

播放加密视频

保利威视频加密采用私有的加密算法，提供包括加密转码、安全传输、解密播放等一体化的视频安全方案。通过对视频数据本身进行高强度的加密，视频文件即使被下载到本地也无法直接播放和传播，可有效防止视频泄露和盗版问题，详见：[视频加密](#)。

为了方便开发者快速集成，我们将解密播放相关逻辑都封装到了播放器内部，与播放普通视频相比，在播放器端，开发者只需多传入播放凭证参数即可实现加密视频的播放。

开发指引

加密视频需使用播放凭证方式进行播放。播放器支持以下播放凭证参数，请根据接入方式传递对应参数。新接入建议使用 `playsafe` 或 `playsafeUrl`。

名称	类型	说明
<code>playsafe</code>	String/Function	播放加密视频推荐使用的授权凭证，适用于 PC Web 和移动端 H5。业务方服务端通过 创建 Playsafe Token 接口获取并返回给播放器。支持直接传入 token，也支持通过 Function 方式动态获取 token。
<code>playsafeUrl</code>	String	获取播放加密视频凭证的接口 URL，适用于 PC Web 和移动端 H5 (**与 <code>playsafe</code> 参数二选一**)。新接入建议优先使用 <code>playsafeUrl</code> 或 <code>playsafe</code> Function，由业务方服务端实时返回 token。
<code>ts</code>	Number	移动端 H5 播放加密视频的兼容参数，表示 13 位毫秒级时间戳。该参数仍然有效，但不再作为推荐接入方式。

名称	类型	说明
sign	String	移动端 H5 播放加密视频的兼容签名，生成规则为点播账号的 secretkey、vid、ts 的值按顺序拼接后进行 md5 计算，由业务方服务端生成并返回给播放器。该参数仍然有效，但不再作为推荐接入方式。 注：sign 签名不需要转大写。 签名示例： 若 secretkey 为 abc，vid 为 123，ts 为 1672829071000，此时 sign 为 md5("abc1231672829071000")

注：新接入建议 PC Web 和移动端 H5 统一使用 **playsafe** 或 **playsafeUrl** 获取播放 token。**ts**、**sign** 仍可用，主要用于历史兼容，不再推荐使用。

播放器代码示例：

```
<div id="player"></div>
<script src="//player.polyv.net/resp/vod-player/latest/player.js"></script>
<script>
var player = polyvPlayer({
  wrap: '#player',
  width: 800,
  height: 533,
  vid: '88083abbf5bcf1356e05d39666be527a_8',
  // 推荐：传入业务方服务端获取到的 playsafe token，PC Web 和移动端 H5 均可使用
  playsafe: '81814fed-bdd0-4506-bec1-ebc8093148c5-hfevwsfxcsbcocx'
  // 如需通过业务方自定义接口动态获取播放凭证，可将 playsafe 替换为：
  // playsafeUrl: 'https://myDomain.com/token'
});

// 切换加密视频时，需要重新获取播放凭证。如果初始化播放器时使用了 playsafeUrl 参数，则播放器会自动获取新的凭证，无需传 playsafe 参数。
player.changeVid({
  vid: '88083abbf5bcf1356e05d39666be527a_9', // 需要切换的视频 vid
  playsafe: '81814fed-bdd0-4506-bec1-ebc8093148c6-hfevwsfxcsbcocx' // 新获取的 playsafe token
});
</script>
```

Web 页面播放加密视频前，需要先访问业务方自己的服务端授权验证接口（可以在这里加上自有业务的授权验证逻辑，例如是否登录、是否购买课程等，建议使用 HTTPS）。如果业务上允许播放，则通过[创建 Playsafe Token](#)接口获取播放凭证，并返回给 Web 端播放器。新接入建议 PC Web 和移动端 H5 都统一使用 Playsafe Token。旧版移动端 H5 的 ts、sign 方式仍然有效，但主要用于历史兼容，不再作为推荐接入方式。

服务端生成播放凭证代码示例：

PHP 代码：

// 接口中应附带自有业务的授权验证逻辑，如判断是否登录、是否购买课程等

// 以下为生成播放凭证的代码示例

```
function get_client_ip() {
    if (!empty($_SERVER['HTTP_CLIENT_IP'])) {
        $ipaddress = $_SERVER['HTTP_CLIENT_IP'];
    } elseif (!empty($_SERVER['HTTP_X_FORWARDED_FOR'])) {
        $ipaddress = $_SERVER['HTTP_X_FORWARDED_FOR'];
    } else {
        $ipaddress = $_SERVER['REMOTE_ADDR'];
    }
    return $ipaddress;
}

$userid = 'your userid';          // 保利威点播账号的 userid
$secretkey = 'your secretkey';    // 保利威点播账号的secretkey
$videoId = '88083abbf5bcf1356e05d39666be527a_8'; // 视频id
$ts = time() * 1000;              // 时间戳
$viewerIp = get_client_ip();      // 观众ip
$viewerId = '12345';              // 观众id
$viewerName = 'testUser';         // 观众昵称，若值为中文需要urlencode('张三')
$extraParams = 'HTML5';           // 自定义扩展参数
$disposable = 'false';            // true 表示 token 仅一次有效。false 则表示在有效期内可以多次验证。默认为 false。

/* 将参数 userid、videoId、ts、viewerIp、viewerId、viewerName、extraParams、disposable 按照 ASCII 升序 key + value
+ key + value ... + value 拼接
*/
$concatd =
'disposable'.$disposable.'extraParams'.$extraParams.'ts'.$ts.'userid'.$userid.'videoId'.$videoId.'viewerId'.$viewerId.'viewerIp'.$viewerIp.'viewerName'.$viewerName;
// 首尾加上secretkey值
$plain = $secretkey.$concatd.$secretkey;
// 取大写MD5
$sign = strtoupper(md5($plain));

// 然后将下列参数用post请求 https://hls.videooc.net/service/v1/token 获取 token
$url = 'https://hls.videooc.net/service/v1/token';
$data = array('userid' => $userid, 'videoId' => $videoId, 'ts' => $ts, 'viewerIp' => $viewerIp, 'viewerName' =>
$viewerName, 'extraParams' => $extraParams, 'viewerId' => $viewerId, 'disposable' => $disposable, 'sign' =>
$sign);
$options = array(
    'http' => array(
        'header' => "Content-type: application/x-www-form-urlencoded",
        'method' => 'POST',
        'content' => http_build_query($data)
    )
);
$context = stream_context_create($options);
$result = file_get_contents($url, false, $context);

// 获取接口返回结果中的token值，并传给播放器播放加密视频
$token = json_decode($result)->data->token;
echo $token;
```

Java SpringMvc代码:

```

@ResponseBody
@RequestMapping("/playerSafe")
public String playerSafe(HttpServletRequest request) {
    String userid = "your userid";        // 保利威点播账号的 userid
    String secretkey = "your secretkey";    // 保利威点播账号的secretkey
    String videoId = "88083abbf5bcf1356e05d39666be527a_8"; // 视频id
    long ts = System.currentTimeMillis();    // 时间戳
    String viewerIp = getClientIp(request); // 观众ip
    String viewerId = "12345";                // 观众id
    String viewerName = "testUser"; // 观众昵称，若值为中文需要urlencode('张三')
    String extraParams = "HTML5"; // 自定义扩展参数
    boolean disposable = false; // true 表示 token 仅一次有效。false 则表示在有效期内可以多次验证。默认为 false。

    /* 将参数 userid、videoId、ts、viewerIp、viewerId、viewerName、extraParams、disposable 按照 ASCII 升序 key +
    value + key + value ... + value 拼接
    */
    String concated = "disposable" + disposable + "extraParams" + extraParams + "ts" + ts + "userid" + userid +
        "videoId" + videoId + "viewerId" + viewerId + "viewerIp" + viewerIp + "viewerName" + viewerName;
    // 首尾加上secretkey值
    String plain = secretkey + concated + secretkey;
    // 取大写MD5，可自行选择md5库
    String sign = md5Hex(plain).toUpperCase();

    // 然后将下列参数用post请求 https://hls.videocc.net/service/v1/token 获取 token
    String url = "https://hls.videocc.net/service/v1/token";

    Map<String, String> params = new HashMap<>();
    params.put("userid", userid);
    params.put("videoId", videoId);
    params.put("ts", String.valueOf(ts));
    params.put("viewerIp", viewerIp);
    params.put("viewerName", viewerName);
    params.put("extraParams", extraParams);
    params.put("viewerId", viewerId);
    params.put("disposable", String.valueOf(disposable));
    params.put("sign", sign);
    // 可自行选择http客户端
    String response = HttpClientUtil.getInstance().sendHttpPost(url, params);

    try {
        //解析json
        ObjectMapper objectMapper = new ObjectMapper();
        TokenResponse tokenResponse = objectMapper.readValue(response, TokenResponse.class);
        // 响应代码，200为成功，403为ts过期或签名错误，400为参数错误（例如缺少 userid 或 videoId）
        if (tokenResponse.getCode() == 200) {
            Map data = (Map) tokenResponse.getData();
            return data.get("token").toString();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    return "";
}

static class TokenResponse {
    int code;

```

```

        String status;
        String message;
        Object data;
        //省略getter、setter...
    }
    public String getClientIp(HttpServletRequest request) {
        String ip = request.getHeader("x-forwarded-for");
        if (ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {
            ip = request.getHeader("Proxy-Client-IP");
        }
        if (ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {
            ip = request.getHeader("WL-Proxy-Client-IP");
        }
        if (ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {
            ip = request.getRemoteAddr();
        }
        return ip;
    }
}

```

如果播放器使用playsafeUrl参数的方式，则需输出json格式：

PHP 代码

```

// 已省略获取token的代码...
$token = json_decode($result)->data->token;
$code = json_decode($result)->code; // 响应代码，200为成功，403为ts过期或签名错误，400为参数错误（例如缺少 userid 或 videoId）
$message=json_decode($result)->message;
$status=json_decode($result)->status;
if($code == 200){
    $array = Array("code"=>$code,'status'=>$status,'message'=>$message,"data"=>$token);
}else{
    $array = Array("code"=>$code,'status'=>$status,'message'=>$message,"data"=>json_decode($result)->data);
}
$json = json_encode($array);
echo $json; //输出json

```

Java SpringMvc代码

```

@ResponseBody
@RequestMapping("/playerSafeUrl")
public Map<String, Object> playerSafeUrl(HttpServletRequest request) {
    // 已省略获取token的代码...
    Map<String, Object> resultMap = new LinkedHashMap<>();
    try {
        //解析json
        ObjectMapper objectMapper = new ObjectMapper();
        TokenResponse tokenResponse = objectMapper.readValue(response, TokenResponse.class);
        resultMap.put("code", tokenResponse.getCode());
        resultMap.put("status", tokenResponse.getStatus());
        resultMap.put("message", tokenResponse.getMessage());
        // 响应代码, 200为成功, 403为ts过期或签名错误, 400为参数错误 (例如缺少 userid 或 videoId)
        if (tokenResponse.getCode() == 200) {
            Map data = (Map) tokenResponse.getData();
            resultMap.put("data", data.get("token"));
        } else {
            resultMap.put("data", tokenResponse.getData());
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    return resultMap;
}

```

playsafeUrl返回结果示例：

请求成功：

```

{
  "code": 200,
  "status": "success",
  "message": "",
  "data": "973d7731803940a1b14fdc93941f493c"
}

```

请求失败（签名错误）：

```

{
  "code": 403,
  "status": "error",
  "message": "sign_invalid",
  "data": "sign parameter invalid."
}

```

playsafeUrl可能存在跨域问题，解决方案请参考：[跨域访问设置](#)。

旧版移动端 H5 的 ts + sign 兼容方式

旧版移动端 H5 播放 Web 加密视频时，也可以返回 sign 和 ts 参数给播放器。该方式目前仍然有效，存量业务可以继续使用；新接入建议使用 playsafe 或 playsafeUrl 获取播放 token。

```
// php
$vid = "88083abbf5bcf1356e05d39666be527a_8"; // 视频vid
$secretkey= "your secretkey"; // 保利威点播账号的secretkey
$ts=time()*1000; // 10位的秒级时间戳，后面加多3个0，最后为13位的数值
$sign = md5($secretkey.$vid.$ts);
```