

版权保护最佳实践

1. 功能介绍

保利威致力于提升视频播放的安全性，保护您的视频资源。通过自研的视频版权保护方案（VRM）和自有专利的加密算法，保利威能够最大限度保证视频文件的安全。您在对接保利威 PlaySafe 视频保护功能时，建议参考本文的最佳实践，以获得最佳的视频安全保护效果。

重要安全原则

`secretKey` 仅用于服务端请求签名，不得写入 Web、移动 App、桌面客户端或其他客户端包中，也不得通过客户端接口下发。客户端不应直接调用创建播放凭证接口；应由业务服务器创建 PlaySafe token，再将 token 返回客户端供播放器使用。

2. 整体方案

对接视频加密功能，需要您的服务器侧和客户端侧进行一定量的修改，以下流程图展示了播放加密视频功能的整体工作流程：



接下来对流程图中的各个步骤进行详细描述：

1. 您的终端向业务服务器请求播放授权

播放前，终端（包括 Web 端、移动 App 端和 PC 播放器等）通过业务接口提交待播放视频的 vid 及必要的身份信息，由业务服务器校验登录态和观看权限。

校验通过后，业务服务器创建对应的 PlaySafe token，并将 token 返回终端供播放器使用。终端不得携带、保存或使用 `secretKey`，业务接口应使用 HTTPS 协议。

2. 您的服务器创建 PlaySafe token

您的服务器端应使用仅保存在服务端的 `secretKey`，通过 [获取视频播放凭证](#) 接口向保利威服务器创建加密视频播放所需的 PlaySafe token。请勿由客户端携带 `secretKey` 创建 token。

- 建议在点播管理后台配置 [IP 白名单](#)，仅添加业务服务器的固定出口 IP。配置后，只有白名单内的服务器可以调用创建 token 接口，其他 IP 无法创建 token。

3. 保利威服务器返回Token给您的服务器

保利威服务器在收到您的获取指定视频播放凭证的请求后，在校验无异常时，会向您的服务器返回一个用于播放加密视频的 Token

4. 您的服务器端返回 token 给您的终端

您的服务器在收到保利威返回的 token 后，将 token 返回终端。业务接口应使用 HTTPS，并根据业务的登录态和视频观看权限校验请求；接口响应中不得包含 `secretKey` 或用于签名的其他服务端密钥。如有额外安全要求，可在服务端对 token 进行二次封装或加密。

5. 您的终端使用点播 SDK 播放视频

在播放加密视频前，终端调用点播 SDK 的接口，传递从业务服务器获得的 token。若业务服务器对 token 进行了二次加密或封装，需要先按对应规则还原，再传递真实 token 给点播 SDK。

下面将对您需要改动的地方进行说明，并提供示例代码以供参考。

3. 服务器侧

您的服务器需要关注的是上图中“客户服务器和保利威服务器的交互”这一部分。



在服务器侧，应提供业务接口：在校验终端用户的登录态和视频观看权限后，使用服务端保存的密钥向保利威服务器创建指定视频（指定 vid）的 PlaySafe token，并将 token 返回终端。

服务端需要遵循以下要求：

- 将 `secretKey` 保存在服务端的环境变量、配置中心或密钥管理服务中；不得写入 App、前端代码、安装包或公开仓库。
- 仅由服务端使用 `secretKey` 计算签名并调用创建 token 接口；终端只请求业务服务器提供的播放 token。
- 在点播管理后台配置 IP 白名单，仅允许业务服务器的固定出口 IP 创建 token。

具体功能点如下：

- 1) 获取接口请求参数：视频 vid，用户 viewerId，以及其他参数；
- 2) 请求保利威服务器的 [获取视频播放凭证](#) 接口，移动端请求需对 viewerId 进行 base64 处理；
- 3) 解析接口返回值，获取到 Token；
- 4) 按业务安全要求对 token 进行二次封装或加密；
- 5) 通过 HTTPS 向客户端返回 token，且不返回 `secretKey` 等服务端密钥。

3.1 示例代码

```
/**
 * 获取点播加密视频的播放token
 */
@PostMapping("/getVodToken")
@ResponseBody
public ResponseVO getVodToken(@RequestBody GetTokenRequestVO tokenReq, HttpServletRequest request) throws
IOException, NoSuchAlgorithmException {
    // USER_ID 和 SECRET_KEY 仅从服务端安全配置读取，禁止下发或写入客户端
    // 请求参数处理
    String videoId = tokenReq.getVideoId();
    String viewerId = tokenReq.getViewerId();
    if (StringUtils.isEmpty(videoId) || StringUtils.isEmpty(viewerId)) {
        return ResponseVO.failure("argument is error");
    }
    Map<String, Object> args = new HashMap<>(16);
    args.put("videoId", videoId);
    args.put("viewerId", viewerId);
    args.put("viewerIp", IPUtil.getIPAddress(request));
    args.put("viewerName", tokenReq.getViewerName());
    args.put("expires", tokenReq.getExpires());
    args.put("disposable", tokenReq.getDisposable());
    args.put("iswxa", tokenReq.getIswxa());
    args.put("userId", USER_ID);
    args.put("ts", System.currentTimeMillis());
    // 去除空值参数
    Map<String, String> params = args.entrySet()
        .stream()
        .filter(entry -> entry.getValue() != null)
        .collect(
            Collectors.toMap(Map.Entry::getKey, entry -> String.valueOf(entry.getValue()))
        );
    // 请求参数签名
    params.put("sign", LiveSignUtil.getSign(params, SECRET_KEY));

    // 向保利威服务器请求Token
    TokenVO vo = RequestTokenService.requestTokenFromServer(params);
    if (vo == null || vo.getCode() == null || vo.getCode() != HttpStatus.OK.value()) {
        return ResponseVO.failure("请求数据失败");
    } else {
        // 可按业务安全要求对 token 进行二次封装；接口须使用 HTTPS
        return ResponseVO.success(vo.getData());
    }
}
```

其中 requestTokenFromServer 方法实现了调用保利威服务器 [获取视频播放凭证](#) 接口获取 Token

```
private const val API_URL = "http://hls.videocc.net/service/v1/token"

/**
 * 向保利威服务器请求Token
 * POST方式, Content-type: application/x-www-form-urlencoded
 */
fun requestTokenFromServer(params: Map<String, String>): TokenVO {
    return httpClient.submitForm(
        url = API_URL,
        formParameters = Parameters.build {
            params.forEach { (key, value) ->
                append(key, value)
            }
        }
    ).bodyAsObject<TokenVO>()
}
```

更详细的代码可以移步 [Gitee地址](#) 进行浏览

3.2 GetTokenRequestVO 参数描述

参数名	必填	类型	说明
videoid	true	String	视频id, 例如: e6b23c6f519c5906e54a13b8200d7bb0_e
viewerId	true	String	观看者id, 要求不同的观看者使用不同的id; 移动端请求需对viewerId进行base64处理
viewerIp	false	String	观看者ip, 如果为空, 会自动获取调用该接口时的ip
viewerName	false	String	观看者名称
expires	false	Long	token有效时长, 单位为秒。为空时默认为10分钟, 有效期为最长24小时。

参数名	必填	类型	说明
disposable	false	Boolean	token有效期，默认为false true: token仅一次有效（验证一次后，token就失效了） false: 在有效期内可以进行多次验证。
iswxa	false	Integer	是否微信小程序播放，默认为0 1: 是 0: 否
extraParams	false	String	自定义的其它参数

3.3 TokenVO 参数描述

参数名	类型	说明
token	String	token值
userId	String	保利威点播账户id
appId	String	账号appId
videoId	String	视频id，例如： e6b23c6f519c5906e54a13b8200d7bb0_e
viewerId	String	观看者id
viewerIp	String	观看者ip
viewerName	String	观看者名称
ttl	Long	token有效时长，单位为毫秒
disposable	Boolean	token有效期 true: token仅一次有效（验证一次后，token就失效了） false: 在有效期内可以进行多次验证。

参数名	类型	说明
iswxa	Integer	是否微信小程序播放 1: 是 0: 否
extraParams	String	自定义的其它参数
createdTime	Long	token创建时间，13位毫秒级时间戳
expiredTime	Long	token过期时间，13位毫秒级时间戳

4. 终端侧

终端侧需要关注的是上图中“客户终端和客户服务器的交互”、“客户终端和保利威点播SDK的交互”这两部分。

播放加密视频工作流程图-2

在终端侧，您需要做的事情如下：

1) 视频播放前，调用业务服务器自行开发的接口，通过视频 vid 获得对应视频的 PlaySafe token。终端不保存或传递 `secretKey`，也不直接调用创建 token 接口。

注意，移动端（iOS端和android端）向您的服务器请求获取播放 Token 时，请求参数的 viewerId 必须经过 Base64 编码。

2) 若业务服务器对 token 进行了二次加密或封装，先根据对应规则还原 token。

3) 通过保利威点播 SDK 所提供的 API 传递从业务服务器获取的 token。

上述步骤1和2，属于“客户终端和客户服务器的交互”，需要您自行开发，这里不予细述，以下示例代码主要展示了步骤3——如何调用点播 SDK 的 API 传递 Token，这部分属于“客户终端和保利威点播SDK的交互”。

4.1 Web 端

完整的示例代码请参考 [点播 JS SDK 播放加密视频](#)

```
var player = polyvPlayer({
    wrap: '#player',
    width: 800,
    height: 533,
    vid: '88083abbf5bcf1356e05d39666be527a_8',
    playsafe: '81814fed-bdd0-4506-bec1-ebc8093148c5-hfewsfxcbsbcocx', // 通过 playsafe 传入 token
    ts: '1568131545000',
    sign: '88313661ba7ded642c7b557b0a364b4b'
});
```

4.2 iOS 端

完整的示例代码请参考 [点播 iOS SDK 4.视频播放](#) 4.1.2 外部传入播放凭证。

```
// 通过配置播放器的 requestCustomKeyTokenBlock 属性，播放器内部在开始播放加密视频时，将会自动执行该 block，并使用
// block 返回的 token 对加密视频进行解密。
self.player.requestCustomKeyTokenBlock = ^NSString *(NSString *vid) {
    // 通过参数 vid，向客户自己的服务器请求 token
    NSString *encodeToken = @"xxxxxxx";
    // 将服务器返回的 token 进行解密
    NSString *decodeToken = @"yyyyyyyy";
    return decodeToken;
};
// 通过 vid 获取在线播放视频模型
[PLVVodVideo requestVideoWithVid:vid completion:^(PLVVodVideo *video, NSError *error) {
    // 播放视频
    weakSelf.player.video = video;
}];
```

4.3 Android 端

4.3.1 视频播放

完整的示例代码请参考 [点播 Android SDK 4.视频播放](#) 1.2 外部传入播放凭证

```
videoView.setVideoTokenRequestListener(new IPLVVideoTokenRequestListener() {
    @Override
    public String onRequestToken(PolyvVideoVO videoVO, String viewerId, String viewerName, String viewerParam) {
        // 返回外部获取的 token
        return token;
    }
});
// 通过vid播放视频
videoView.setVid(vid, bitrate, isMustFromLocal);
```

4.3.2 视频下载

完整的示例代码请参考 [点播 Android SDK 5.视频下载](#)

```
download.setDownloaderTokenRequestListener(new IPLVDownloaderTokenRequestListener() {  
    @Override  
    public String onRequestToken(@NotNull String videoId, int bitRate) {  
        // 返回视频下载的token  
        return token;  
    }  
});
```

4.4 C++ SDK

C++ SDK 如何传递 Token 并播放视频代码示例：

```
// 创建播放器对象，window为播放窗口句柄  
auto player = PLVPlayerCreate(window);  
// 播放前要先设置vid，videoPath为离线播放的视频存放地址，videoRate为视频的清晰度  
PLVPlayerSetVideo(player, vid, videoPath, videoRate);  
// 开始在线播放，token通过业务侧获取传入，seekMillisecond开始播放时想要seek的位置，sync是否同步(播放时会请求videoJson，  
// 同步时会阻塞等待请求结果，异步时通过回调通知结果)  
PLVPlayerPlay(player, token, seekMillisecond, sync);
```

4.5 APICloud 平台

APICloud 如何调用点播 SDK 的 API 传递 Token 并播放视频代码示例：

```
// 设置 播放外部播放 token  
this.polyvVideo.setCustomVideoToken({token:""}, function (ret, err) {  
});  
// 通过 vid 播放在线视频  
this.polyvVideo.setVid({  
    vid: ""  
});
```

4.6 uniapp 平台

uniapp 如何调用点播 SDK 的 API 传递 Token 并播放视频代码示例：

```
// 设置 播放外部播放 token
this.$refs.vod.setCustomVideoToken({token: ""},(ret) => {
})
// 通过 vid 播放在线视频
this.$refs.vod.setVid({
    vid: "",
    level: 0
},(ret) => {
})
```