

FacePass Android版MegSafe软授权操作指南 V3.3.0

1. App在启动进入主界面前，必须先进行授权激活操作；没有授权的设备，在加载人脸算法SDK时，会导致应用直接闪退。
2. 相关文件列表
 - a. SDK/libs/authapi-release.aar MegSafe软授权接口
 - b. Doc/get_authed_file.py 联网获取授权文件的脚本
 - c. Demo/TestFacePass_NewAuth MegSafe软授权使用demo，demo中的lib目录包含了人脸算法sdk包（FacePassAndroidSDK-Gemalto-release.aar）
3. authapi-release.aar接口使用说明：
 - a. 配置app工程的build.gradle 添加authapi-release.aar到当前使用人脸算法SDK的工程，如下图所示：

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    compile(name: 'FacePassAndroidSDK-Gemalto-release', ext: 'aar')  
    compile(name: 'authapi-release', ext: 'aar')
```

- b. 在进入用户程序主界面前，先进入授权页面，这步由用户自行实现。也可参考demo中的流程：SplashActivity->AuthActivity->MainActivity.
- c. 获取设备的指纹文件，接口如下：
 - i. AuthApi obj = new AuthApi();
 - ii. ReturnInfo text = new ReturnInfo();
 - iii. obj.GenC2vFile(text);
 - iv. 上面三步调用，将在/sdcard/根目录生成request.c2v文件，该文件为设备的指纹文件。参考AuthActivity模块中的generate_fingerprint函数，如下图所示：

```
private void generate_fingerprint() {  
    Log.d(TAG, msg: "generate fingerprint enter ");  
    AuthApi obj = new AuthApi();  
    ReturnInfo text = new ReturnInfo();  
    obj.GenC2vFile(text);  
  
    mAuthStatus.setTextColor(Color.BLUE);  
    mAuthStatus.setTextSize(TypedValue.COMPLEX_UNIT_SP, size: 20);  
    mAuthStatus.setBackgroundColor(Color.YELLOW);  
    mAuthStatus.setText(text.retInfo);  
  
    Log.d(TAG, text.retInfo);  
    Log.d(TAG, msg: "generate fingerprint exit ");  
}
```

- d. 联网生成授权文件
 - i. 在可以联网的pc上，运行get_authed_file.py，如下：
 - ii. 将request.c2v拷贝到该pc机上，
get_auth_file.py 里需要根据服务的具体url(修改ip地址和端口号，或者替换成域名)

```
1 url = 'https://srp.megvii.com/sdk/api/v5/authdev'
```

- iii. 单一认证方法

```
python get_auth_file.py --key 3516CV500FaceCapture-one-stage.cert --fp fingerprint.c2v
```

说明：

--key 参数：单一认证通讯文件

--pf 参数：设备指纹文件名

运行结果

成功

打印结果：get auth file ok，设备授权文件：authFile.v2c

失败

打印结果：

错误信息	原因	处理方法
detail: auth key invalid	通讯认证文件非法	到网站上重新获取通讯认证文件
--key input invalid	通讯认证文件非法	到网站上重新获取通讯认证文件
false to get auth file	授权失败	可能是指纹文件非法

4. 激活设备
- a. 将authFile.v2c修改为update.v2c; 并且放并放置在设备的/sdcard/Download目录
 - b. 调用如下接口，激活设备：
 - i. AuthApi obj = new AuthApi();
 - ii. ReturnInfo text = new ReturnInfo();
 - iii. obj.AuthDevice(path, text);
 - iv. 参考AuthActivity模块中的activate_device函数，如下：

```
private void activate_device() {
    Log.d(TAG, msg: "activate_device enter ");
    String path = "update.v2c";
    AuthApi obj = new AuthApi();
    ReturnInfo text = new ReturnInfo();
    obj.AuthDevice(path, text);

    mAuthStatus.setTextColor(Color.BLUE);
    mAuthStatus.setTextSize(TypedValue.COMPLEX_UNIT_SP, size: 20);
    mAuthStatus.setBackgroundColor(Color.YELLOW);
    mAuthStatus.setText(text.retInfo);

    Log.d(TAG, text.retInfo);
    Log.d(TAG, msg: "activate_device exit ");
}
```

- c. ReturnInfo text中返回的是设备激活状态，是一串文本描述。如果文本等于"Apply update: OK"或者"Apply update: Update already added"，表示设备授权激活成功。
5. 激活设备后，才可以进入主界面。
- 需要由用户自己去记录设备的激活状态，只有激活成功才能进入主界面，才能使用人脸sdk算法

