

如何正确使用SDK加密串

保利威点播SDK需要用户提供 `userid`、`readtoken`、`writetoken`、`secretkey` 几个配置信息才能解密播放视频。基于安全性考虑，建议这些参数保存在服务端，APP 在启动时从服务端获取并配置。

为防止 APP 端被嗅探这几个参数，需要对传输的内容进行加密。因此，就有了 `SDK加密串` ——在服务器端对以上四个参数进行加密，APP端对应的进行解密。开发者应当设计自己的加解密方式，并且APP端应通过 **https**协议请求服务端接口。

为方便快速集成和调试SDK的功能，保利威管理后台提供了一种固定加密方式的临时SDK加密串，请勿直接在正式环境使用。

点播SDK集成文档详见：

- [iOS SDK集成文档](#)
- [Android SDK集成文档](#)

正式使用解决方案

一、设计自己的加解密方式。

服务端Java示例:

```
/*该示例采用AES加密算法，开发者也可以选择使用不同的加密算法，并在APP端做对应的解密。*/
String plainString = userid + "," + secretkey + "," + readtoken + "," + writetoken;
String key = "1234567812345678"; // 加密密钥
String iv = "1234567812345678"; // 加密向量

Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
int blockSize = cipher.getBlockSize();

byte[] dataBytes = plainString.getBytes();
int plaintextLength = dataBytes.length;
if (plaintextLength % blockSize != 0) {
    plaintextLength = plaintextLength + (blockSize - (plaintextLength % blockSize));
}

byte[] plaintext = new byte[plaintextLength];
System.arraycopy(dataBytes, 0, plaintext, 0, dataBytes.length);

SecretKeySpec keyspec = new SecretKeySpec(key.getBytes(), "AES");
IvParameterSpec ivspec = new IvParameterSpec(iv.getBytes());

cipher.init(Cipher.ENCRYPT_MODE, keyspec, ivspec);
byte[] encrypted = cipher.doFinal(plaintext);
String baseString = Base64.encode(encrypted);
System.out.println(baseString);
```

服务器端 PHP 加密示例：

```
<?php
/*该示例采用AES加密算法，开发者也可以选择使用不同的加密算法，并在APP端做对应的解密。*/
$privateKey = "1234567812345678";// 加密密钥
$iv = "1234567812345678"; // 加密向量

$userid = "你的userid";
$secretkey = "你的secretkey";
$readtoken = "你的readtoken";
$writetoken = "你的writetoken";
$plainstring = $userid.", ".$secretkey.", ".$readtoken.", ".$writetoken;

$str_padded = $plainstring;
if (strlen($str_padded) % 16) {
    $str_padded = str_pad($str_padded, strlen($str_padded) + 16 - strlen($str_padded) % 16, "\0");
}

$encrypted = openssl_encrypt($str_padded, 'aes-128-cbc', $privateKey, OPENSSL_RAW_DATA, $iv);
echo(base64_encode($encrypted));

?>
```

Android 端解密和配置

```
public class PolyvApplication extends Application {
    @Override
    public void onCreate() {
        new LoadConfigTask().execute();
    }

    private class LoadConfigTask extends AsyncTask<String, String, String> {

        @Override
        protected String doInBackground(String... params) {
            String config = PolyvSDKUtil.getUrl2String("https://demo.ipolyv.cn/demo/appkey.php");
            if (TextUtils.isEmpty(config)) {
                try {
                    throw new Exception("没有取到数据");
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }

            return config;
        }

        @Override
        protected void onPostExecute(String config) {
            PolyvSDKClient client = PolyvSDKClient.getInstance();
            String iv = "1234567812345678"; // 加密密钥
            String aeskey = "1234567812345678"; // 加密向量
            client.settingsWithConfigString(config, aeskey, iv);
            /*
            *开发者若使用其它加密算法，可自行解密并使用
            *client.settingsWithUserid(String userid, String secretkey, String readtoken, String writetoken)方法进行初
            始化。
            */

        }
    }
}
```

iOS 端解密和配置：

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    NSData *data = [NSData dataWithContentsOfURL:[NSURL
URLWithString:@"https://demo.ipolyv.cn/demo/appkey.php"]];
    NSString *configString =[[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding];
    NSString *aesKey = @"1234567812345678";
    NSString *aesIv = @"1234567812345678";
    NSError *error = nil;
    //使用保利威点播 SDK 提供的加密方式得到 SDK 配置信息
    [PLVodSettings settingsWithConfigString:configString key:aesKey iv:aesIv error:&error];

    /* 客户也可以用自己的方法解密得到 secretkey 等信息，并使用以下方法初始化 SDK
    [PLVodSettings settingsWithUserId:userId readtoken:readtoken writetoken:writetoken secretkey:secretkey];
    */
    return YES;
}
```

二、保证服务端接口稳定

服务器端接口不稳定可能会导致加密串无法成功获取，APP播放视频会出现无法解码播放的错误，建议：

- 程序上对接口请求增加失败重试机制；
- 针对加密串和不经常改动的配置，利用CDN加速和缓存，保证高可用性；
- 增加备用接口，当主接口请求和重试都失败时，调用备用接口。

三、对APP进行加固

对于安全性要求较高的公司，可对 APP 进行加固，防止 APP 被反编译得到源码，从而导致账号及用户隐私数据的泄漏。市面上有很多对 APP进行加固的产品可以使用。