**一键登录标准版H5 jssdk 集成文档**

（V1.02）

目录

[一、 业务及交互流程 3](#_Toc67297559)

[1.1 业务流程 3](#_Toc67297560)

[1.流程图 3](#_Toc67297561)

[2. 流程说明 4](#_Toc67297562)

[1.2 交互流程 5](#_Toc67297563)

[1.时序图 5](#_Toc67297564)

[2. 流程说明 5](#_Toc67297565)

[二、 重要参数说明 6](#_Toc67297566)

[2.1 集成页面地址（referer） 6](#_Toc67297567)

[2.2 请求来源（origin） 7](#_Toc67297568)

[2.3 businessType 7](#_Toc67297569)

[三、 jssdk集成说明 7](#_Toc67297570)

[3.1 引用 7](#_Toc67297571)

[3.2 使用方法 8](#_Toc67297572)

[3.3 获取网络类型（可选） 9](#_Toc67297573)

[四、 token校验接口（服务端） 10](#_Toc67297574)

[4.1 接口定义 10](#_Toc67297575)

[4.2 Request 10](#_Toc67297576)

[4.3 Response 12](#_Toc67297577)

[五、 返回码及描述 13](#_Toc67297578)

[5.1 jssdk 13](#_Toc67297579)

[5.1.1 取号 13](#_Toc67297580)

[5.1.2 获取token 16](#_Toc67297581)

[5.2 服务端token校验 16](#_Toc67297582)

[附录1：token校验接口工具 17](#_Toc67297583)

[1. RSA公私钥 17](#_Toc67297584)

[方法1：使用下面的demo(请下载下面的附件:maven项目) 17](#_Toc67297585)

[方法2：在线生成，url: http://web.chacuo.net/netrsakeypair 18](#_Toc67297586)

[2. SHA256withRSA签名方法 19](#_Toc67297587)

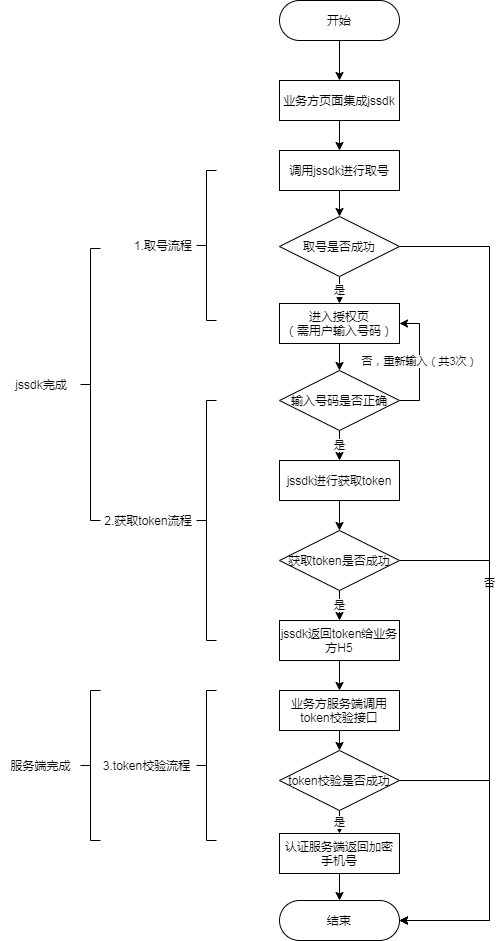
[3. AES解密手机号 19](#_Toc67297588)

[附录2：常见问题 20](#_Toc67297589)

# 业务及交互流程

## 业务流程

### 1.流程图



### 2. 流程说明

1）业务方页面集成 一键登录H5取号jssdk；

2）业务方H5调用jssdk提供的方法获取token，jssdk完成取号流程和获取token流程

* jssdk取号流程：

jssdk进行取号，若取号成功，则进入授权页，用户在授权页补充正确的号码授权后，jssdk进行获取token流程；若取号失败则返回报错

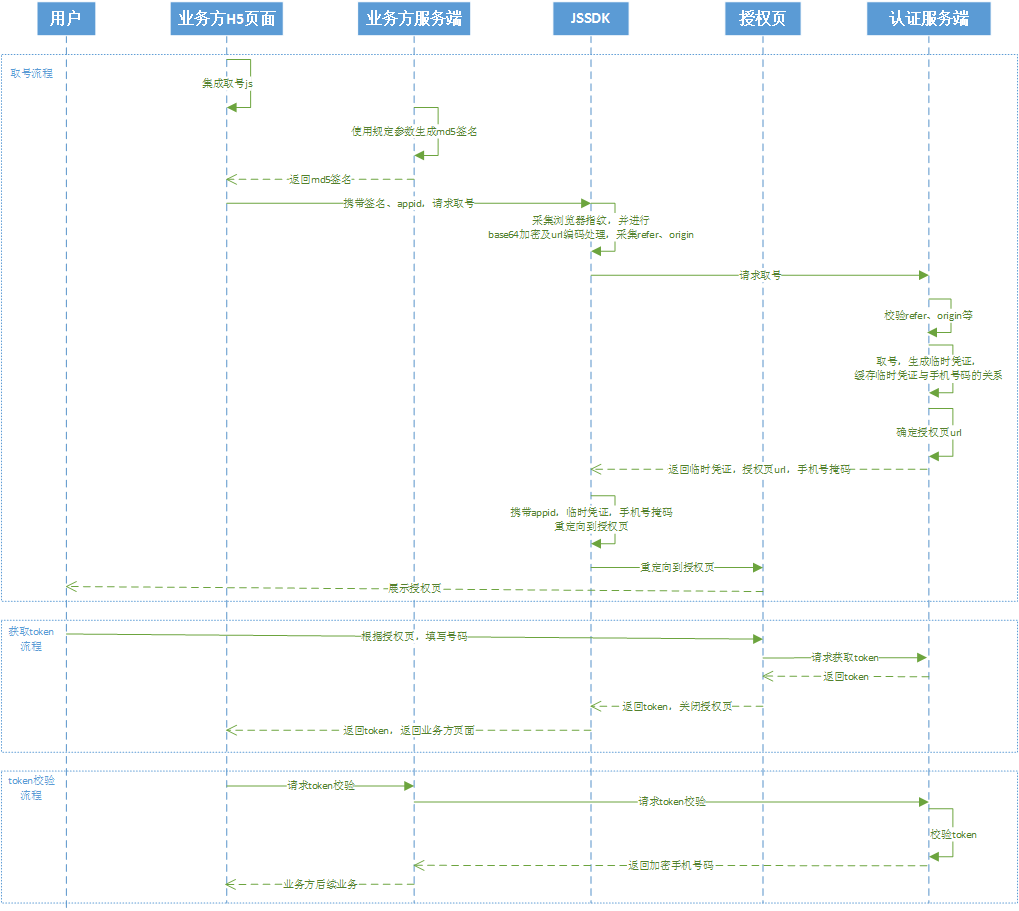
* jssdk获取token流程：

jssdk进行获取token，若获取token成功过，则返回token至业务方H5；若token获取失败则返回报错

3）token校验流程：业务方H5获取token后可传至业务方服务端，由业务方服务端调用token校验接口，若校验通过，则返回加密手机号；若失败则返回报错

## 交互流程

### 1.时序图



### 2. 流程说明

1）业务方H5集成取号js，获取md5签名

2）业务方H5携带签名、appid，调用jssdk请求取号

3）jssdk向认证服务端发起取号请求，取号成功后认证服务端返回临时凭证、授权页url、手机号掩码，jssdk重定向展示授权页面

4）用户授权并补齐正确的手机号码，授权页面发起请求获取token，认证服务端返回token并返回业务方页面

5）业务方服务端发起token校验请求，认证服务端校验通过后返回手机号

# 重要参数说明

## 2.1 集成页面地址（referer）

指集成了取号的页面访问地址，若多个页面集成用英文逗号分隔上报。必须报备可访问的完整页面地址，不允许使用通配符！

注：

（1） 假如业务方集成取号的页面访问地址为<

http://www.abc.com/a/mobile.html/#/abc>，则报备地址为<

http://www.abc.com/a/mobile.html/，

（2） 假如业务方集成取号的页面访问地址为<

http://www.abc.com/a/mobile.html/?abc>，则报备地址为<

<http://www.abc.com/a/mobile.html/>）。

（3） 生产上有 https 访问的，https 请求跨域，会导致上报的 referer 为空，请在 head 中添加代码：<meta content="always" name="referrer">解决此问题。

## 2.2 请求来源（origin）

指发起请求的业务来源。一般为 protocol+host，不包含路径等信息。几种示例如下：

（1）假如业务方集成取号的页面访问地址为<http://www.abc.com/a/mobile.html>，则origin 为<http://www.abc.com>

（2）假如业务方集成取号的页面访问地址为<https://www.abc.com/a/mobile.html>，则origin 为<https://www.abc.com>

（3）假如业务方集成取号的页面访问地址为<http://127.0.0.1:8080/a/mobile.html>，则origin 为<http://127.0.0.1:8080>

（4）假如业务方集成取号的页面访问地址为<https://127.0.0.1:8080/a/mobile.html>，则origin 为<https://127.0.0.1:8080>

## 2.3 businessType

业务类型，本业务businessType=8

# jssdk集成说明

## 3.1 引用

（1）引用crypto-js.js（支持本地引入）



（2）引用jssdk.min.js

<script type="text/javascript" src=" https://www.cmpassport.com/h5/js/jssdk\_auth/jssdk-1.0.0.min.js "></script>

## 3.2 使用方法

YDRZAuthLogin.getTokenInfo({

data: { //请求的参数

version: '2.0', //接口版本号 （必填）

appId: appId, //应用Id （必填）

sign: sign, //加密后的sign（必填）。sign生成规则： MD5(appId +

businessType + msgId + timestamp + traceId + version+appkey)

注：msgID要和traceID保持一致，此处sign与token校验接口sign的生成规则不同

traceId: traceId, //业务方生成唯一标识

timestamp:timestamp,

//请求消息发送的系统时间，精确到毫秒，共17位，格式：20121227180001165

openType: '1', //取号类型

expandParams: '', //扩展参数 格式：参数名=值 多个时使用 \| 分割 （选填）

authPageType: '0' , //若值为“1”时展示弹窗，若为其它值则展示页面

},

success: function(res) { //成功回调},

error: function(res) { //错误回调}

})

**成功回调：**

{code："103000"，message："token获取成功",token:"",userInformation:'浏览器指纹'}

**失败回调**

1）取号失败

{

msgId:'',

YDData:{code:' 错误码', message:'移动取号失败描述，详见表格'},

CTData:{ code:' 错误码', message:'电信取号失败描述，详见表格'},

CUData:{ code:' 错误码', message:'联通取号失败描述，详见表格'}

}

1. 获取token失败

{msgId:’’, code:’错误码’,message:’错误描述，详见表格’}

## 3.3 获取网络类型（可选）

注：此方法为可选，可以获取也可不获取。iOS系统下由于兼容问题，netType基本返回结果为unknown 未知。

定义connection变量通过YDRZAuthLogin.getConnection方法获取当前手机网络状态

示例：

var connection = YDRZAuthLogin.getConnection(appid);

返回对象：

{

appid:"appid",

msgid:"唯一标识",

netType:（cellular 数据流量、unknown 未知、wifi）3种状态

}

可根据netType状态自行选择后续流程。

# token校验接口（服务端）

## 4.1 接口定义

|  |  |
| --- | --- |
| 接口名称 | tokenValidate |
| 接口描述 | h5前端通过请求电信回调接口,获取到凭证 |
| 承载协议 | HTTP |
| 承载网络 | 公网 |
| 请求方式 | POST |
| 请求限制 |  |
| 数据格式 | Json |
| 接口URL | https://www.cmpassport.com/h5/onekeylogin/tokenValidate |
| 使用说明 |  |

## 4.2 Request

注：请使用post json 格式

#### Header(http 请求头)

|  |  |  |
| --- | --- | --- |
| 字段名称 | 是否必填 | 描述 |
| interfaceVersion | 是 | 接口版本号，填1.0 |
| appId | 是 | 应用ID |
| traceId | 是 | 时间跟踪ID |
| timestamp | 是 | 请求消息发送的系统时间，精确到毫秒，共17位，格式：20121227180001165 |
| businessType | 是 | 业务类型，填8 |

#### body

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段标识 | 字段类型 | 是否必填 | 字段含义 | 说明 |
| token | String | 是 | token身份凭证 |  |
| sign | String | 是 | 签名，业务方RSA私钥生成签名，(appId+traceId+timestamp+token+version)（注：“+”号为合并意思，不包含在被加密的字符串中，生成工具详见附录1） |  |
| userInformation | String | 是 | 浏览器加密指纹（根据jssdk返回） |  |
| expandParams | String | 否 | 扩展参数，不同业务取号特定参数放在此参数中，格式：param1=value1|param2=value2 方式传递，参数以竖线 | 间隔方式传递，此参数需urlencode编码。 |  |

**请求示例：**

**注：设置 header 头：-H 'Content-Type: application/json'**

curl -L -X POST 'https://www.cmpassport.com/h5/onekeylogin/tokenValidate' -H 'interfaceVersion: 2.0' -H 'appId: 300012030047' -H 'traceId: 68ccd9f2-8d8e-4e6d-b646-6aec73278822' -H 'timestamp: 20210318183422402' -H 'businessType: 8' -H 'Content-Type: application/json' --data '{"token":"H5HTTPS6c325f3467873abbcbf9d90bObf09b67a67a5527ed7f48c19f6377f3b401523c","sign":"72AD4B4AFEF5B22110E722B2B27B7846B432836EDAA5CF1191E2EEA449A5401C79830F9D544168649AEA3AE14584DA963D601A43C68DEA6278F63020803BA0C8D5C00C602F862F9D117883D05BE1A8183FCA6291E49F5D36C18431E90E0BA64B93751B7F0FFB6753330E0F291276590A9FC1E499D75962479C6B978EBBF7FC92","userInformation":"aVBob25IQEBNb3ppbGxhLzUuMCAoaVBob25I0yBDUFUgaVBob251IE9TIDEyXzNfMSBsaWtllE1hYyBPUyBYKSBBcHBsZVdlYktpdC82MDUuMS4xNSAoS0hUTUwslGxpa2UgR2Vja28pIFZIQEAOZmNmYmZhY2Y1YzU3Mjg4NzlkY2Y2ODYOYTc5Mzg20Q=="}'

## 4.3 Response

#### Response Header

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 约束 | 参数类型 | 说明 |
| traceId | 必选 | string | 对应请求头的traceId |
| appId | 必选 | string | 应用ID |
| timestamp | 必选 | string | 响应消息发送的系统时间，精确到毫秒，共17位，格式：20121227180001165 |

#### Response Body

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 父对象 | 约束 | 参数类型 | 说明 |
| resultCode |  | 必选 | string | 状态码 |
| desc |  | 必选 | string | 状态码描述 |
| bcode |  | 可选 | string | 附加业务状态码 |
| serviceTime |  | 必选 | string | 时间戳，如：202010201200030 |
| data |  | 必选 | object | 数据子对象, resultCode 成功时候返回 |
| msisdn | data | 必选 | string | AES加密手机号码，加密key使用appkey （AES解密方法见附录1） |
| expandParams | data | 可选 | string | 扩展参数格式：param1=value1|param2=value2 方式传递，参数以竖线 | 间隔方式传递，此参数需urlencode编码。 |
| securityPhone | data | 必选 | string | 手机号掩码，如：138\*\*\*\*5491 |

响应示例：

{

"resultCode": "103000",

"desc": "success",

"serviceTime": "20210310150254479",

"data": {

"msisdn": "m+FDykYrD1qsjp0cBATZug==",

"expandParams": null,

"securityPhone": "188\*\*\*\*1079"

}

}

# 返回码及描述

## 5.1 jssdk

### 5.1.1 取号

1）移动取号：

|  |  |
| --- | --- |
| 返回码 | 描述语 |
| 103000 | 成功 |
| 500 | 网络异常，请检查网络设置 |
| 503 | 参数缺失 |
| 130010 | 参数为空 |
| 105002 | 移动网关取号失败 |
| 105112 | 时间戳非法 |
| 105113 | APPID非法或为空 |
| 103101 | 错误的请求签名 |
| 103134 | WAP取号为空 |
| 110024 | businessType配置错误 |
| 110023 | 应用没有权益 |
| 110025 | 权益已失效 |
| 110026 | Origin校验失败 |
| 103111 | WAP网关IP错误 |
| 111002 | 黑名单号码用户 |
| 103609 | 物联网IP不允许取号 |
| 105002 | 移动网关取号失败 |
| 170001 | referer校验失败 |
| 103211 | 其他错误 |

2）电信取号：

|  |  |
| --- | --- |
| 返回码 | 描述语 |
| 103000 | 成功 |
| 301 | 参数错误 |
| 500 | 网络异常，请检查网络设置 |
| 502 | 电信/联通取号能力关闭 |
| 503 | 参数缺失 |
| 103101 | 错误的请求签名 |
| 105113 | APPID非法或为空 |
| 301 | 参数错误 |
| 130030 | 参数为空 |
| 105117 | TRACEID为空 |
| 108001 | businessType is null（电信预取号接口） |
| 104001 | businessType config error |
| 110025 | 权益已失效 |
| 105113 | APPID非法或为空 |
| 108001 | businessType is null（电信回调接口） |
| 105003 | 电信网关取号失败 |
| 110023 | 应用没有权益 |
| 170001 | referer校验失败 |
| 110026 | Origin校验失败 |
| 130035 | 其他错误 |
| 0 | 处理结果正常 |
| -10000 | 取号异常 |
| -10001 | 取号失败 |
| -10002 | 参数错误 |
| -10003 | 解密失败 |
| -10004 | 无效的IP |
| -10005 | 异网授权回调参数异常 |
| -10006 | 授权失败，且属于电信网络 |
| -10007 | 重定向到异网取号 |
| -10008 | 超过预设取号阀值 |
| -10009 | 时间超期 |
| -10010 | 号码识别异常 |
| -10011 | 运营商不匹配 |
| -10012 | 区域不匹配 |
| -10013 | 业务类型不支持该运营商 |
| -10014 | AES解密失败 |
| -10015 | Ipv6取号失败 |
| -10016 | 安全校验失败 |
| -10017 | redirect方式需要https的callback地址 |
| -20005 | 签名非法 |
| -20006 | 应用不存在 |
| -20007 | 公钥数据不存在 |
| -20100 | 内部解析错误 |
| -20102 | 加密参数解析失败 |
| -30001 | 时间戳非法 |
| -30003 | topClass失效 |
| -99999 | 服务内部错误 |
| 51002 | 参数为空 |
| 51114 | 无法获取手机号数据 |
| 51207 | 获取accessCode使用的appid与本次操作的appid不一致 |
| 51208 | 无效的accessCode,该accessCode无法在该业务中使用 |

3）联通取号：

|  |  |
| --- | --- |
| 返回码 | 描述语 |
| 103000 | 成功 |
| 500 | 网络异常，请检查网络设置 |
| 502 | 电信/联通取号能力关闭 |
| 503 | 参数缺失 |
| 103101 | 错误的请求签名 |
| 105113 | APPID非法或为空 |
| 301 | 参数错误 |
| 130030 | 参数为空 |
| 105117 | TRACEID为空 |
| 108001 | businessType is null（电信预取号接口） |
| 104001 | businessType config error |
| 110025 | 权益已失效 |
| 105001 | 联通网关取号失败 |
| 105113 | APPID非法或为空 |
| 108001 | businessType is null（电信回调接口） |
| 105003 | 联通网关取号失败 |
| 110023 | 应用没有权益 |
| 110025 | 权益已失效 |
| 170001 | referer校验失败 |
| 110026 | Origin校验失败 |
| 130035 | 其他错误 |

### 5.1.2 获取token

|  |  |
| --- | --- |
| 返回码 | 描述语 |
| 501 | 用户取消授权 |
| 103002 | 没有填写必传参数 |
| 104000 | app不存在 |
| 104001 | businessType校验失败 |
| 104003 | 应用没有权益 |
| 104004 | 权益已失效 |
| 104005 | referer校验失败 |
| 104006 | origin校验失败 |
| 104007 | accessToken不存在 |
| 104008 | accessToken校验失败 |
| 104009 | 超过失败次数限制获取Token |
| 104010 | 配置错误 |
| 104011 | 手机号不能为空 |
| 104012 | 本机号码校验失败 |
| 104014 | busineseTYpe校验失败 |
| 104015 | 运管配置错误 |
| 104016 | 开放配置错误 |
| 104018 | 正在处理，请稍后 |
| 104020 | 电信取号异常 |
| 104021 | 联通取号异常 |

## 5.2 服务端token校验

|  |  |
| --- | --- |
| 返回码 | 返回码描述 |
| 103000 | 成功 |
| 103001 | fail |
| 103002 | 没有填写必传参数 |
| 103003 | forbidden |
| 103004 | resource not found |
| 103005 | inner error |
| 104000 | app不存在 |
| 104001 | businessType校验失败 |
| 104003 | 应用没有权益 |
| 104004 | 权益已失效 |
| 104007 | token不存在(token 只有2分钟有效期) |
| 104008 | token校验失败 |
| 104010 | 配置错误 |
| 104013 | 验证签名失败 |
| 104014 | busineseType校验失败 |
| 104015 | 运管配置错误 |
| 104016 | 开放平台配置错误 |
| 104017 | IP校验失败（接口设置了ip白名单） |
| 104019 | token校验失败[MSISDN\_OWNERS] |

# 附录1：token校验接口工具

## 1. RSA公私钥

### 方法1：使用下面的demo(请下载下面的附件:)

|  |  |
| --- | --- |
| 版本 | 下载附件 |
| java 版本 （maven项目） |  |
| Php 版本 |  |

#### 1.1.1 Java版本说明

入口方法：test.Test#main

//生成公私钥

generateKeyPair();

// 校验token 接口 参数 需要签名

testGenSign();

//校验token接口 返回手机号 被 aes 加密了 需要解密

testAesDecode();

**详细描述：**

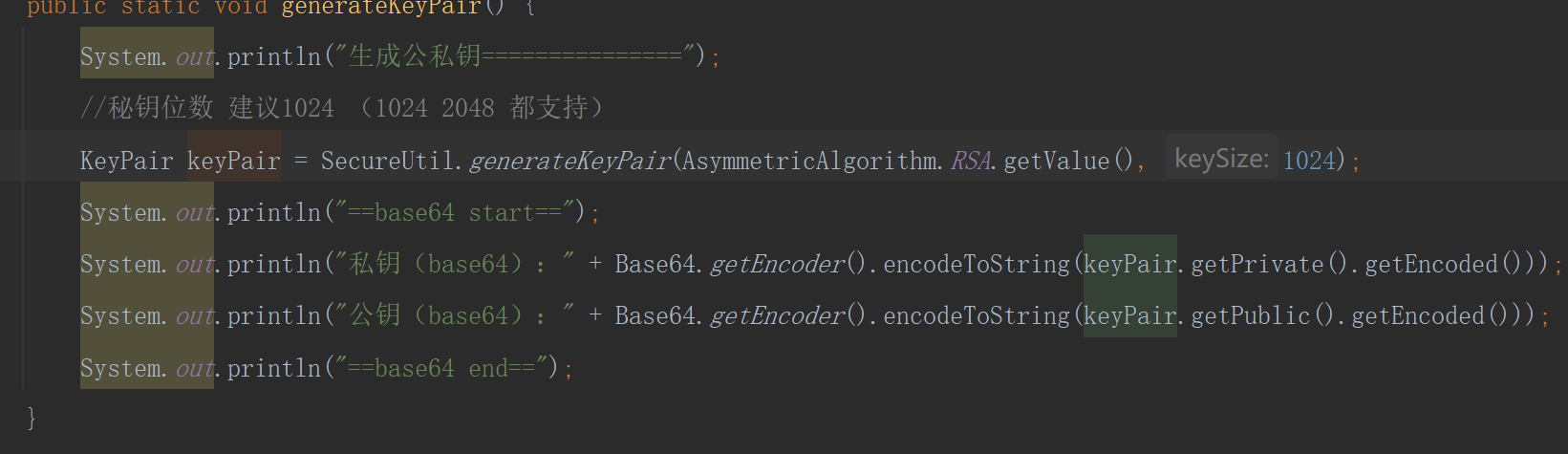
生成RSA公私钥 方法：

KeyPair keyPair = SecureUtil.generateKeyPair(AsymmetricAlgorithm.RSA.getValue(), 1024);

//注意转成 base64

System.out.println("私钥（base64）：" + Base64.getEncoder().encodeToString(keyPair.getPrivate().getEncoded()));

System.out.println("公钥（base64）：" + Base64.getEncoder().encodeToString(keyPair.getPublic().getEncoded()));



#### 1.1.2 php版本说明

入口方法：

// 校验token 接口 参数 需要签名

testGenSign();

//校验token接口 返回手机号 被 aes 加密了 需要解密

testAesDecode();

//生成公私钥

generateRsaKeyPair();

**详细描述：**

生成RSA公私钥 方法：(记得更改 config 配置指向自己本地目录)

function generateRsaKeyPair()

{

$config = array(

'config' => 'C:\app\php\php-8.0.6-Win32-vs16-x64\extras\ssl\openssl.cnf', //安装PHP7会自带这个配置文件

'digest\_alg' => 'sha256',//可以用openssl\_get\_md\_methods() 查看支持的加密方法

'private\_key\_bits' => 1024,

'private\_key\_type' => OPENSSL\_KEYTYPE\_RSA,

);

$res = openssl\_pkey\_new($config);

openssl\_pkey\_export($res, $private\_key\_pem, null, $config);

$details = openssl\_pkey\_get\_details($res);

$public\_key\_pem = $details['key'];

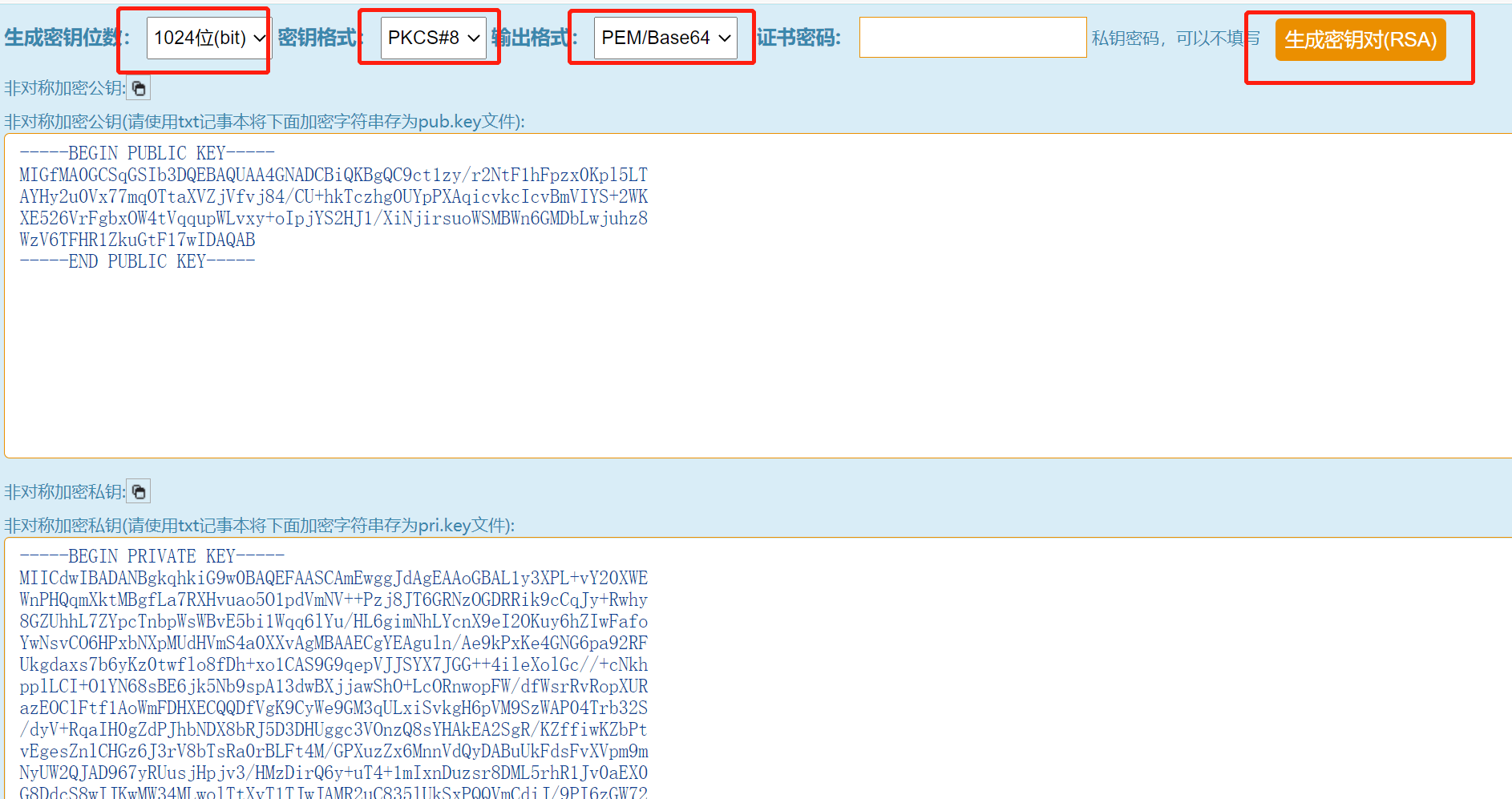
$keypair = array('private\_key' => $private\_key\_pem, 'public\_key' => $public\_key\_pem);

// var\_dump($keypair);

}

### 方法2：在线生成，url: <http://web.chacuo.net/netrsakeypair>

设置项参考下图：



## SHA256withRSA签名方法（使用上面的demo）

#### 2.1 java版本

LySign lySign = LySign.of(SignAlgorithm.SHA256withRSA, privateKey, null, true);

System.out.println("签名数据：" + lySign.sign(data));



#### 2.2 php版本

$lySign = new LySign(privateKey, publicKey);

$sign = $lySign->sign($data);System.out.println("签名数据：" + lySign.sign(data));



## AES解密手机号（使用上面的demo）

#### 3.1 java版本

String phone = "18664805491";

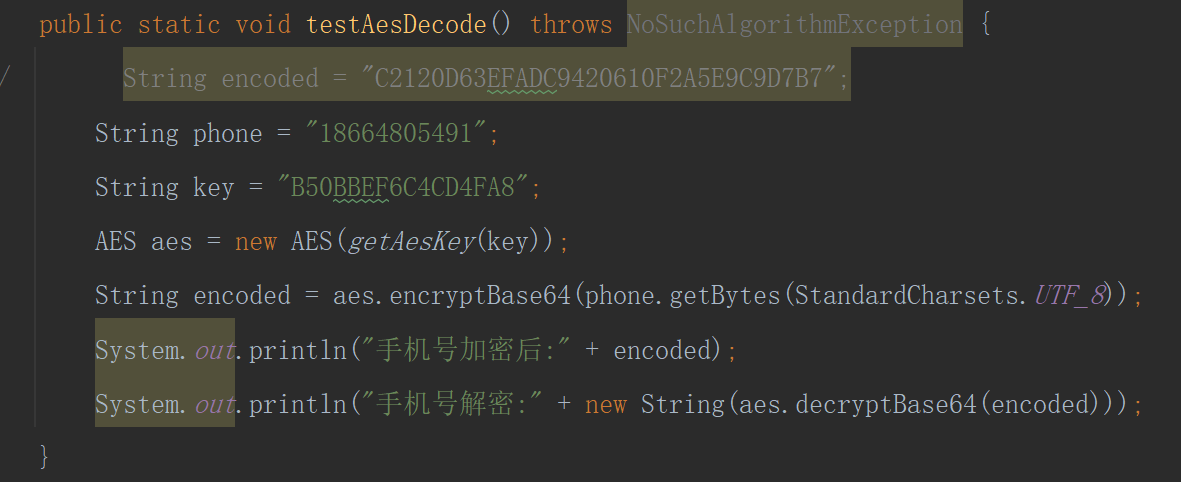
String key = "B50BBEF6C4CD4FA8";

AES aes = new AES(getAesKey(key));

String encoded = aes.encryptBase64(phone.getBytes(StandardCharsets.UTF\_8));

System.out.println("手机号加密后:" + encoded);

System.out.println("手机号解密:" + new String(aes.decryptBase64(encoded)));



#### 3.2 php版本

$phone = "18664805491";

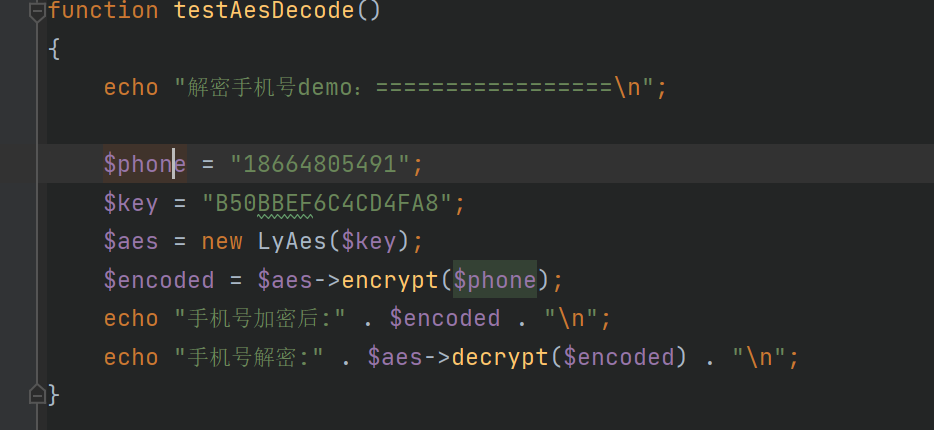
$key = "B50BBEF6C4CD4FA8";

$aes = new LyAes($key);

$encoded = $aes->encrypt($phone);

echo "手机号加密后:" . $encoded . "\n";

echo "手机号解密:" . $aes->decrypt($encoded) . "\n";



# 附录2：常见问题

1. 是否支持wifi环境下使用？

答：仅支持数据流量下使用，不支持wifi、热点环境下使用。请提醒用户请勿在热点环境下使用。

2. 授权页输入错误号码是否有限制次数？

答：同一号码连续输入3次错误，将会被锁定。为避免此问题，请开发者注意在联调过程中请勿连续输错3次。

3. jssdk和token校验接口中的sign是否相同？

答：不相同。两者的生成工具也不同，生成jssdk中的sign请使用附录1中MD5签名工具类；生成token校验接口的sign请使用附录2中“SHA256withRSA签名方法”。

4. 返回“referer校验失败”是什么原因？

答：请检查使用的referer与报备的集成页面地址不完全一致，请使用报备的referer或联系移动认证运营同事更换报备地址

5. token校验接口返回“验证签名失败”是什么原因？

答：①使用的公钥与报备内容是否一致；②未使用附录2提供的工具类生成签名。

6. token校验接口返回“userinfomation 校验失败”是什么原因？

答：请检查token校验接口使用的userinfomation与jssdk返回是否完全一致。由于token校验接口的userinfomation 是业务方前端传给业务方后端的，可能由于该字段作为 url parameter，服务端收到后可能会解码导致出错。请业务方收到userinfomation不要解码（案例：jssdk返回的userinfomation中部分“%3d”，经服务端解码后变成了“=”）。

7. token校验接口返回“token不存在”是什么原因？

答：可能为token失效，token有效期2分钟。

8. token校验接口返回“inner error”是什么原因？

答：①请检查签名是否正确，若直接使用curl签名填写，会导致报inner error；②接口 post json（-H 'Content-Type: application/json'（否则也会报inner error）），header是http的请求头。若确认无误，请联系移动认证侧协助排查。