

# 加密流程第二步--前端调用简洁版

算法： 郭辉铭

时间：2023.1

第三方库：openssl。与之前保持一致

1.md5校验：方案同前，只是已经独立为一个方法

前端调用：encode\_decode.cpp中的

bool encryptStep1(string randStr, string md5App)

其中：

1) randStr.为前端请求算法生成的randStr

2) md5App.为前端通过请求算法生成的随机字符串+前端固定的字符串key（keyFromApp）生成的md5散列值

3) keyFromApp。为前端固定的字符串。需要填写在encode\_decode.cpp中作为外部变量

2.校验通过后进行加密流程的第二步

3.前端直接调用encode\_decode.cpp中的\_getPubkey(function)

--输入：

string function需要一一对应，否则出错。对应关系如下：

曲谱页面框选： "music\_page\_frame"

曲谱正反判断： "updown\_pruned\_v2"

曲谱页面检测： "music\_det"

曲谱中文识别： "music\_rec\_ch"

曲谱外文识别： "music\_rec\_fri"

--输出：生成json类型的字符串如：

{

```
"base64_pu" :  
"LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUIJQklEQU5CZ2txaGtpRzl3MEJBUUVGQUFPQ0F  
RMEFNSUlcQ0FLQ0FRRUExcjJ3RjY4OU1jODhtL3NNWlZXSgpNWG9xTDQ2SkRZam5YaXdXSdImRk  
l5MjJyaGNVQzlidWJlbThjaE16OGd0OVVemZwcFBPWGl5MmUxUHdpNUpQClhXOHNBZHJ6ZWtt  
TjBFaHljeHM3NFlQeWlocXVaWnVYZ0JJM0R3UFRjWDQ4bDRhZ0M2bUhqMVMxT1Y1NG1hOW0KcS  
9nZzlxXbXRmYkFwK2JST3ptUGdqa2prSDZPb2k0UGF0Yk5kR2srM3c0NzFGVm1FM3UyYaml3SnVXU  
mlvSFIVNwp2dWZxOGh5ZEtCSG1ZVUVQbnBNYXFmVmxodTBxTERhMEEvblZJNWVqNmNaUy9hc  
G13VGtlYVNHNEt4blpBRWJTCmxrMFNXWkxIT0JsVXBvSTVVb2I2ZGlVaGEzMHRXMIpPNGlLNpBR2  
dOOHRDbU56aEgyOXVQTIV6QjB5S2VBaElKclFJQkF3PT0KLS0tLS1FTkQgUFVCTEldIEtFWS0tLS0tC  
g==",  
"public_key_sign" : "c03c839bddeddfa11d6bda7e8e532fcf"  
}
```

然后由前端提取，并交给后端打包成对应字段的json格式，提交授权服务器。

4.授权服务器验证通过后，返回一串字符给后端，再交给前端，并传给算法端。然后调用当前功能的算法模型即可