华兴银行证书联调手册

（内部文稿）

版本号	修订内容	日期	作者
V1.0	内容修改	20160830	詹浪奇

目录

1、 证书命名规则

【发起公司】+【接受公司】作用(app/ssl)+环境(uat/prd)+后缀名(全部小写) (其中
uat 测试环境 prd 生产环境)
例如:某 xx 公司生产的证书供给银行 xx.ghb.ssl.uat.cer、xx.ghb.ssl.uat.csr、ghb.xx.ssl.uat.cer、
ghb.xx.ssl.uat.pfx

2、 SSL 证书生成

以**测试环境**为例：
公司方提供给行方的证书文件：
xx.ghb.ssl.uat.csr
行方提供给公司方的证书文件：
ghb.ssl.uat.cer、xx.ghb.ssl.uat.crt

2.1 第三方公司生成 CSR 文件

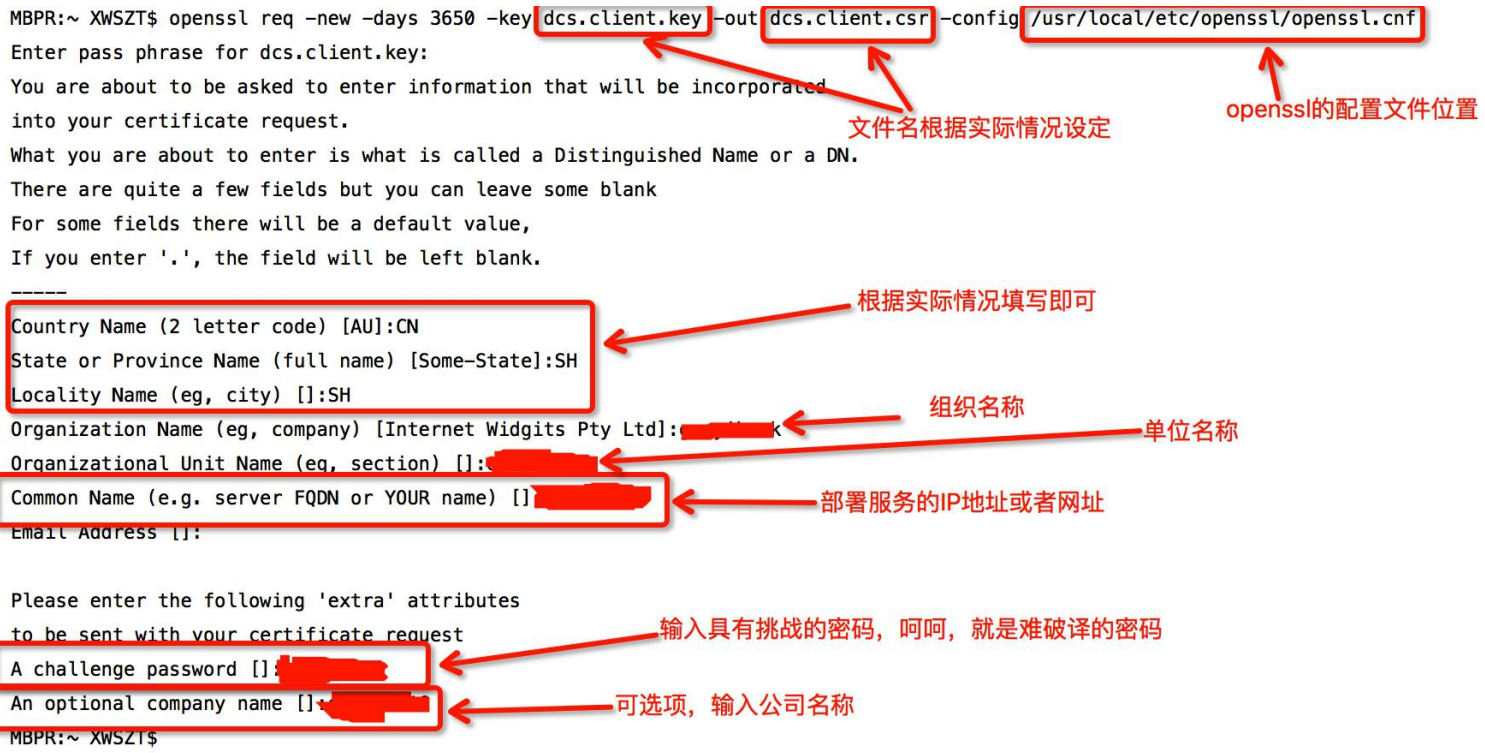
生成服务私钥（2048 长度），运行以下命令即可：

```
MBPR:~ XWSZT$ openssl genrsa -des3 -out dcs.client.key 2048
Generating RSA private key, 2048 bit long modulus
.....................................................+++
.......+++
e is 65537 (0x10001)
Enter pass phrase for dcs.client.key:
Verifying - Enter pass phrase for dcs.client.key:
```

文件名可根据自己情况设定

输入密码，记牢

生成 CSR 文件，运行以下命令即可。



可以查看生成的证书内容，命令如下：

```
MBPR:~ XWSZT$ openssl req -text -in dcs.client.csr
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=CN, ST=SH, L=SH, O=guoyibank, OU=guoyibank, CN=g
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
                Modulus (2048 bit):
                    00:ca:aa:a6:14:95:83:f4:a6:74:bd:cc:ba:b1:49:
                    af:27:22:fd:7f:91:84:55:d0:49:92:6c:28:f1:2c:
                    2e:c2:e1:59:de:7e:d0:74:75:db:71:18:9e:39:9b:
```

2.2 银行 CA 签名

根据 2.1 步骤中生成的 CSR 文件，提供给银行，由银行进行 CA 签名，并由银行返回 CRT 格式文件给第三方公司。

2.3 ssl 证书生成（私钥证书生成）

附：
cer 转换为 jks 参考命令：
keytool -import -alias ghb.btb -file ghb.btb.cer -keystore ghb.btb.jks

3、 证书验证流程

说明：这里需要进行两步操作，**首先**进行应用证书的验证，即进行数据的加密解密及验签；**其次**进行 ssl 层的证书验证，即进行 https 数据传输的证书验证。
验证步骤：

3.1 应用证书验证

步骤如下：
1）将应用证书生成的 pfx 文件及银行提供的公钥文件（.cer 文件）部署到应用系统中；
2）运行程序，对请求数据进行加密、签名，并提交数据给行方；
3）由行方进行数据的验签、解密，并判断是否验签通过。
4） 如果验签通过，则进行 3.2 的 ssl 证书验证。

3.2 ssl 证书验证

说明：由于第三方已将公钥证书提交给了行方，且行方的 https 已经 OK，这一步的验证，只需第三方公司使用相关证书配置 https 服务即可。

步骤如下：（以 Tomcat 为例）
1）准备证书：行方提供的 jks 文件（ghb.ssl.uat.jks），第三方公司生成的 pfx 文件（xx.ghb.ssl.uat.pfx）；
2）配置 tomcat 的 https 服务。 tomcat 默认情况下未开启 https 服务，需要进行重新配置。需要修改的文件位置（%Tomcat.Home%/conf/server.xml)，使用文本编辑工具打开 server.xml 文件，搜索到以下内容：

```
described in the APR documentation -->
<!--
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
           maxThreads="150" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS" />
-->
```

将注释的内容复制，并在注释下方粘贴一份新的 Connector 标签，从而进行修改。如下：

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
    maxThreads="150" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS"
    keystoreFile="keystore/[redacted]client.ca.pfx" keystorePass="[redacted]" keystoreType="PKCS12"
    truststoreFile="keystore/ghb.ssl.dep.jks" trustorePass="999999" truststoreType="JKS" />
```

说明：
属性 keystoreFile：是用来配置 keystore 的文件路径；属性 keystorePass：用来配置 keystore 的密钥；属性 keystoreType：用于配置 keystore 的类型；
属性 truststoreFile：用于配置信任证书的文件路径；属性 trustorePass:用于配置 truststore 的密钥；属性 truststoreType：用于配置 truststore 的类型；
这里的 keystoreFile 配置的是 1.3 中生成的私钥，即 pfx 文件；keystorePass 配置的是私钥的密钥；keystoreType 配置为 PKCS12（必须的，否则 tomcat 启动会报错）；truststoreFile 配置行方提供的 ssl 的 jks 文件路径；trustorePass 配置 jks 文件的密钥；truststoreType 配置为 JKS
即可。

配置的路径，可为相对路径，即相对于 tomcat 的路径；也可配置绝对路径。
上面图中的例子为相对路径，在 tomcat 下建立 keystore 目录，将两个证书文件放入其中即可。

3.3 注意事项

1、双方证书对环境需要确认部署环境所需的网络环境、硬件环境，明确各自系统、软件、开发包、工程架包等软件环境，包括各类硬件、软件各类的版本号；
2、双方在证书调试每个步骤建议自己都留下记录，保证在连调过程中及时回顾制作流程中能够及时发现问题。
3、双方在调试过程中术语保持统一，在沟通中形成一致认识，避免不必要的误解。存在的接口调试环境（测试环境）、UAT 环境（业务环境），生产环境（正式运行环境），术语不统一容易导致误会，双方术语在形成一致认识。

4、注意 QQ、微信、邮件、电话等沟通手段灵活使用。

5、出现问题时，双方及时根据制作流成确认交互内容的是否一致，流程上认识是否一致。

部署过程中，遇到网络不通的情况，需要双方协同检测，采用排除法，逐一排除问题，尤其注意网络防火墙是否打开及是否封堵了端口等等；