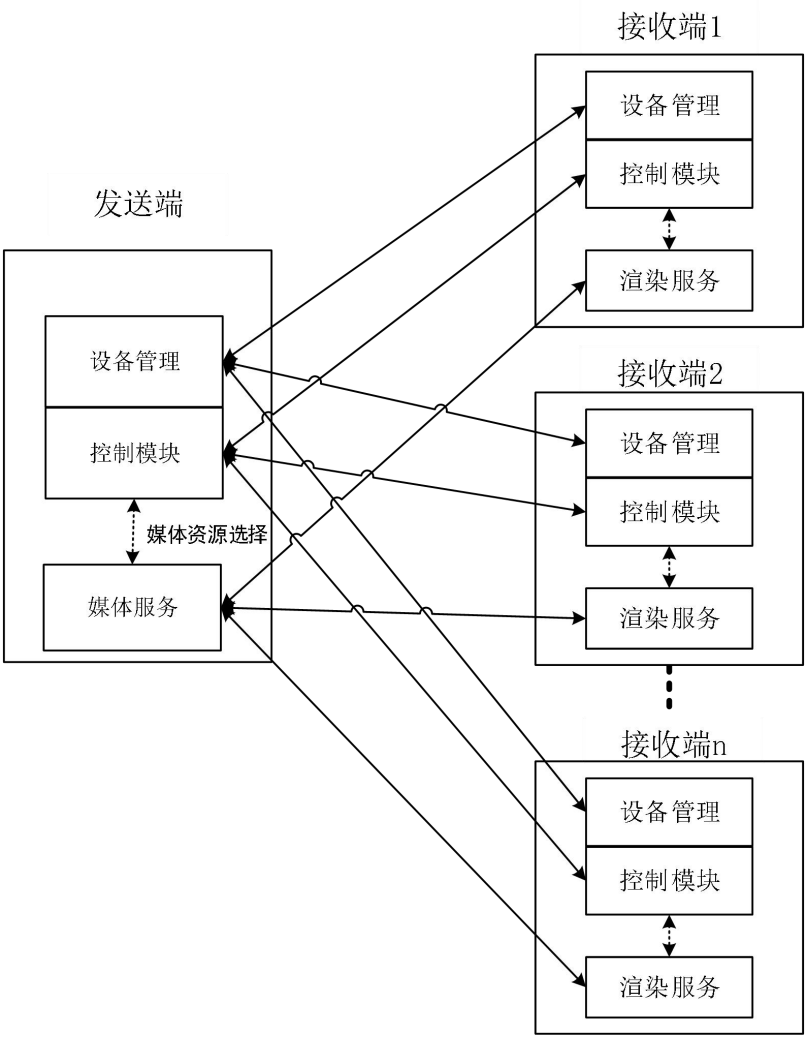

图 4　一对多本地资源协同

发送端的控制模块和媒体服务宜具备启动多个连接的能力。发送端可以推送不同的资源到不同的接收端，本场景中各接收端之间不具有相关性。发送端应部署设备管理、控制模块和媒体服务器，接收端应部署设备管理、控制模块和渲染服务。发送端与每个接收端之间的互动流程与 5.4.1 章节定义一致。

5.4.4　一对多在线资源协同

一对多在线资源协同流程如图 5 所示。

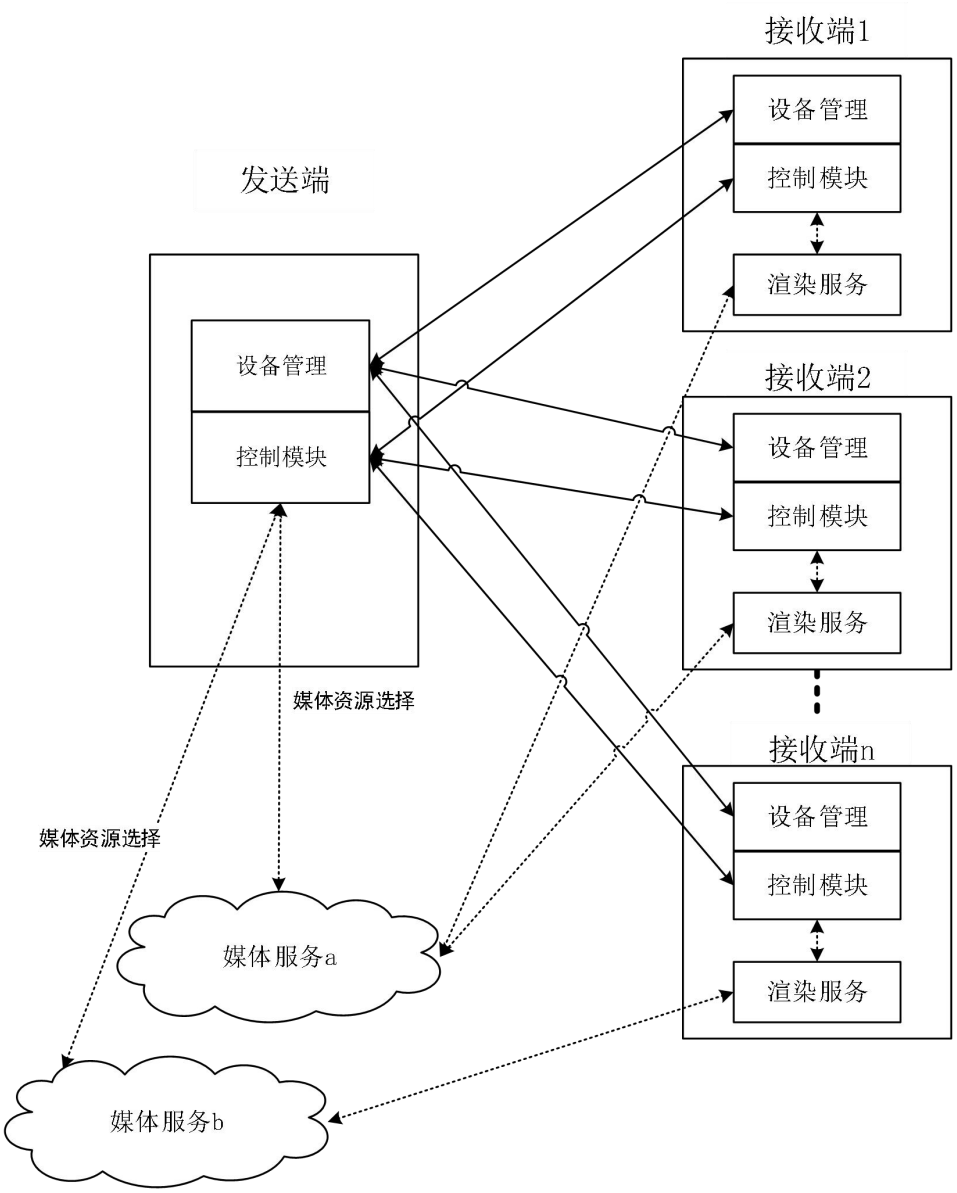

图 5　一对多在线资源协同

接收端控制模块宜具备启动多个连接的能力。发送端可以推送资源到不同的接收端，本场景中各接收端之间不具有相关性。发送端应部署设备管理和控制模块，接收端应部署设备管理、控制模块和渲染服务，媒体服务部署设备为单独另外设备。发送端、媒体服务与每个接收端之间的互动流程与 5.4.2 章节定义一致。

5.5　交互流程

5.5.1　概述

多屏互动业务流程如图 6 所示，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。

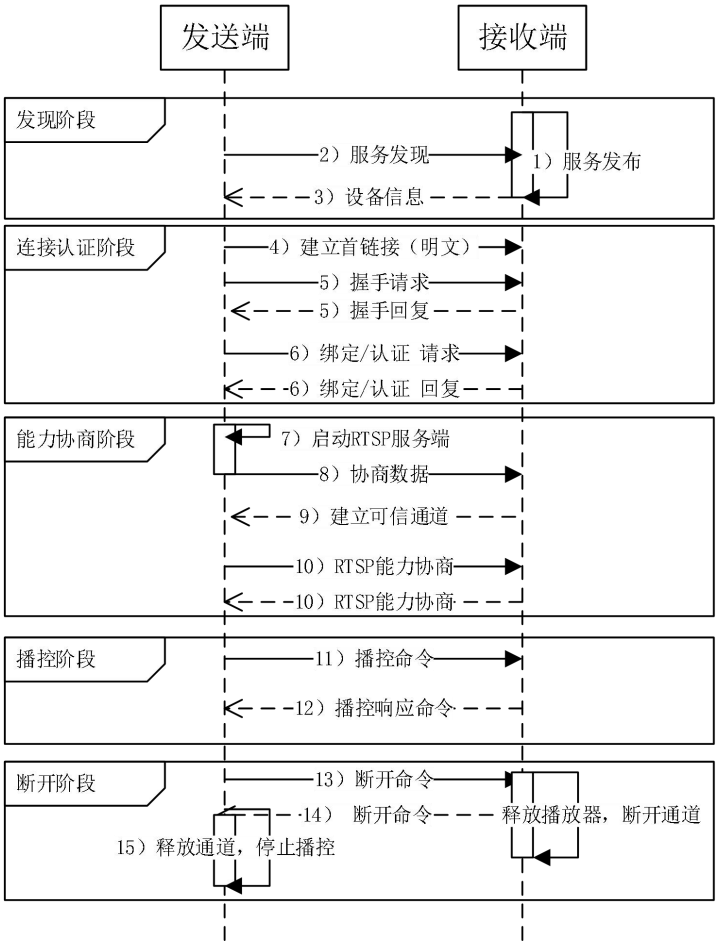

图 6　多屏互动业务流程

各阶段流程说明如下：

发现阶段：

1） 接收端开启设备管理，进行服务发布，等待被发现，宜默认开启；

2） 发送端启动设备管理，进行服务发现，以获取设备列表；

3） 接收端回复设备信息，包含支持的服务、渲染能力以及服务端口等信息；

连接认证阶段：

4）发送端发起连接，通过发现阶段获得的接收端 IP 地址和服务端口建立信道 a（TCP）；

5）发送端发送握手报文，接收端接收到握手报文后结合本端能力进行回复；

6）当有任一端不存在对端信任关系时应进入绑定流程，当双端均存有对端设备的可信关系时应进入认证流程，具体协商可参看 6.2 章节，最终生成会话密钥；

能力协商阶段：

7）发送端启动控制模块，生成 RTSP 服务端口，端口应随机，等待连接；

8）RTSP 服务端口应使用 AES-128-CTR 加密后通过信道 a 发送至接收端；

9）接收端使用 AES-128-CTR 解密获得 RTSP 服务端口，连接到 RTSP 服务，该链接本文件统一称为控制通道；

10）双端通过 RTSP 命令交互进行播放能力协商并业务响应，详细描述见 7.2 章节；

播控阶段：

11）发送端通过控制通道进行播控命令发送；

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

断开阶段：

13）发送端可发送断开命令；

14）接收端接收到断开命令后，应进行渲染服务回收，发送断开命令到发送端，后断开连接；

15）发送端收到断开命令后，应停止本端服务停止，断开与接收端设备的连接。

5.5.2 本地资源推送流程

本地资源推送流程如图 7 所示。

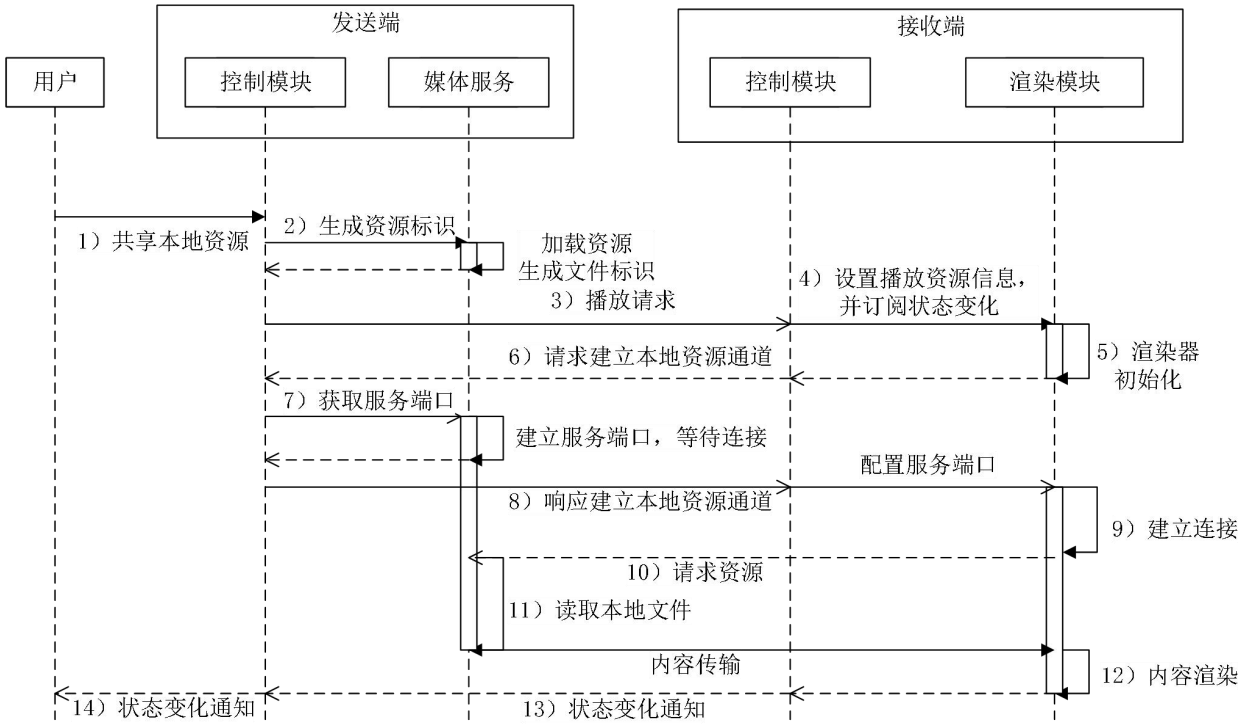

图 7 本地资源推送流程

流程说明如下：

1) 用户在应用中选择待推送的本地资源文件，将文件路径等信息传递给发送端控制模块；

2) 媒体服务应生成文件标识（不应超过 100bytes），并保存文件标识和文件的映射关系；

3) 发送端控制模块封装播放请求命令通过 RTSP 通路传输至接收端控制模块；

4) 接收端控制模块接收到后，解析媒体信息，设置给媒体渲染服务；

5) 接收端媒体渲染服务启动，获取播放内容；

6) 识别为本地内容时，应通过控制模块请求建立本地资源传输通道；

7) 发送端收到请求后，向媒体服务获取 TCP 套接字服务端口；

8) 发送端控制模块在响应中封装服务端口；

9) 接收端控制模块配置服务端口到渲染模块，由渲染模块建立 TCP 连接，本链接应使用 7.2.1 章节协商的流媒体加密算法进行加解密；

10) 接收端通过获取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，通过本地资源传输通道向发送端媒体服务模块请求媒体资源；

11) 发送端媒体服务收到请求后，将请求信息中的文件标识映射成本地文件路径，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，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；

12）接收端接收到 HTTP Response 报文后，进行内容播放；
13）接收端媒体渲染器播放时，应将播放进度及状态等信息按照 8.2 章节生成事件，通过控制模块将事件同步到发送端媒体控制模块；
14）发送端媒体控制模块将事件传递给媒体应用，媒体应用进行相应的响应。

5.5.3 在线资源推送流程

在线资源推送流程如图 8 所示。

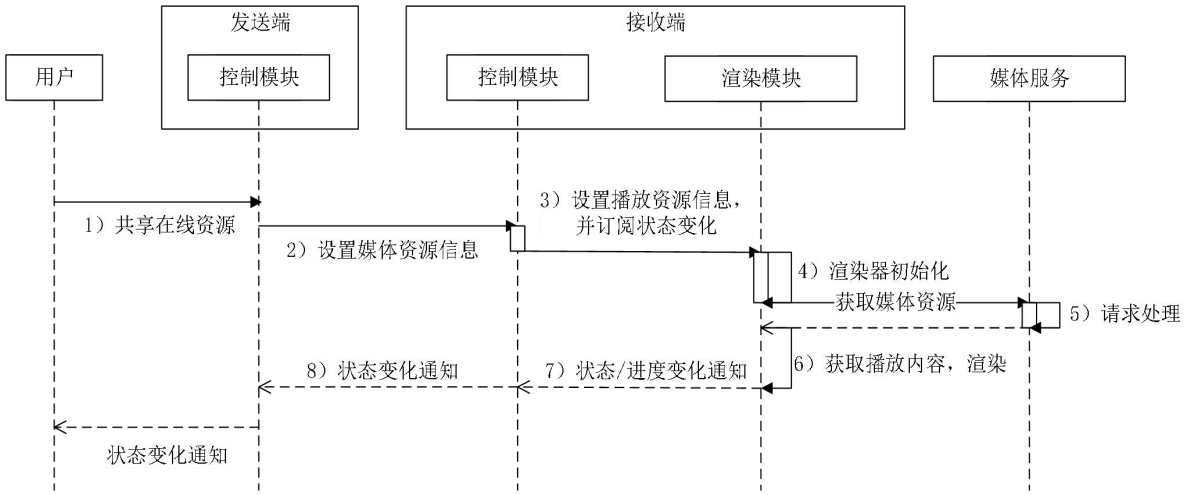

图 8 在线资源推送流程

流程说明如下：

1) 用户在发送端应用中，选择待推送的在线资源，将在线资源及相关信息传递给控制模块；
2) 发送端控制模块将 URL、标题、作者等相关媒体信息封装播放命令发送至接收端控制模块；
3) 接收端控制模块接收播放请求后，解析媒体信息，设置给媒体渲染服务模块；
4) 接收端媒体渲染器播放文件，通过 URL 链接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 中携带读取的播放内容；
6) 接收端媒体渲染服务接收到 Response 报文后，进行内容渲染；
7) 接收端媒体渲染器播放时，应将播放进度及状态等信息按照 8.2 章节生成事件，通过控制模块将事件同步到发送端媒体控制模块；
8) 发送端控制模块将事件传递给媒体应用，媒体应用进行相应的响应。

5.5.4 在线 DRM 资源推送流程

在线 DRM 内容推送流程如图 9 所示。

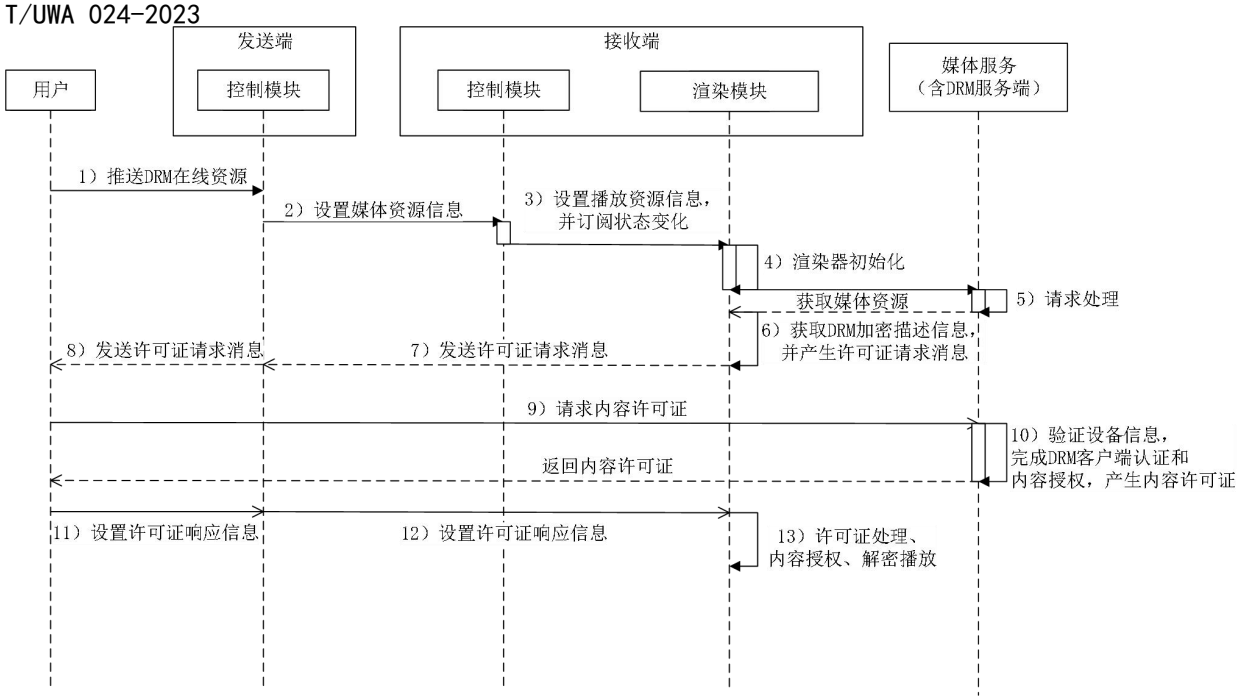

图 9　在线 DRM 资源推送流程

流程说明如下：

1）　用户在发送端应用中，选择待推送的在线 DRM 资源，应用负责和媒体服务交互，完成鉴权，获取内容播放链接等业务鉴权结果，将资源及相关媒体信息传递给发送端控制模块，宜根据获取的接收端设备的 DRM 能力进行推送，如不支持，可停止推送；

2）　发送端控制模块将 URL、标题、作者等相关媒体信息封装播放命令发送至接收端控制模块；

3）　接收端控制模块接收播放请求后，解析媒体信息，设置给媒体渲染服务模块；

4）　接收端媒体渲染器播放文件，通过 URL 链接向网络媒体服务模块请求媒体资源；

5）　媒体服务系统根据播放链接请求，返回请求结果；

6）　接收端解析内容中的 DRM 加密描述信息，发送到控制模块获取许可证请求消息；

7）　接收端将许可证请求消息通过控制模块发送消息到发送端请求许可证；

8）　发送端将许可证请求发送到用户使用的应用。

9）　应用向媒体服务发起内容许可证请求；

10）　媒体服务 DRM 服务端验证许可证请求消息，完成 DRM 客户端认证和内容授权，产生许可证响应消息，并返回内容许可证；

11）　应用发送许可证响应消息到发送端控制模块；

12）　发送端控制模块发送许可证响应信息到接收端；

13）　接收端渲染服务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 6)-13)更新许可证。

5.5.5　播控操作流程

播控操作流程如图 10 所示。

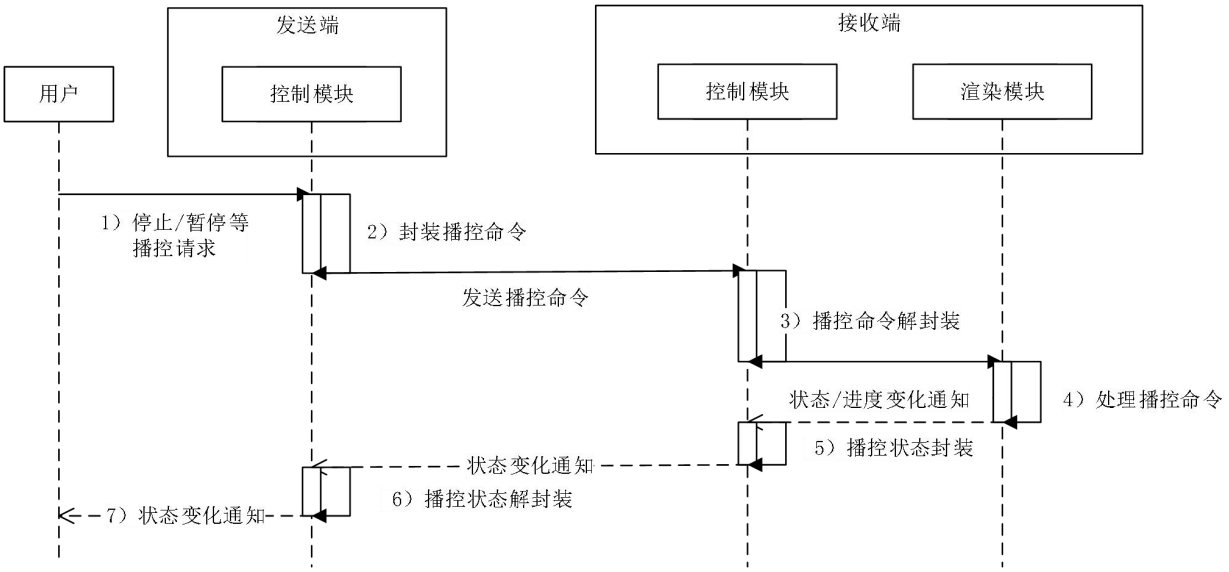

图 10　播控操作流程

流程说明如下：

1) 用户在发送端应用中，将要或者正在投播的资源进行播放控制请求发送给控制模块；
2) 发送端控制模块将用户操作按照 8.2 章节定义封装成 RTSP 报文，发送至接收端控制模块；
3) 接收端控制模块接收到报文后，解析操作命令，校验合法性通过后，设置给媒体渲染服务模块；
4) 接收端渲染服务模块应根据播控指令进行响应，将指令的处理结果和渲染器状态通知到控制模块；
5) 接收端控制模块应按照 8.2 章节定义的响应生成回调命令封装成 RTSP 报文，发送至发送端控制模块；
6) 发送端控制模块接收到状态变化通知后，校验回调行为事件合法性，更新发送端控制模块状态；
7) 发送端控制模块将事件传递给媒体应用，媒体应用进行相应的响应。

6 设备管理

6.1　设备发现

设备发现应遵循 mDNS（rfc6762）和 DNS-SD（rfc6763）协议，本节规定了发现过程中协议配置。

6.1.1　设备发现报文定义

发现报文基础配置（DNS-SD 中 PTR 记录和 SRV 记录）应按照表 1 的规定。

表 1 发现报文基础配置

名称	类型	描述	可选/必选
服务名称	string	厂商定义，发送端可以显示为设备名称，应采用 UTF-8 编码方式，长度不应超过 32bytes	必选
服务类型	string	_cast-remote._tcp.local	必选
端口	Int32	接收端发布服务前应通过启动套接字服务端生成随机端口并等待连接	必选

DNS-SD 中 TXT 记录为本标准定义详细能力交互部分，以键值对格式填充数据，宜采用 JSON 格式，见表 2 定义。

表 2 TXT record 字段配置

关键字	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，宜落盘存储，最小长度不应低于 32bytes，最大长度不应超过 64 bytes	必选
DeviceType	Int32	设备类型，详细定义应按照表 3 的规定	必选
Features	Int32	特性,以 bit 进行定义的，详细定义应按照表 4 的规定	必选
ExtraInfo	String	设备扩展信息，详细定义见表 5	可选

表 3 设备类型定义

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、智能电视
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
电子标牌	0x000D	电子标牌
电子墨水屏	0x000E	电子墨水屏
智能头戴式设备	0x000F	智能头戴设备 AR、VR、MR
3D 显示设备	0x0010	3D 显示设备

表 4 基础特性字段的定义

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	是否可访问网络 0：不支持 1：支持
7	Screen_3D	是否显示设备支持裸眼 3D，0：不支持 1：支持
8~31	Reserved	保留，默认为 0

表 5 扩展信息定义

关键字	值类型	描述	可选/必选
DeviceSubType	String	设备子类型，由厂商随产品定义，最大长度不应超过 64bytes	可选
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。9.7 章节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
ScreenFPS	Int32	屏幕最高刷新率，如：60， 90 等 fps	可选
AudioEffect	Int32	0x0001-Audio Vivid 0x0002-DOLBY 杜比音效 0x0004-HISTEN 音效	可选
HDRCap	Int32	0x0001-HDR Vivid 0x0002-HDR10 0x0004-HDR10+ 0x0008-DOLBY VISION	可选
Multilink	bool	支持多链接无缝衔接播放能力，可以优化支持广告播放，支持：true；不支持：false	可选
MultiHttpDownload	bool	支持 HTTP 多线程下载的能力，可以优化播放器起播耗时，支持：true；不支持：false	可选
DRMCap	String	DRM 能力，应按照表 6 规定以键值对方式填充，宜采用 JSON 格式。	可选

表 6 DRM 能力定义

关键字	值类型	描述
drmSchema	Int32	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互动连接认证鉴权主体。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。

绑定流程应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。

认证流程应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，用户宜在双端设备上无确认操作。

连接鉴权流程如图 11 所示。

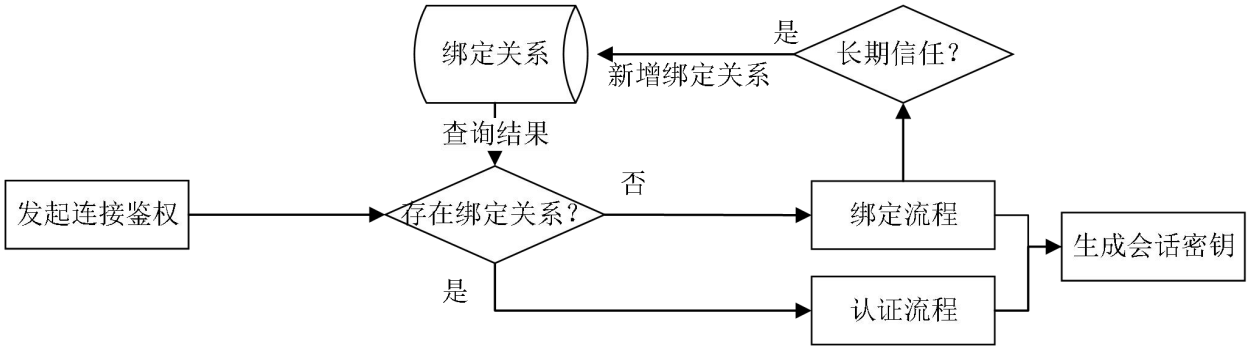

图 11　设备可信关系与连接鉴权流程

6.2.2　设备绑定流程

首次认证的双端设备，应进行设备绑定流程。采用标准 SPEKE 协议[ISO/IEC 11770-4:2017]。其中，认证码宜由发送端或接收端采用安全的随机数生成 6 位数字，并通过带外传输至对端设备。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中的设备唯一标识与公钥进行绑定。SPEKE 协议应由 TCP 报文承载，所有报文应按照 JSON 数据格式解析和生成，具体流程如图 12 所示：

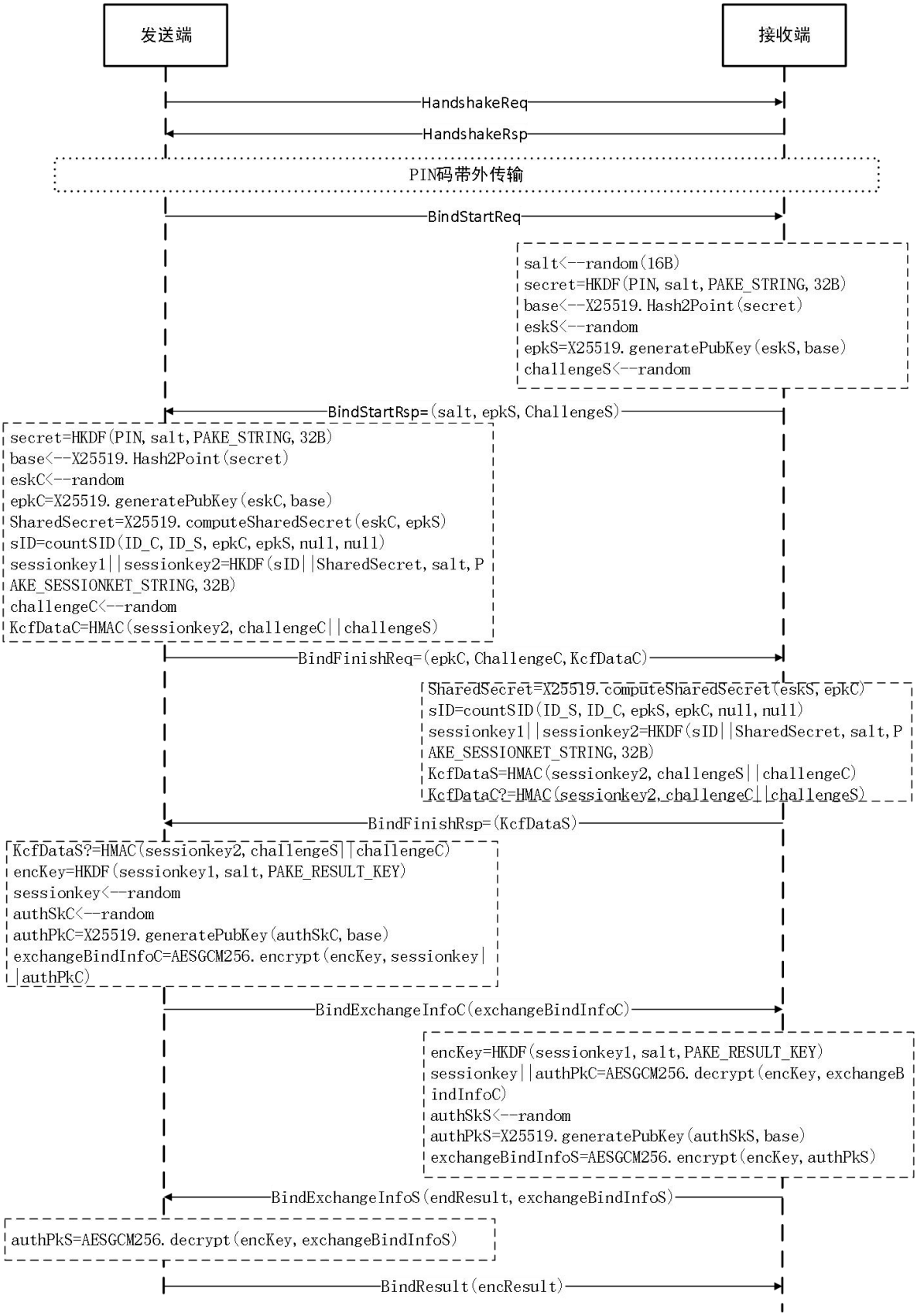

图 12　绑定协议交互

1、 HandshakeReq：握手请求消息

握手请求消息 JSON 对象信息如表 7 所示。

表 7 握手请求消息参数

关键字	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，最大长度不应超过 64bytes
deviceName	String	长度不应超过 32bytes
sequenceNumber	Int32	命令的序列号，可随机生成
isGenericTrusted	Bool	绑定关系库中是否存在接收端通用模式的绑定关系，存在：true，不存在：false
isPwdTrusted	Bool	绑定关系库中是否存在接收端密码模式的绑定关系，存在：true，不存在：false
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：握手响应消息

握手响应消息 JSON 对象信息如表 8 所示。

表 8 握手请求回复消息参数

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 说明接收端就绪状态，可以发起后续流程。 DEVICE_BUSY = 4 说明接收端正在忙碌，发送端应终止连接,宜提示用户接收端正在忙。 HANDSHAKE_FAILED = 255 说明接收端拒绝连接，握手失败，发送端应终止连接。
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	bool	HandShakeReq 中为 true 且本端绑定关系库中存在发送端通用模式的绑定关系时为 true，否则为 false
isPwdTrusted	bool	HandShakeReq 中为 true 且本端绑定关系库中存在发送端密码模式的绑定关系时为 true，否则为 false
allowedAlways	bool	保留可信凭证，由业务或者用户参与后确定，false：不保留凭证，仅本次可信；true：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

通用模式是指连接时认证码显示在接收端，密码模式是指认证码需要预先配置在系统中，连接时不显示。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

绑定开始请求消息 JSON 对象信息如表 9 所示。

表 9 绑定流程开始-请求消息参数

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

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。

sID = countSID(ID_C, ID_S, epkC, epkS, DF1, DF2) 计算方法定义：

输入：ID_C（发送端设备 ID），ID_S（接收端设备 ID），espkC(发送端公钥) ， espkS(接收端公钥)，DF1 (派生因子 1，此处为空)， DF2 (派生因子 2，此处为空)。

返回：sID

计算方法：

a) 记 X_C 为 espkC 的 x 坐标， X_S 为 espkS 的 x 坐标；

b) 计算 S_1= hash(ID_C || X_C || DF1)，并转成整型数；

c) 计算 S_2= hash(ID_S || X_S || DF2)，并转成整型数；

d) 计算 sID：MAX(S_1, S_2)||MIN(S_1, S_2) 。

绑定开始回复消息 JSON 对象信息如表 10 所示。

表 10 绑定流程开始-回复消息参数

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。

5、 BindFinishReq：绑定流程结束-请求消息

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。

sID = countSID(ID_S, ID_A, epkS, epkC, DF1, DF2) 计算方法定义：

输入：ID_C（发送端设备 ID），ID_S（接收端设备 ID），espkC(发送端公钥) ，espkS(接收端公钥)，DF1 (派生因子 1，此处为空)， DF2 (派生因子 2，此处为空)

返回：sessionID

a) 记 X_C 为 espkC 的 x 坐标， X_S 为 espkS 的 x 坐标；

b) 计算 S_1= hash(ID_S || X_S || DF1)，并转成整型数；

c) 计算 S_2= hash(ID_C || X_C || DF2)，并转成整型数；

d) 计算 sessionID：SID = MAX(S_1, S_2)||MIN(S_1, S_2)；

此处 hash 选用 SHA-256。

绑定结束请求消息 JSON 对象信息如表 11 所示。

表 11 绑定流程结束-请求消息参数

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

表 11 （续）

名称	值类型	描述
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 HMAC 计算结果。

6、 BindFinishRsp：绑定流程结束-回复消息

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，并应等于对端传来的 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。

绑定结束回复消息 JSON 对象信息如表 12 所示。

表 12 绑定流程结束-回复消息参数

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 HMAC 计算结果

7、 BindExchangeInfoC：交换密钥-请求消息

发收端根据 Sessionkey2 计算 challengeS||challengeC 的 HMAC 值，即 KcfDataS，并比应等于对端传来的 KcfDataS，否则鉴别失败，结束流程。

发送端根据 Sessionkey1 派生出 32Bytes 加密密钥 encKey。

发送端生成 16 字节随机数作为业务加密密钥 Sessionkey，并采用 encKey 加密 Sessionkey 得到 encSessionKey，加密模式为 AES-256-GCM，IV 值为随机生成的 16 字节。

若为长期信任关系，则发送端生成公私钥对(authSkC,authPkC)，authPkc 与 SessionKey 拼接位 32 字节数据后，再进行加密处理。

交换秘钥请求消息 JSON 对象信息如表 13 所示。

表 13 交换密钥-请求消息参数

名称	值类型	描述
Version	String	支持协议版本号， 当前为 1.0
OperType	Int32	4，代表交换密钥命令
exchangeBindInfoC	Byte[64]	Sessionkey、authPkC 数据加密和加密使用的 IV

8、 BindExchangeInfoS：交换密钥-回复消息

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。

若为长期信任关系，解密 sessionKey 时可获得发送端 authPkC，则接收端生成公私钥对(authSkS, authPkS)，并采用 encKey 加密 authPkS 得到 encPkS，加密模式为 AES-256-GCM，IV 值为随机生成的 16 字节。

交换秘钥回复消息 JSON 对象信息如表 14 所示。

表 14 交换密钥-回复消息参数

名称	值类型	描述
Version	String	支持协议版本号， 当前为 1.0
OperType	Int32	4，代表交换密钥命令
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
exchangeBindInfoS	Byte[64]	authPkS 解密数据+16 字节 IV

9、 ExchangeBindFinish：绑定确认消息

绑定确认消息 JSON 对象信息如表 15 所示。

表 15 交换密钥-回复消息参数

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 并将接收端 authPkS 与接收端 ID_S 绑定，并将 (versionIndex，AuthPkS，ID_S，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，ID_C，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 参数，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。具体流程如图 13 所示。

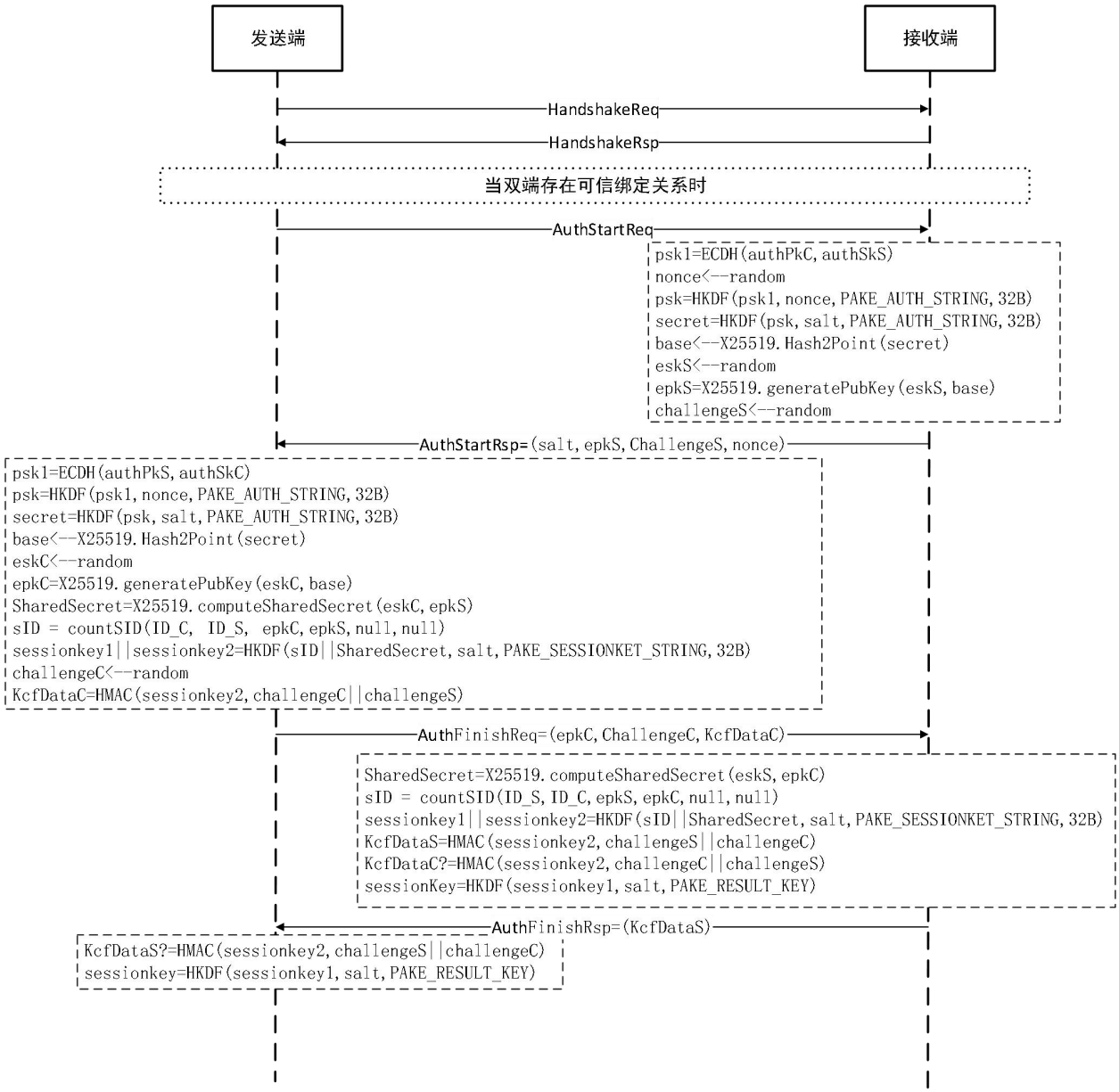

图 13　认证协议交互

1、 AuthStartReq：认证流程开始-请求消息

认证开始请求消息 JSON 对象信息如表 16 所示。

表 16 认证流程开始-请求消息参数

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 0x01，即默认值，则进行以下交互。

2、 AuthStartRsp：认证流程开始-回复消息

认证开始回复消息 JSON 对象信息如表 17 所示。

表 17 认证流程开始-回复消息参数

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

认证结束请求消息 JSON 对象信息如表 18 所示。

表 18 认证流程结束-请求消息参数

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

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

认证结束回复消息 JSON 对象信息如表 19 所示。

表 19 认证流程结束-回复消息参数

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可使用如下验证算法如下。

a） HKDF 算法

密钥推导算法，选择 SHA-256 作为 Hash 函数。

b） HMAC 算法

HMAC（散列消息鉴别码）中，选择 SHA-256 作为 Hash 函数。

c） 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。

d） Hash2Point 算法

选择 Elligator2 方案作为 Hash2Point 算法。

注：可参考 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。

e） 绑定流程使用的对称加密

AES-256 算法 GCM 模式。

验证算法应标注版本号，基础版本为 V1.0。

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，应支持基础版本。

7 能力协商和播控命令定义

7.1 概述

发送端和接收端控制模块所有的交互信令格式在 RTSP 规范（rfc2326）的基础上，定义了扩展的方式。本章规定涉及 RTSP 协议方法及详细定义。

本文件使用 ANNOUNCE、OPTIONS、SET_PARAMETER、GET_PARAMETER、TEARDOWN 方法，进行状态、播控及响应信令交互。消息中*斜体*描述部分应根据实际上下文进行配置。

7.1.1 ANNOUNCE

该方法用于协商多屏互动时双端加密算法，接收端连接发送端控制通道建立成功后，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=*加密算法集*

加密算法集定义应按照表 20 的规定，根据系统支持加密能力选择。

表 20 加密算法集参数

值定义	描述
aes128ctr, aes128gcm	宜支持，控制 RTSP 信令应选 AES-128-GCM 加密，视频内容应选 AES-128-CTR 加密。
aes128ctr	控制 RTSP 信令和视频内容均使用 AES-128-CTR 加密算法。

3） 响应定义：

同 RTSP1.0,本文件不做要求。

7.1.2 OPTIONS

该方法用于请求获取对端支持的方法。

7.1.3 GET_PARAMETER

该方法用于请求获取接收端播放能力、播放质量。空消息体的 GET_PARAMETER 可用于探测对端是否在线。

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：

请求的消息

3） 响应定义：

同 RTSP1.0,携带相应参数值。

请求的消息见表 21。

表 21 消息体定义

值定义	描述
his_player_controller_capability	播控能力交互，详细参数定义见 7.3 节
his_player_qoe	播放质量交付，详细参数定义见 7.4 节

7.1.4 SET_PARAMETER

该方法用来设置到对端的参数值。本规范通过该方法实现控制模块的播控操作。具体播控命令携带在消息体中，见 8.2 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：

请求的消息

3）响应定义：

同 RTSP1.0,本规范不做要求。

示例可参考附录 A，参数值按需携带在消息体中，具体定义如表 22 所示。

表 22 SET_PARAMETER 参数值定义

值定义	描述	取值	必选/可选
his_version	协议版本号	1.0	可选
his_execute_method	触发方法执行	SETUP - 控制模块启动播控 RENDER_READY - 渲染服务模块通知播放就绪 SEND_EVENT_CHANGE – 控制模块进行播放控制	可选
module_id	事件分发的模块，应用于 SEND_EVENT_CHANGE 方法	播控模块标识符 1009	可选
event	需要分发的事件，应用于 SEND_EVENT_CHANGE 方法	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 本地资源流通道端口号 104	可选
param	分发的事件详细内容，应用于 SEND_EVENT_CHANGE 方法	应采用 JSON 格式，具体的赋值可参考 8.2 节播控的命令定义	可选

a) 启动命令报文定义：

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

his_execute_method: SETUP

b) 渲染服务就绪报文定义：

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: RENDER_READY

c) 播控命令报文定义：

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: 1009(媒体播控模块标识符)
event: *action*
param: *param*

7.1.5 TEARDOWN

断开连接。请求会停止业务，释放相关资源。
1）报文定义：
TEARDOWN rtsp://localhost/hisight1.1 RTSP/1.0
CSeq: *seq*
2）消息体定义：
无
3）响应定义：
同 RTSP1.0,本规范不做要求。

7.2 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。详细的消息定义见表 23。

表 23 RTSP 消息定义

阶段	发起者	数据体	行为	描述
Announce1	接收端	7.1.1	7.2.1	请求加密算法列表
Announce2	发送端	7.1.1	7.2.1	返回协商后加密算法列表
M1	发送端	7.1.2	7.2.2	宣称本端支持方法(可选)
M2	接收端	7.1.2	7.2.2	宣称本端支持方法(可选)
M3	发送端	7.1.3	7.2.2	请求对端参数
M4	发送端	7.1.4	7.2.2	发送端发送经过协商后的参数
M5	发送端	7.1.4	7.2.3	会话建立

7.2.1 协商加密算法阶段

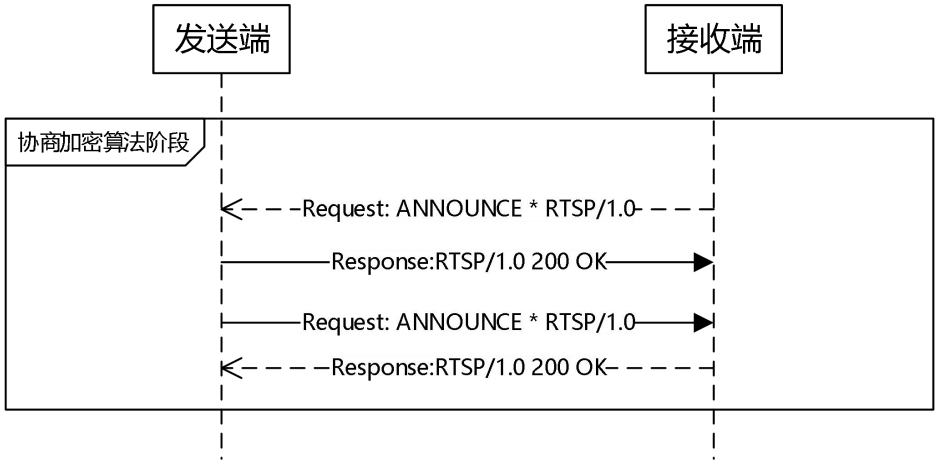

图 14　协商加密算法阶段

如图 14 所示算法协商流程，明文交互，确定算法后，后续所有报文应使用协商算法加密传输。

- Announce 1 阶段：

接收端主动发起，携带本端加密算法列表发送至对端，当前版本宜支持 AES-CTR-128 和 AES-GCM-128，至少支持 AES-CTR-128；

- Announce 2 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

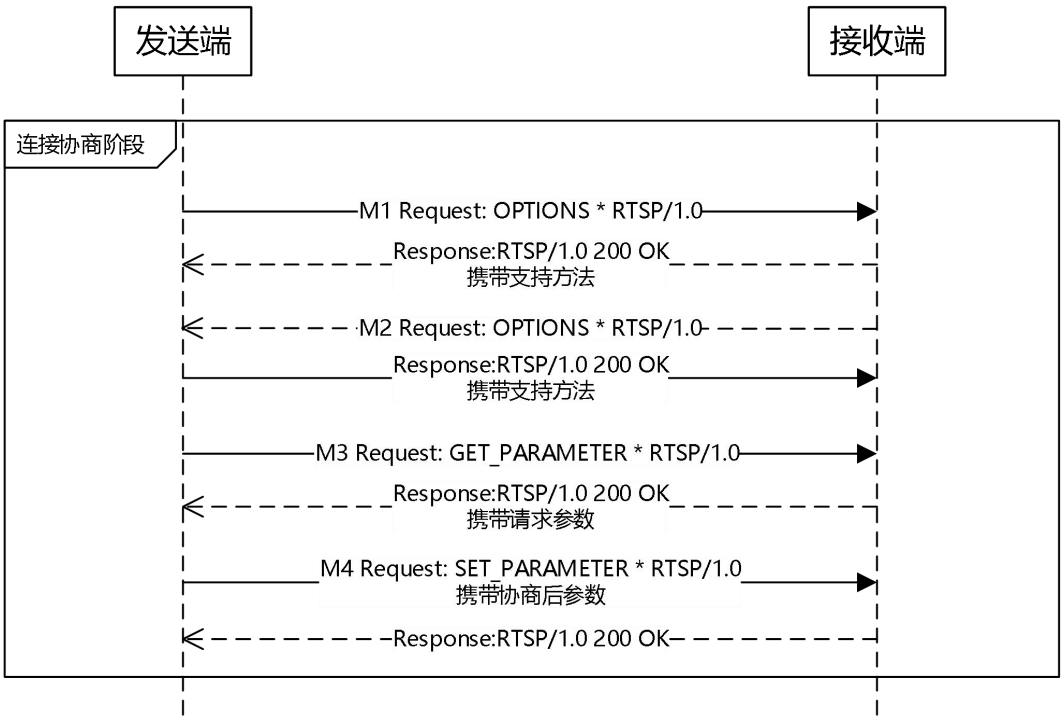

图 15　连接协商阶段

如图15所示连接协商流程，说明如下：

- M1&M2 阶段：

该阶段双端皆可发起，请求对端设备支持的方法；至少应支持“ANNOUNCE，OPTIONS，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，具体播放器参数参考 7.4 节；

- 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

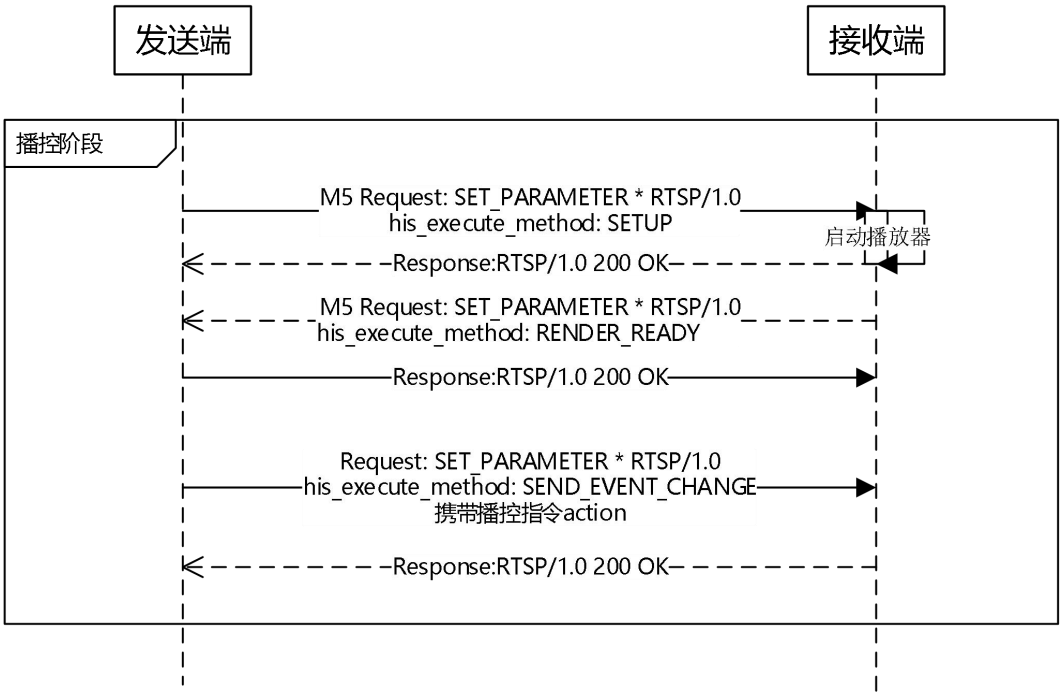

图 16 播控阶段

如图16所示，播放启动及播控命令交互。

M5 阶段：

发送端发起，发送 SETUP 命令，接收端接收端命令后，应启动接收端渲染服务，确保服务运行正常后，回复 RENDER_READY。接收端接收到命令后可发起播控命令进行资源推送和播控。

7.2.4 断开连接

发送端和接收端均可以主动发送 RTSP TEARDOWN 断开命令，发送后，可等待一段时间（应不超过5s），若未等到回复，可直接停止本端服务，拆除 RTSP 链路，恢复到未连接状态。

在接收端正常播放状态，未收到发送端断开命令，发现与发送端设备断开链接，宜继续播放。

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，应以 JSON 格式封装传递。具体播放能

力参数应按照表 24 的规定。

表 24 设备播放能力参数

关键字	值类型	描述	必选/可选
SUPPORT_4K	bool	接收端是否支持 4K 解码播放及渲染能力，支持：true；不支持：false	可选
SUPPORT_8K	bool	接收端是否支持 8K 解码播放及渲染能力，支持：true；不支持：false	可选
MEDIA_VOLUME	Int32	接收端初始音量，0~100	必选
DRM_CAPABILITY_PROPERTIES	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	必选
DECODE_CAPABILITY	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。9.7 章节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
SOUND_EFFECT	Int32	0x0001-Audio Vivid 0x0002-DOLBY 杜比音效 0x0004-HISTEN 音效	可选

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、重新缓冲等因素，播放质量参数应按照表 25 的规定。

表 25 播放质量参数

关键字	值类型	描述	必选/可选
PLAY_SUCCESS	bool	接收端是否播放成功，支持：true；不支持：false	必选
START_PLAY_TIME	Int32	起播时长，内容播放从接收播放请求到链接到首帧播放耗时。单位：毫秒	可选
CACHE_TIME	Int32	重新缓冲耗时。单位：毫秒	可选

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，并通过可信通道发送给接收端控制模块，并由其下发给媒体渲染服务进行响应，媒体渲染服务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 播控操作及响应定义

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。具体的使用可参考 7.1.4 章节。

8.2.1 播放命令定义

播放命令定义应按照表 26 的规定。

表 26 播放命令定义

ACTION	DATA	类型	描述	可选/必选
play	CURRENT_INDEX	Int32	当前音频/视频/图片资源在列表中的索引	必选
	PROGRESS_INTERVAL	Int32	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。应不小于 30 秒，不大于 60 秒，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 应按照表 27 的规定	必选

表 27 播放信息的全部字段信息定义

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	Int32	媒体资源开始播放的起始位置
KEY_DURATION	Int32	媒体资源总时长
KEY_CLOSING_CREDITS_POSITION	Int32	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

表27 （续）

字段	类型	描述
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	Int32	DRM证书类型，比如Widevine DRM、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 播控命令的响应定义

8.2.2.1 播放媒体信息回调

接收端渲染服务在开始播放任一资源时，应发送媒体信息到发送端，发送命令信息应按照表 28 规的规定。

表 28 播放媒体资源信息回调参数

CALLBACK_ACTION	DATA	类型	描述
onMediaItemChanged	playInfo	PlayInfo	返回当前播放的媒体资源，内容信息应按照表 29 的规定，播放资源存在相关信息时应尽量填充。

表 29 onMedialtemChanged 返回的 PlayInfo 参数

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

8.2.2.2 播放状态信息回调

接收端渲染器播放状态变化时，应发送播放状态到发送端，进行播放状态同步，发送命令信息应按照表 30 的规定。

表 30 播放状态回调参数

CALLBACK_ACTION	DATA	类型	描述
onPlayerStatusChanged	PLAYBACK_STATE	Int32	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	bool	准备就绪状态下，应配置播放状态，是否正在播放中，false：表示暂停，true：表示播放中

8.2.2.3 播放位置信息回调

接收端渲染器播放中时应把播放位置信息同步到发送端设备，应符合如下规则：

1） 接收端启动播放时应发送位置同步信息到发送端；

2） 接收端播放中应周期性同步播放位置到发送端，该周期应不小于 30s，不大于 60s。在周期内发送端宜自行计时进行进度更新，发送端回复主要用于校准进度；

3） 接收端因本端播放控制（如：快进、快退等）导致播放位置变化时，应发送播放位置信息；

4） 接收端发生缓存慢、倍速等导致播放进度不规律时，可每秒同步一次播放进度；

5） 发送端设备发送请求位置信息命令时，接收端应立刻回复该信息。

发送播放位置信息应按照表 31 的规定。

表 31 播放位置回调参数

CALLBACK_ACTION	DATA	类型	描述
onPositionChanged	POSITION	Int32	播放位置，单位毫秒
	BUFFER_POSITION	Int32	缓冲位置，单位毫秒
	DURATION	Int32	媒体资源总时长，单位毫秒

8.2.3 暂停命令定义

发送端对接收端的播控命令，接收端响应后应遵照 8.2 章节回复响应回调。暂停命令定义应按照表 32 的规定。

表 32 暂停命令参数

ACTION	DATA	类型	描述
Pause	NA	NA	仅填充 action

8.2.4 继续命令定义

发送端对接收端的播控命令，接收端响应后应遵照 8.2 章节回复响应回调。继续命令定义应按照表 33 的规定。

表 33 继续命令参数

ACTION	DATA	类型	描述
Resume	NA	NA	仅填充 action

8.2.5 停止命令定义

发送端对接收端的播控命令，接收端响应后应遵照 8.2 章节回复响应回调。停止命令定义应按照表 34 的定义。

表 34 停止命令参数

ACTION	DATA	类型	描述
Stop	NA	NA	仅填充 action

8.2.6 新增媒体项命令定义

发送端对接收端的新增媒体信息用于播放中增加新的播放信息放入接收端播放列表。新增媒体项命令定义应按照表 35 的规定。

表 35 增媒体项命令参数

ACTION	DATA	类型	描述
addMediaItem	playInfo	PlayInfo	需要新增的媒体资源，定义应按照表 27 的定义

注意：输入的 PlayInfo 应包含全部的字段数据

8.2.7 播放下一条命令的定义

发送端对接收端的播控命令，接收端响应后应遵照 8.2 章节回复响应回调。命令定义应按照表 36 的规定。

表 36 播放下一条命令参数

ACTION	DATA	类型	描述
next	NA	NA	仅填充 action

8.2.8 播放前一条命令的定义

发送端对接收端的播控命令，接收端响应后应遵照 8.2 章节回复响应回调。命令定义应按照表 37 的规定。

表 37 播放上一条命令参数

ACTION	DATA	类型	描述
previous	NA	NA	仅填充 action

8.2.9 指定位置播放命令的定义

发送端对接收端的播控命令，接收端响应后应遵照 8.2 章节回复响应回调。命令定义应按照表 38 的规定。

表 38 指定位置播放命令参数

ACTION	DATA	类型	描述
seek	POSITION	Int32	指定的播放位置，单位毫秒

8.2.10 快进命令的定义

发送端对接收端的播控命令，接收端响应后应遵照 8.2 章节回复响应回调。命令定义应按照表 39 的规定。

表 39 快进命令参数

ACTION	DATA	类型	描述
fastForward	DELTA	Int32	快进时长，单位毫秒

8.2.11 快退命令的定义

发送端对接收端的播控命令，接收端响应后应遵照 8.2 章节回复响应回调。命令定义应按照表 40 的规定。

表 40 快退命令参数

ACTION	DATA	类型	描述
fastRewind	DELTA	Int32	快退时长，单位毫秒

8.2.12 设置重播模式命令的定义

发送端对接收端的播控命令。命令定义应按照表 41 的规定。

表 41 重播模式命令参数

ACTION	DATA	类型	描述
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

8.2.13 设置重播模式命令的响应定义

接收端重播模式变化，应发送重播模式响应命令到发送端。命令定义应按照表 42 的规定。

表 42 重播模式响应命令参数

CALLBACK_ACTION	DATA	类型	描述
onRepeatModeChanged	REPEAT_MODE	Int32	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

8.2.14 设置倍速命令的定义

发送端对接收端的播控命令。命令定义应按照表 43 的规定。

表 43 倍速命令参数

ACTION	DATA	类型	描述
setSpeed	SPEED	float	播放速度，可取值 0.25/0.5/0.75/1.0/1.25/1.5/1.75/2.0/3.0，最高不宜超过 4.0

8.2.15 设置倍速命令的响应定义

接收端倍速变化后，应发送倍速响应命令到发送端。命令定应按照表 44 的规定。

表 44 倍速响应命令参数

CALLBACK_ACTION	DATA	类型	描述
onPlaySpeedChanged	SPEED	float	播放速度，可取值 0.25/0.5/0.75/1.0/1.25/1.5/1.75/2.0/4.0，最高不宜超过 4.0

8.2.16 设置音量命令的定义

发送端对接收端的播控命令。命令定义应按照表 45 的规定。

表 45 设置音量命令参数

ACTION	DATA	类型	描述
setVolume	VOLUME	Int32	音量，取值范围是 0—100

8.2.17 设置静音命令的定义

发送端对接收端的播控命令。命令定义应按照表 46 的规定。

表 46 设置静音命令参数

ACTION	DATA	类型	描述
setMute	MUTE	Int32	是否静音，1：静音，0：恢复上一次的音量

8.2.18 设置音量/静音命令的响应定义

接收端音量发生变化时，应发送音量变化响应命令到发送端设备。命令定义应按照表 47 的规定。

表 47 音量变化命令参数

CALLBACK_ACTION	DATA	类型	描述
onVolumeChanged	VOLUME	Int32	音量，取值范围是 0—100，静音则值为 0

8.2.19 设置音频轨道信息命令的定义

发送端对接收端的播控命令。命令定义应按照表 48 的规定。

表 48 设置音频轨信息命令参数

ACTION	DATA	类型	描述
setAudioTrackInfo	trackInfo	TrackInfo	音频轨道信息，如立体声、环绕声等。 TrackInfo 应按照表 49 的规定

表 49 轨道信息的全部字段信息

DATA	类型	描述
KEY_RENDERER_INDEX	Int32	渲染器索引
KEY_GROUP_INDEX	Int32	组索引
KEY_TRACK_INDEX	Int32	轨迹索引
KEY_TRACK_NAME	String	轨迹名称
KEY_SELECTED	bool	是否被选中，false：未被选中；true：被选中
KEY_WIDTH	Int32	宽度
KEY_HEIGHT	Int32	高度
KEY_BITRATE	Int32	比特率
KEY_CHANNEL_COUNT	Int32	通道数目
KEY_LANGUAGE	String	语言

8.2.20 设置视频轨信息命令的定义

发送端对接收端的播控命令。命令定义应按照表 50 的规定。

表 50 设置视频轨信息命令参数

ACTION	DATA	类型	描述
setVideoTrackInfo	trackInfo	TrackInfo	视频轨道信息，如标清、高清等。TrackInfo 应按照表 49 的定义

8.2.21 设置音/视频轨信息命令的响应定义

接收端音、视频轨信息变化时，应发送音视频轨信息响应命令。命令定义应按照表 51 的规定。

表 51 音视频轨信息响应命令参数

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 应按照表 52 的规定

表 52 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，TrackInfo 定义应按照表 46 的规定
KEY_AUDIO_HAS_SELECTION_OVERRIDE	bool	是否使用默认音频轨，true：使用默认音频轨，false：不使用默认音频轨
KEY_IS_AUDIO_SUPPORT_AUTO	bool	是否支持自动音频轨，支持：true；不支持：false
KEY_VIDEO_TRACKINFO_LIST	List<TrackInfo>	视频轨列表，TrackInfo 应按照表 46 的规定

表52 （续）

DATA	类型	描述
KEY_VIDEO_HAS_SELECTION_OVERRIDE	bool	是否使用默认视频轨，false：不使用默认视频轨；true：使用默认视频轨
KEY_IS_VIDEO_SUPPORT_AUTO	bool	是否支持自动视频轨，false：不支持自动视频轨；true：支持自动视频轨

8.2.22 DRM 密钥请求的定义

播放 DRM 资源时，接收端渲染服务模块需要获取解密密钥，应发送秘钥请求命令道发送端请求。命令定义应按照表 53 的规定。

表 53 DRM 密钥请求命令参数

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

8.2.23 DRM 密钥回复的定义

发送端设备获取到 DRM 解密秘钥响应后，应发送密钥响应命令到接收端。命令定义应按照表 54 的规定。

表 54 DRM 密钥回复命令参数

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

8.2.24 播放器错误的定义

在接收端操作或发送端通过播控命令进行播放，过程中会产生播放异常，接收端应发送播放异常信息到发送端。命令定义应按照表 55 的规定。

表 55 播放错误回调参数

CALLBACK_ACTION	DATA	类型	描述
onPlayerError	ERROR_CODE	Int32	错误码，定义见表 56
	ERROR_MSG	string	错误信息，定义见表 56

表 56 异常信息定义

错误类型	ERROR_MSG	ERROR_CODE
未知异常	ERROR_CODE_UNSPECIFIED	1000
图片渲染异常	ERROR_CODE_BITMAP	10000
图层不可用	ERROR_CODE_SURFACE	10001
播放器报错	ERROR_CODE_PLAYER_ARGUMENT_EXCEPTION	10002
播放器不支持数据源	ERROR_CODE_MEDIA_PLAYER_SET_DATA_SOURCE	10003
网络异常	ERROR_CODE_CREATE_CHANNEL_TIME_OUT	10004

表56 （续）

错误类型	ERROR_MSG	ERROR_CODE
播放器不支持对应控制参数	ERROR_CODE_PLAY_PARAMS_UNAVAILABLE	10005
DRM 密钥异常	ERROR_CODE_PROVIDE_KEY_RESPONSE	10006
播放器不支持对应的 DRM 种类	ERROR_CODE_NOT_SUPPORT_DRM_CAPABILITY	10007
播放器启动失败	ERROR_CODE_PLAYER_START_FAIL	10008
不支持的流媒体协议	ERR_CODE_UNSUPPORTED_SCHEME	10009
不支持的文件格式	ERR_CODE_UNSUPPORTED_FILE_FORMAT	10010
音频格式不支持	ERR_CODE_UNSUPPORTED_AUDIO_CODEC	10011
视频格式不支持	ERR_CODE_UNSUPPORTED_VIDEO_CODEC	10012
DNS 解析错误	ERR_CODE_DNS_RESOLVE	10013
媒体流数据拉取超时	ERR_CODE_MEDIADATA_TIMEOUT	10014

9 规格定义

9.1 概述

本章定义接收端渲染服务播放器可支持的播放内容格式，涉及视频、音频、图片格式，由于多媒体格式较多，且在不断更新，本章节列举部分，应支持必选，宜支持可选。

9.2 视频支持的格式

接收端支持视频内容播放时，支持的媒体类型见表 57 中规定，应支持必选，宜支持可选。

表 57 视频媒体格式

支持的媒体类型		可选/必选
视频	MP4	必选
	fMP4	必选
	WebM	可选
	AVI	可选
	MOV	可选
	MKV	可选
	VOB	可选
	FLV	可选
	MPEG	可选
	WMV	可选
	3G2	可选
	3GP	可选
	RM	可选
	RMVB	可选

9.3 音频支持的格式

接收端支持音频内容播放时，支持的媒体类型见表 58 中规定，应支持必选，宜支持可选。

表 58 音频媒体格式

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

9.4 图片支持的格式

接收端支持图片内容播放时，支持的媒体类型见表 59 中规定，应支持必选，宜支持可选。

表 59 图片媒体格式

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

9.5 流媒体通信协议类型

接收端支持视频流媒体内容播放时，支持的媒体类型见表 60 中规定，应支持必选，宜支持可选。

表 60 流媒体格式定义

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

9.6 DRM 支持的类型

接收端支持在线视频 DRM 内容播放时，见表 61 中规定，应支持必选，宜支持可选。

表 61 DRM 媒体类型

支持的媒体类型		可选/必选
DRM	ChinaDRM	可选
	WIDEVINE	可选
	PLAYREADY	可选
	CLEARKEY	可选

9.7 编解码格式

接收端支持视频内容播放时，解码能力见表 62 中规定，应支持必选，宜支持可选。

表 62 视频解码类型

支持的媒体编解码格式		可选/必选
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

附录 A

(资料性）

播控命令示例

A.1 概述

本附录描述了常用的播控命令示例。

A.2 发送端获取接收端播放能力

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

接收端回复信息：
RTSP/1.0 200 OK
Cseq:*seq*
Content-Type: text/parameters
Content-Length: *消息体长度*

his_player_controller_capability: *播放能力*

A.3 发送端获取接收端播放质量

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_qoe

接收端回复播放质量信息：
RTSP/1.0 200 OK
Cseq:*seq*
Content-Type: text/parameters
Content-Length: *消息体长度*

his_player_qoe: *播放质量*

A.4 双端保活（keep alive）

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

A.5 发送端播放命令示例

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: 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"*}

A.6 接收端播放响应示例

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*}}

参考文献

[1] IETF RFC2104 HMAC: Keyed-Hashing for Message Authentication 用于消息身份验证的密钥散列
[2] IETF RFC2326 Real Time Streaming Protocol 实时流媒体协议
[3] IETF RFC6762 Multicast DNS 多播域名系统
[4] IETF RFC6763 DNS-Based Service Discovery 基于多播域名系统的服务发现
[5] IETF RFC5869 HMAC-based Extract-and-Expand Key Derivation Function (HKDF) 基于 HMAC 的提取和扩展密钥派生函数