GeeMeeSDK Android 通讯加密集成说明文档

一． 集成方式：

采用.so 动态库的形式进行集成。将集成动态库以 armeabi 包的形式放到 libs 目录下，动态库的名字为 MdSecurity.so。

清单文件中引入权限

```
<uses-permission android:name="android.permission.INTERNET" /> <uses-permission
android:name="android.permission.READ_PHONE_STATE" /> <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
```

导入相关 jar 和 so 库

导入 jmsecurity.jar 和 libs/armeabi 目录下添加 libGeeMeeSDKBase.so 、libgnustl_shared.so、libMdSecurity.so

主 Acitivity 加载 so 库

static {

System.loadLibrary("gnustl_shared"); System.loadLibrary("MdSecurity");

}

引入 Delegate 类设置 license 权限如下：

```
Delegate dl= new Delegate(); String pagename = dl.GetPackageName(MdActivity.this);
String Appname = dl.GetAppName(this); MdSecurity.getPageAppInfo(pagename, Appname);
MdSecurity.SetLicense("BIBRFf=");//设置 license
```

功能接口

1. 获取通讯加密密文

属性名称	String getCiphertext(String plaintext,String runkey)
参数	String plaintext,runkey
备注	String plaintext 输入的明文 String runkey 表示 32 位随机数或设置为 null
返回值	String 类型，加密后的通讯数据

2. 设置 license 权限

属性名称	String SetLicense(String license)
参数	String license
备注	设置 License 开发权限
返回值	String 类型，加密后的通讯数据

3. 获取包名和应用名

属性名称	String getPageAppInfo(String packagename,String appname)
参数	String packagename,appname
备注	Packagename 表示包名 appname 表示 app 名称
返回值	无

GeeMeeSDK iOS 通讯加密集成文档

> 极密 SDK 所有版本均完美兼容 IPv6，大家可以放心使用。

注意:

- iOS 7.1+
- Xcode 7.3+
- ARC,BitCode 支持
- 建议不要与其他带有安全通信加密的 SDK 同时使用，可能影响功能的正常使用；
- 集成有问题，建议查看 FAQ，或者联系技术支持人员；

步骤

1. 下载最新 SDK 压缩包

序号	文件	说明
1	libcrypto.a	库文件
2	libJMSDKPassGuardCrypto.a	静态库文件
3	libGMCSEncrypt.a	静态库文件
4	GMCSEncrypt.h	头文件

2. 解压缩

1. 将形如 gmcs*lib*v1.x.x 的文件夹拖入工程目录
2. 确认勾选了“Copy items to destination's group folder”选项，并选择你要添加到的 Target

3. 系统依赖库配置

XCode APP 配置，`Build Phases` -> `Link Binary With Libraries` 里添加以下 `framework`：

```
Security.framework  | libstdc++6.0.9.tbd
```

4. 设置极密开放平台密码通讯加密 SDK appLicense

1. 获取 GMSDK AppLicense。如果你之前已经在极密开放平台注册了应用通讯加密 SDK，获得了 AppLicense，可以继续使用之前获得 AppLicense。
2. 如果你尚未在极密开放平台注册账号，需要先注册，注册之后登录你的账号，点击添加新应用，完成新应用填写之后，将进入应用管理页面。在该页面就能得到 AppLicense。
3. 在代码中设置你的极密 SDK 通讯加密 AppLicense，在 `AppDelegate` 文件内设置你的 AppLicense：

   如果是 Swift 项目，请在对应的 bridging-header.h 中导入

   Objective-C

```
 #import "GMSecurityStore.h"
 - (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
           //此处 license 建议 app 客户端配置参数，从服务器端动态下发

           [GMCSEncrypt setAppLicense:@"AppLicense"];

           return YES;
    }
```

5.ViewController 中调用通讯加密

1. ViewController 中引入 **GMCSEncrypt.h**
2. 开始集成开发，调用演示

   Objective-C

```
- (void)viewDidLoad {
     [super viewDidLoad];
     //通信数据加密
     NSString *enText = [[[GMCSEncrypt alloc] init]
encrypt:_encrypttext.text
rNum:@"12345678901234567890123456789012"];
     //数据解密 服务器端解密
}
```

更多使用参数说明 请参考 *iOS 极密 SDK 通讯加密代码接口说明文档*

编译运行 App，命令行数据输出，成功了！