

CFCA 全球信任证书 (SSL 证书)

中金金融认证中心

2025 年 12 月

文档修订记录

版本	内容	日期	编写	审核
1.0	第一版	2013.3.7	张诚	
2.0	增加证书格式转换	2013.12.2	张诚	
2.1	增加应用服务证书配置	2014.1.9	张诚	
2.2	修改部分应用服务证书配置	2014.1.20	张诚	
2.3	修改证书 DN 生成规则	2014.4.16	张诚	
2.4	增加 SSLv3 禁用方式	2014.11.6	张诚	
2.5	增加 Nginx 双向 SSL 配置	2015.4.13	张诚	
3.0	完善证书介绍	2015.5.18	张诚	
3.1	完善证书介绍	2015.7.7	张诚	
3.2	完善证书介绍	2015.7.13	张诚	
4.0	完善证书介绍、办理流程等	2015.12.3	张诚	
4.1	增加域名或者公网 IP 的证明示例和常见问题	2016.2.5	胡俊燕	
4.2	修改服务器证书支持范围（增加 MAC OS 和 IOS 系统），根据业务部的 新申请表更新本文申请表	2016.10.28	王瑞萍	
4.3	Weblogic 证书配置章节增加内容	2016.11.16	王瑞萍	
4.4	更新全球服务器证书支持范围	2017.5.7	王天昊	
4.5	完善 SSL 证书部署	2018.1.1	王天昊	
4.6	删除产品介绍，增加阿里云、腾讯云、金蝶证书部署，增加 Kyr 格式转换，细化 tomcat 证书部署，增加域名验证方式	2020.2.1	张彪	毕鑫龙、仇大伟
4.7	补充证书更新说明内容	2020.4.22	陈超	施威
5.0	更新证书申请、更新等说明内容	2024.11.28	赵莎	

5.1	修改生成 CSR 内容	2025.12	郑晓娟	
-----	-------------	---------	-----	--

目 录

一、证书办理说明	1
1.1 申请材料说明	1
1.2 审核说明	1
1.3 证书签发	2
1.4 证书更新、延期、吊销	2
二、CFCA 全球信任 SSL 证书制作	2
2.1 证书制作说明	2
2.2 密钥和证书请求文件 CSR	3
2.3 证书文件格式	3
2.4 证书制作	4
2.4.1 使用 Keytool 工具制作证书	4
2.4.2 使用 OpenSSL 工具制作证书	7
2.4.3 使用 iKeyman 工具制作证书	9
2.5 证书格式转换	13
2.5.1 工具转换（优先推荐）	14
2.5.2 JKS 转换为 PFX	14
2.5.3 PFX 转换为 JKS	14
2.5.4 KEY&CRT 转换为 PFX	14
2.5.5 PFX 转换为 KEY&CRT	14
2.5.6 KDB 转换为 PFX	15
2.5.7 PFX 转换为 KDB	16
2.5.8 KYR 格式证书制作	20
2.6 证书部署方法	22
2.6.1 Apache 证书配置	22
2.6.2 Tomcat 证书配置	23
2.6.3 Nginx 证书配置	25
2.6.4 Weblogic 证书配置	28
2.6.5 IBM Http Server 证书配置	31
2.6.6 JBoss 证书配置	31
2.6.7 IIS 证书配置	32
2.6.8 Websphere 证书配置	38
2.6.9 IHS+WAS 证书配置	46
2.6.10 F5 设备证书配置	47
2.6.11 SAP 证书配置	49
2.6.12 阿里云通用证书配置	54
2.6.13 阿里云 LSB 证书配置	56
2.6.14 腾讯云证书配置	59
2.6.15 金蝶 Apusic 证书配置 1	61
2.6.16 金蝶 Apusic 证书配置 2	62

附录一、CFCA 全球信任证书（SSL 证书）申请表及授权书	65
附录二、CFCA 域名验证方式	68
附录三、CFCA 全球信任根证书获取方式	73
附录四、CFCA 全球信任证书链	74
附录五、SHA 摘要算法介绍	79
附录六、常见问题	80

一、证书办理说明

CFCA 全球信任 SSL 证书办理，申请机构必须提供真实的材料，以证明机构的真实身份、申请人的真实身份、机构对域名的控制权等。CFCA 将对机构提供的材料进行严格审查。

1.1 申请材料说明

申请机构需要向 CFCA 提供如下材料：

- ① CSR
- ② 《CFCA 全球信任服务器证书申请表》

申请材料说明：

1、通过 CloudPKI 平台进行 CFCA SSL 证书申请时，无需提供《CFCA 全球信任服务器证书申请表》。

2、《CFCA 全球信任服务器证书申请表》可通过 CFCA 官网（<https://www.cfca.com.cn>）获取最新版。首页--下载专区--申请表下载--全球信任服务器证书申请表。

3、《CFCA 全球信任服务器证书申请表》无须加盖公章，CFCA 将采用电话验证方式（申请机构工商注册电话等），联系申请机构进行申请信息及意愿验证。电话验证不成功时，需提供加盖公章的《CFCA 全球信任服务器证书申请表》，模板为[附录一](#)。

4、申请表中机构信息、经办人信息验证不成功时，需提供机构证件复印件及经办人身份证复印件。

5、当域名所有人与申请人不一致时，需要域名所有人对代理申请人进行授权，提供证书申请授权证明。授权书可由域名所有人手写签名，无需盖章。模板为[附录一](#)。

1.2 审核说明

CFCA 将对申请机构提供的材料进行审查，主要包括：

- 1、检查证书申请表中机构信息和经办人信息真实性与有效性
- 2、检查证书申请人代表机构申请证书的真实性与有效性。
- 3、检查申请者对证书申请表中域名控制权。

4、CSR 文件，DN 规则要求符合如下规范：

- (1) DN 中各项顺序依次为：CN、O、L、ST、C；
- (2) CN 项是域名或者 IP，与证书申请表中域名一致；
- (3) O 项必须是真实的、完整的机构名称，与证书申请表中机构名称一致；
- (4) L 项、ST 项、C 项是必须是机构注册地，与机构注册地区一致。

1.3 证书签发

CFCA 审核通过后，申请者将收到证书公钥及证书链。

1.4 证书更新、延期、吊销

使用证书过程中，如果出现证书遗失、损坏、密钥泄露等问题，需要重新签发证书，机构应按照 1.1 章节重新提供材料办理证书。证书有效期内 CFCA 免费进行证书操作。

证书到期前一个月内，CFCA 会主动提醒机构申请联系人办理证书延期。

OV SSL 证书、EV SSL 证书换发操作成功时，已过期的证书换发不吊销老证书，按照新申请处理。新证书有效期将从证书换发之日起至原证书到期为止再另加一个证书有效周期（已经过期的证书换证，其有效期仅为新证书有效周期）。

机构如果不再使用证书，可以联系 CFCA 办理证书吊销，并将该证书从网站服务器上移除。

二、CFCA 全球信任 SSL 证书制作

2.1 证书制作说明

CFCA 全球信任 SSL 证书需要部署在网站 Web 服务上，制作证书之前请先了解网站 Web 服务所使用的软件或者硬件。

1、如果网站使用网关、负载均衡等硬件设备提供 Web 服务，请咨询相应的硬件厂商制作和部署 SSL 证书的方法。

2、如果网站使用 IBM Http Server (IHS)、Internet Information Services (IIS)、Oracle Weblogic 等商业软件提供 Web 服务，请优先咨询软件厂商制作和部署 SSL 证书的方法。本章节也提供了部分商业软件制作 SSL 证书方法，仅供参考。

3、如果网站使用 Nginx、Apache、Tomcat 等开源软件，请优先通过开源软

件的技术文档了解制作和部署 SSL 证书的方法。本章节也提供了部分开源软件制作 SSL 证书方法，仅供参考。

4、部署 SSL 证书时，必须部署相应的证书链。办理 CFCA EV SSL 证书，需要部署根证书 CFCA EV Root 和中级证书 CFCA EV OCA；办理 CFCA OV SSL 证书，需要部署根证书 CFCA EV Root 和中级证书 CFCA OV OCA。证书链文件会与证书文件一起发送到证书申请表中的申请人邮箱。关于证书链的详细信息可参考“[附录四、CFCA 全球信任证书链](#)”。

2.2 密钥和证书请求文件 CSR

SSL 证书中含有一套非对称密钥，用于客户端浏览器和网站服务器之间的数据加密。证书申请机构应当在安全的服务器或者设备上生成密钥和证书请求文件 CSR（Certificate Signing Request）。其中，CSR 提交给 CFCA 用于签发证书（详见 1.1 章节），**密钥由证书申请机构保管**。

特别提醒，任何机构或者个人如果拥有该密钥，即可通过技术手段解密客户端浏览器和网站服务器之间的加密数据，给网站和网站用户造成极大的安全威胁。因此，证书申请机构务必妥善保管密钥！一旦密钥泄露，请证书申请机构重新生成密钥和 CSR，并立即联系 CFCA 进行证书更新（参考 1.4 章节）。CFCA 将使用新提供的 CSR 签发证书，并将原证书吊销。

2.3 证书文件格式

一般来说，主流的 Web 服务软件，通常都基于两种基础密码库：OpenSSL 和 Java。

Tomcat、Weblogic、JBoss 等，使用 Java 提供的密码库。通过 Java 的 Keytool 工具，生成 Java Keystore（JKS）格式的证书文件。

Apache、Nginx 等，使用 OpenSSL 提供的密码库，生成 PEM、KEY、CRT 等格式的证书文件。

此外，IBM 的产品，如 Websphere、IBM Http Server（IHS）等，使用 IBM 产品自带的 iKeyman 工具，生成 KDB 格式的证书文件。微软 Windows Server 中的 Internet Information Services（IIS），使用 Windows 自带的证书库生成 PFX 格式的证书文件。

2.4 章节提供了部分主流的 Web 服务软件生成密钥、CSR、证书文件的方法，

2.5 章节提供了常用证书文件格式相互转换的方法，仅供参考。

2.4 证书制作

2.4.1 使用 Keytool 工具制作证书

Keytool 是 JDK 中自带的密钥管理工具，可以制作 Keystore (jks) 格式的证书文件。下载并安装 JDK 后，可以通过相关命令制作服务器证书。

以下地址可以下载 JDK：

<https://www.oracle.com/java/technologies/>

以下以 Windows 平台为例，介绍制作证书的方法。

1、进入 Keytool 目录：

```
cd C:\Program Files\java\jdk1.6.0_39\bin
```

2、生成证书文件 Keystore，文件后缀名可以是 jks、keystore，证书文件中包含密钥；

```
keytool -genkey -alias server -keyalg RSA -keysize 2048 -keystore D:\server.jks
```

其中，keyalg 是密钥类型，必须为 RSA；keysize 是密钥长度，必须是 2048，alias 是证书别名，可自定义；keystore 是证书文件保存的路径。

而后，输入证书文件的密码：

输入 keystore 密码：

再次输入新密码：

而后，输入名称（CN），即证书申请表中的域名；

您的名字与姓氏是什么？

[Unknown]: www.cfca.com.cn

而后，输入组织单位（OU），即证书申请表中申请人的部门名称；

您的组织单位名称是什么？

[Unknown]: 技术支持部

而后，输入组织（O），即机构身份证件中机构名称全称；

您的组织名称是什么？

[Unknown]: 中金金融认证中心有限公司

而后，输入城市（L），即机构身份证件中机构所在市级地区；

您所在的城市或区域名称是什么？

[Unknown]: 北京

而后，输入省份（ST），即机构身份证件中机构所在省级地区；

您所在的州或省份名称是什么？

[Unknown]: 北京

而后，输入机构身份证件中机构所在的国家或者行政区（C），限定两位字母，如中国输入 CN，美国输入 US 等；

该单位的两字母国家代码是什么？

[Unknown]: CN

输入完成后，确认输入内容是否正确；

CN=www.cfca.com.cn, OU=技术支持部, O=中金金融认证中心有限公司, L=北京, ST=北京, C=CN 正确吗？

[否]: Y

而后，提示输入密钥（Key）密码，可以与证书（Keystore）密码一致；

输入<server>的主密码

（如果和 keystore 密码相同，按回车）：

确认后，即在 keystore 保存的路径下，生成证书文件（server.jks）。

3、通过证书文件，生成证书请求；

```
keytool -certreq -sigalg SHA256withRSA -alias server -keystore d:\server.jks -file d:\certreq.csr
```

其中，sigalg 是摘要算法，推荐 SHA256withRSA；alias 是别名，必须与第 2 步生成证书文件时定义的别名一致；keystore 是证书文件的路径，file 是产生证书请求（CSR）的路径。

而后，提示输入 keystore 的密码；

输入 keystore 密码：

确认后，即产生证书请求（CSR）文件（certreq.csr）。

4、证书申请机构将证书请求文件（certreq.csr）连同相关材料（详见 2.1）提供给 CFCA，并妥善保管证书文件（server.jks）。

5、CFCA 审核资料后，将公钥证书和证书链反馈给证书申请机构。

6、证书申请机构将收到的公钥证书和证书链（包括根证书和中级证书）装

回到证书文件（server.jks）中。

其中，证书文件一般以申请单位全称命名；EV SSL 证书的根证书是 CFCA_EV_CA.cer；中级证书是 CFCA_EV_OCA.cer；OV SSL 证书的根证书是 CFCA_EV_CA.cer；中级证书是 CFCA_OV_OCA.cer（详见附录四）。

7、导入根证书（以 EV SSL 证书为例，OV SSL 证书请导入 OV 的根证书，下同）；

```
keytool -import -alias evca -keystore d:\server.jks -trustcacerts -file d:\CFCA_EV_CA.cer
```

而后，输入证书文件密码；

输入 keystore 密码：

而后，会显示根证书的属性；

```
所有者: CN=CFCA EV ROOT, O=China Financial Certification Authority, C=CN
发布者: CN=CFCA EV ROOT, O=China Financial Certification Authority, C=CN
序列号: 184accd6
有效期开始日期: Wed Aug 08 11:07:01 CST 2012, 截止日期: Mon Dec 31 11:07:01 CST 2029
证书指纹:

    MD5: 74:E1:B6:ED:26:7A:7A:44:30:33:94:AB:7B:27:81:30

    SHA1: E2:B8:29:4B:55:84:AB:6B:58:C2:90:46:6C:AC:3F:B8:39:8F:84:83

    SHA256:

5C:C3:D7:8E:4E:1D:5E:45:54:7A:04:E6:87:3E:64:F9:0C:F9:53:6D:1C:CC:2E:F8:00:F3:55:C4:C5:FD:7
0:FD

    签名算法名称: SHA256withRSA

.....
```

而后，确认信任认证，导入完成。

信任这个认证？ [否]: Y

认证已添加至 keystore 中

8、导入中级证书；

```
keytool -import -alias evoca -keystore d:\server.jks -trustcacerts -file d:\CFCA_EV_OCA.cer
```

而后，输入证书文件密码；

输入 keystore 密码：

导入完成。

认证已添加至 keystore 中

9、导入服务器证书（证书文件一般以申请单位的全称命名）；

```
keytool -import -alias server -keystore d:\server.jks -trustcacerts -file d:\中金金融认证中心有限公司.cer
```

其中，别名（alias）必须是生成证书文件时设置的别名，必须与第 2 步生成证书文件时定义的别名一致；

而后，输入证书文件密码；

输入 keystore 密码：

导入完成。

认证已添加至 keystore 中

完成上述操作后，生成完整的证书文件（server.jks），可以部署在 Tomcat、Weblogic 等 Web 应用中。

2.4.2 使用 OpenSSL 工具制作证书

使用 OpenSSL 工具可以制作 KEY 和 CRT 格式的证书文件，OpenSSL 工具可以从以下地址下载：

<http://www.openssl.org/>

以下以 Windows 平台为例，介绍制作证书的方法。

1、进入 OpenSSL 目录；

```
cd D:\OpenSSL\bin
```

2、生成证书文件 key 和证书请求文件 csr；

```
openssl req -new -newkey rsa:2048 -nodes -keyout server.key -out server.csr
```

其中，newkey 必须是 rsa:2048，key 为密钥文件，csr 为证书请求文件，默认都在 OpenSSL 目录下；

```
Generating a 2048 bit RSA private key
```

```
.....+++
```

```
.....+++
```

```
writing new private key to 'server.key'
```

```
-----
```

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

而后，输入机构身份证件中机构所在的国家或者行政区（C），限定两位字母，如中国输入 CN，美国输入 US 等；

Country Name (2 letter code) [AU]:CN

而后，输入省份（ST），即机构身份证件中机构所在省级地区；

State or Province Name (full name) [Some-State]:Beijing

而后，输入城市（L），即机构身份证件中机构所在市级地区；

Locality Name (eg, city) []:Beijing

而后，输入组织（O），即机构身份证件中机构名称全称；

Organization Name (eg, company) [Internet Widgits Pty Ltd]:China Financial Certification Authority

而后，输入组织单位（OU），即证书申请表中申请人的部门名称；

Organizational Unit Name (eg, section) []:Technology Department

而后，输入名称（CN），即证书申请表中的域名；

Common Name (e.g. server FQDN or YOUR name) []:www.cfca.com.cn

而后，以下几项均可不填写；

Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request

A challenge password []:

An optional company name []:

Please enter the following 'extra' attributes

而后，将在 OpenSSL 目录下，产生证书文件 key 和证书请求文件 csr。

3、证书申请机构将证书请求文件（server.csr）连同相关材料（详见 2.1 章节）提供给 CFCA，并妥善保管密钥文件（server.key）。

4、CFCA 审核资料后，将公钥证书和证书链反馈给申请机构。

其中，证书文件一般以申请单位全称命名；EV SSL 证书的根证书是 CFCA_EV_CA.cer；中级证书是 CFCA_EV_OCA.cer；OV SSL 证书的根证书是 CFCA_EV_CA.cer；中级证书是 CFCA_OV_OCA.cer。

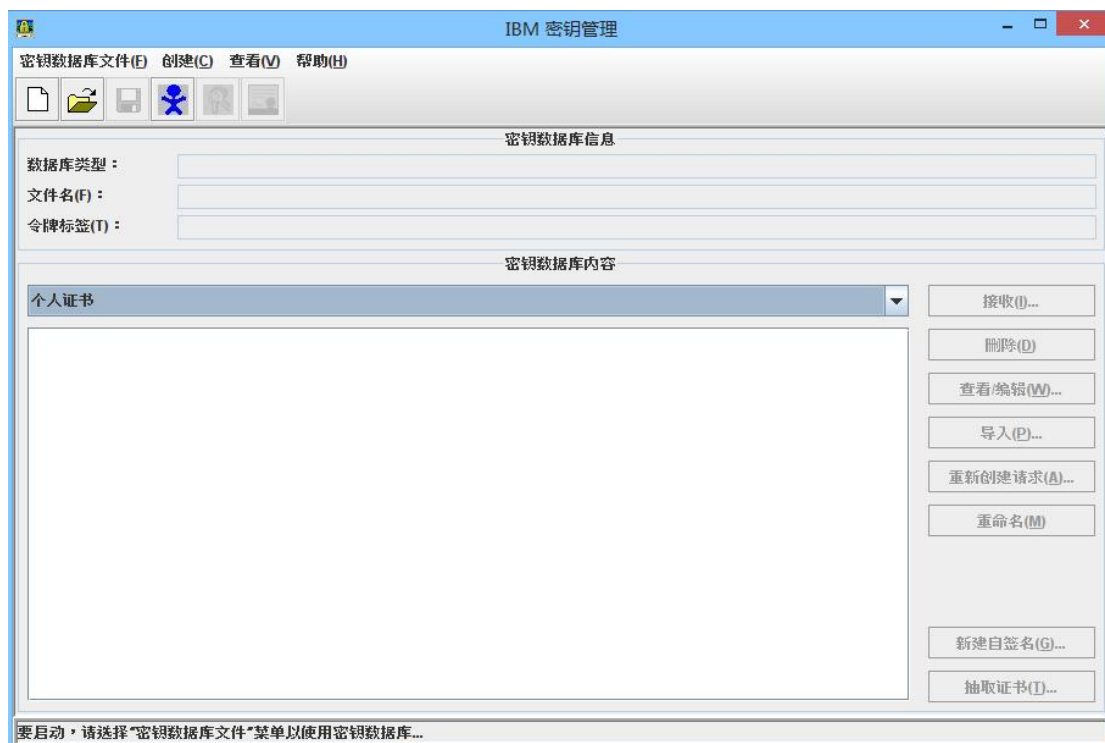
5、将服务器证书公钥另存为 server.crt。

完成上述操作后，server.key 为密钥文件、server.crt 为服务器证书文件，和证书链文件一起，可以部署在 Apache、Nginx 等 Web 应用中。

2.4.3 使用 iKeyman 工具制作证书

IBM HTTP Server 含有，可以制作证书。

1、执行 IHS7 安装目录下，“bin”目录下的“ikeyman”命令，进入 iKeyman 界面。



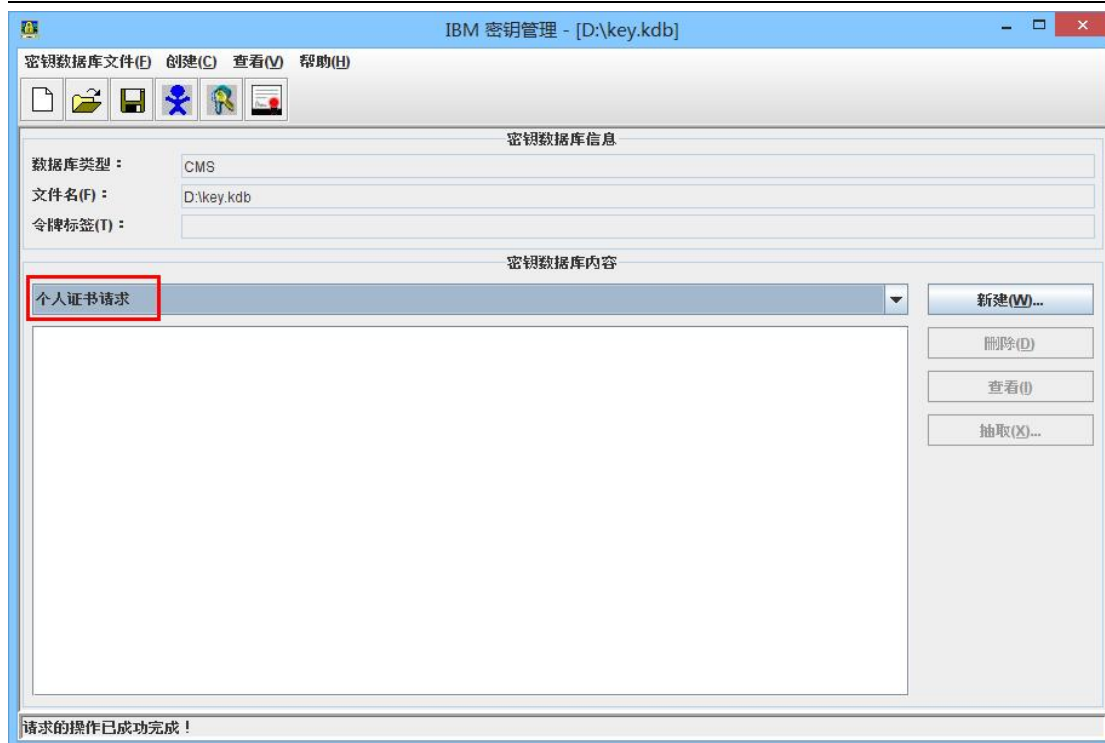
2、选择“密钥数据库文件——新建”，弹出以下对话框。



3、在密钥数据库类型中选择“CMS”。点击“浏览”选择密钥数据库文件所在路径，默认情况下应该在执行 ikeyman 的 bin 目录下。点击确定进入如下界面：



4、输入密钥数据库的密码，选择“将密码存储到文件中”点击“确定”进入如下界面，同时生成密码存储文件“key.sth”。该密码存储文件必须和密码数据库文件放在同一目录下。



5、切换到“个人证书请求”，选择界面上方的“创建——新建证书请求”进入如下界面。其中：

密钥大小必须为 2048；

公用名（CN），即证书申请表中的域名；

组织（O），即机构身份证件中机构名称全称；

组织单元（OU），即证书申请表中申请人的部门名称；

市、县、区（L），即机构身份证件中机构所在市级地区；

省、直辖市（ST），即机构身份证件中机构所在省级地区；

国家或地区（C），输入机构身份证件中机构所在的国家或者行政区，限定两位字母，如中国输入 CN，美国输入 US 等。

6、点击确定后，即在该路径下生成证书请求文件 certreq.arm。证书申请机构将证书请求文件提供给 CFCA，并妥善保管证书文件（key.kdb）和证书密码文件（key.sth）。

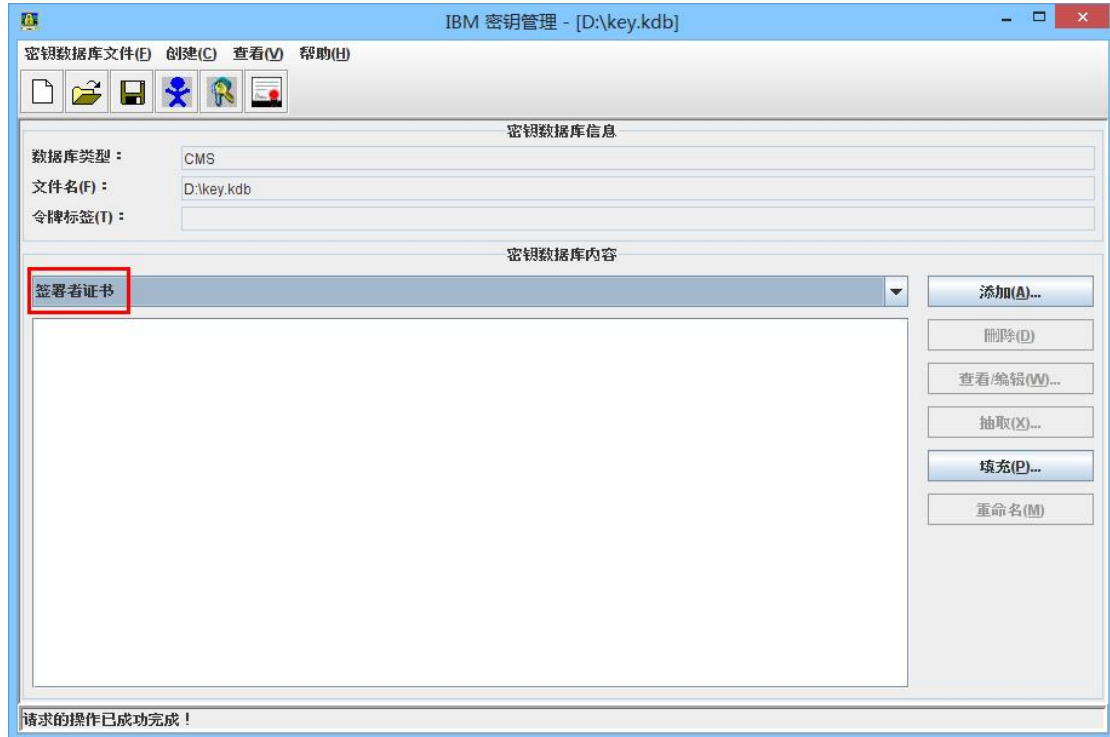
7、CFCA 审核资料后，将公钥证书和证书链反馈给申请机构。

8、证书申请机构将收到的公钥证书和证书链装回到证书文件（key.kdb）中。

其中，证书文件一般以申请单位全称命名；EV SSL 证书的根证书是 CFCA_EV_CA.cer；中级证书是 CFCA_EV_OCA.cer；OV SSL 证书的根证书是

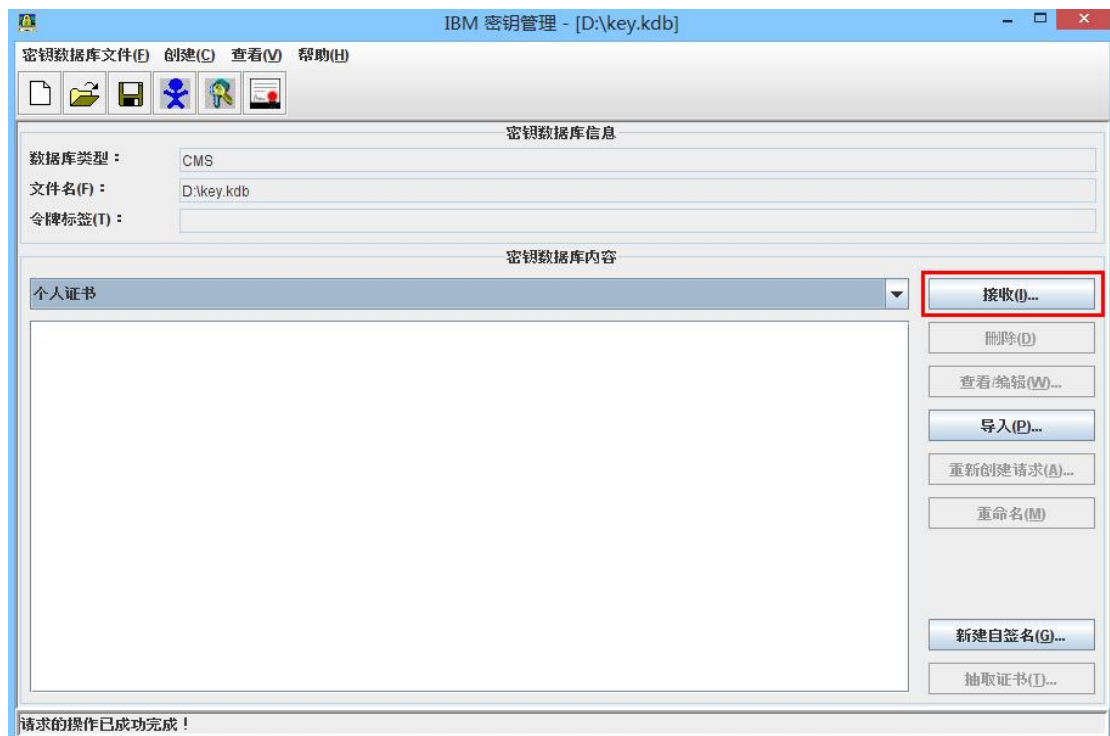
CFCA_EV_CA.cer；中级证书是 CFCA_OV_OCA.cer。

9、选择“签署者证书”，进入如下界面。

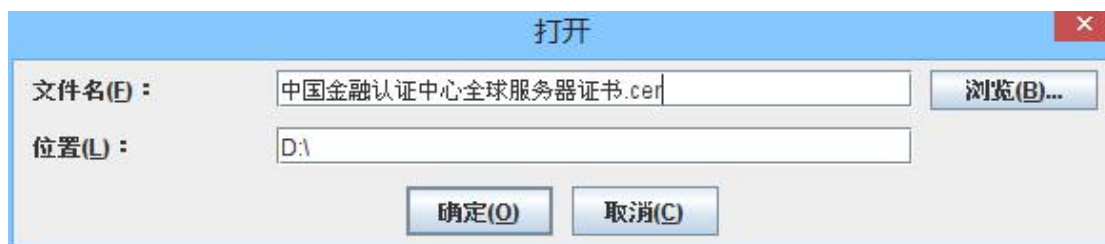


10、选择“添加”，弹出选择证书对话框，选择根证书，点击确定，导入完成。同样，将中级证书也导入到 key.kdb 中。

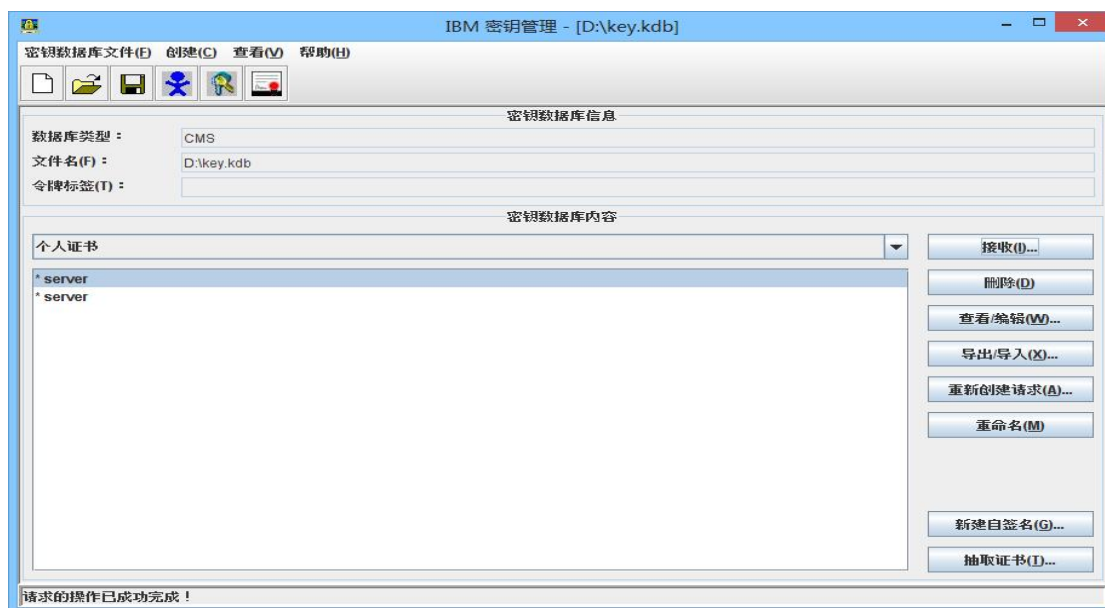
11、在密钥数据库中选择“个人证书”，进入下面的界面。



14、点击“接收”，弹出如下对话框。



15、选择服务器证书公钥文件，点击“确定”，导入完成。



2.5 证书格式转换

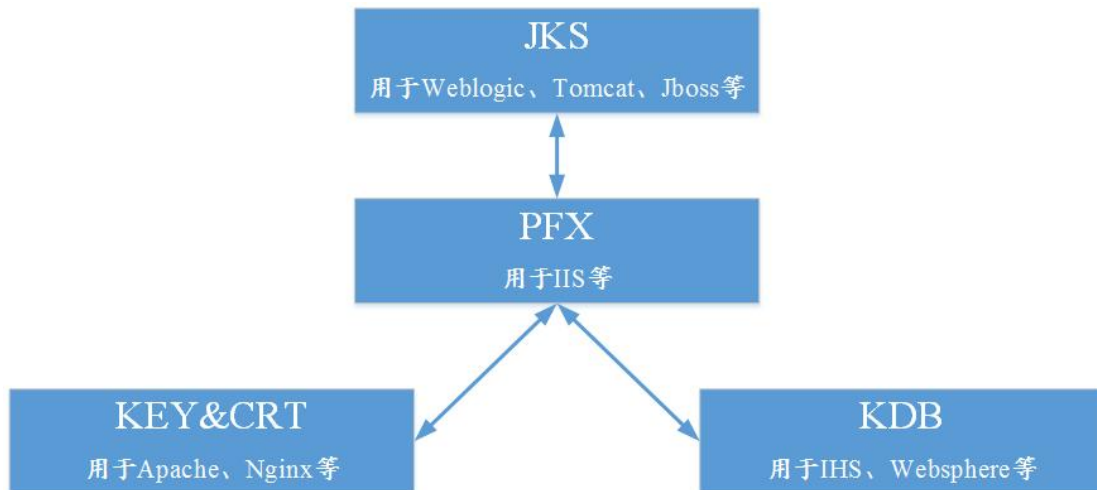
Tomcat、Weblogic、JBoss 等，使用 Java Keystore（JKS）格式的证书文件；

Apache、Nginx 等，使用 KEY、CRT 格式的证书文件；

IBM Websphere、IBM Http Server 等，使用 KDB 格式的证书文件；

IIS 等，使用 PFX（P12）格式的证书文件；

JKS、KEY&CRT、KDB、PFX 等格式的证书文件可以相互转换。如下图所示：



2.5.1 工具转换（优先推荐）

CFCA 提供离线的证书格式转换工具，可满足大部分需求场景。

离线工具：



SSLCertTool.exe

2.5.2 JKS 转换为 PFX

可以使用 Keytool 工具，将 JKS 格式转换为 PFX 格式。

```
keytool -importkeystore -srckeystore D:\server.jks -destkeystore D:\server.pfx  
-srcstoretype JKS -deststoretype PKCS12
```

2.5.3 PFX 转换为 JKS

可以使用 Keytool 工具，将 PFX 格式转换为 JKS 格式。

```
keytool -importkeystore -srckeystore D:\server.pfx -destkeystore D:\server.jks  
-srcstoretype PKCS12 -deststoretype JKS
```

2.5.4 KEY&CRT 转换为 PFX

使用 OpenSSL 工具，可以将密钥文件 KEY 和公钥文件 CRT 转化为 PFX 文件。

将密钥文件 KEY 和公钥文件 CRT 放到 OpenSSL 目录下，打开 OpenSSL 执行以下命令：

```
openssl pkcs12 -export -out server.pfx -inkey server.key -in server.crt
```

2.5.5 PFX 转换为 KEY&CRT

使用 OpenSSL 工具，可以将 PFX 文件转化为密钥文件 KEY 和公钥文件 CRT。

将 PFX 文件放到 OpenSSL 目录下，打开 OpenSSL 执行以下命令：

```
openssl pkcs12 -in server.pfx -nodes -out server.pem
```

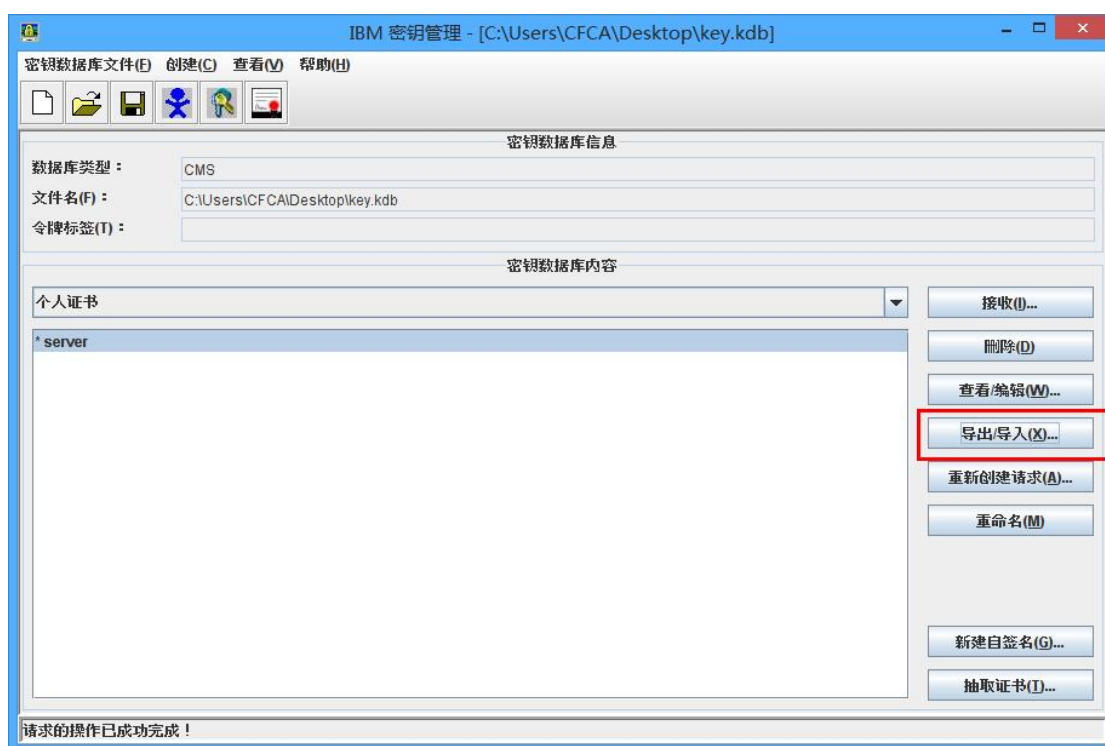
```
openssl rsa -in server.pem -out server.key
```

```
openssl x509 -in server.pem -out server.crt
```

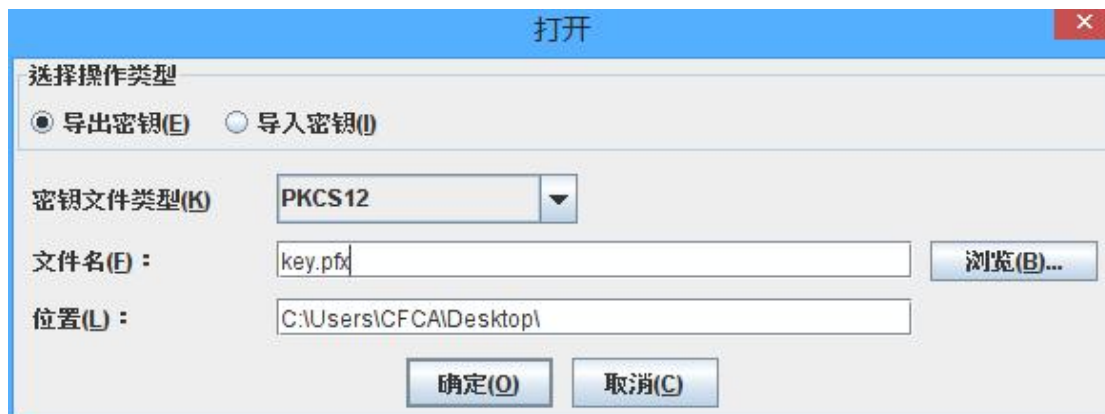
2.5.6 KDB 转换为 PFX

使用 iKeyman 工具，可以将 KDB 文件转化为 PFX 文件。

打开 KDB 文件，点击“导出”按钮。



选择“导出密钥”，选择 PKCS12 格式。



设置 PFX 密码。

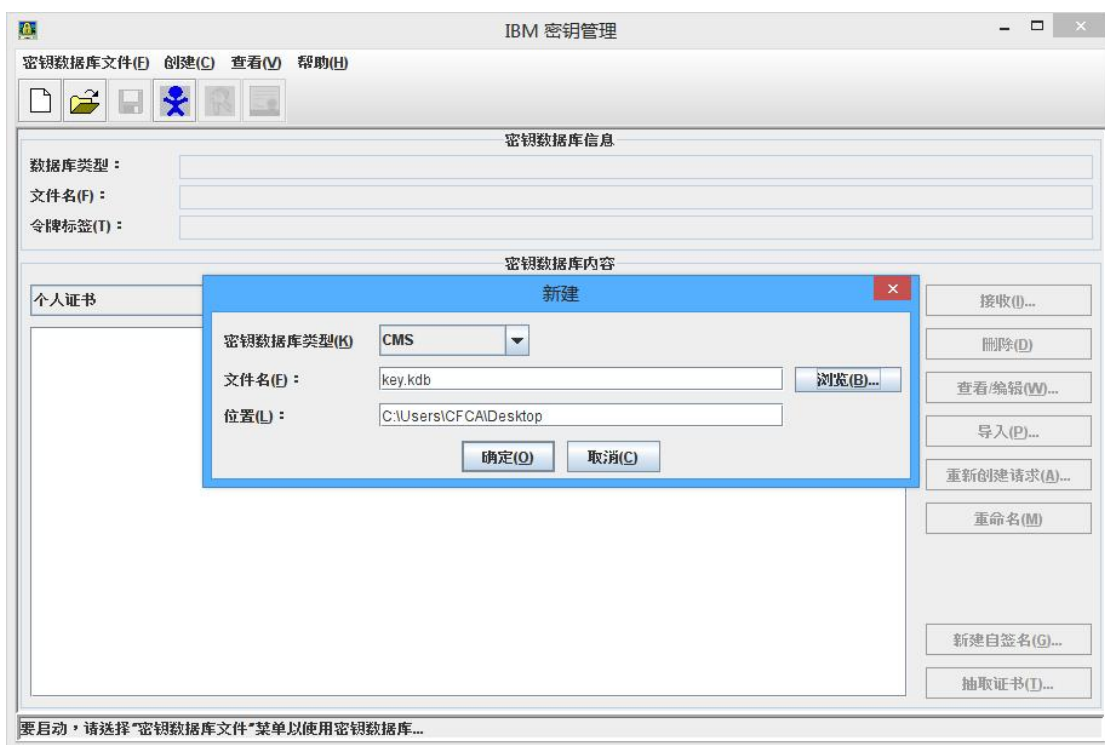


即可导出 PFX 文件。

2.5.7 PFX 转换为 KDB

使用 iKeyman 工具，可以将 PFX 文件转化为 KDB 文件。

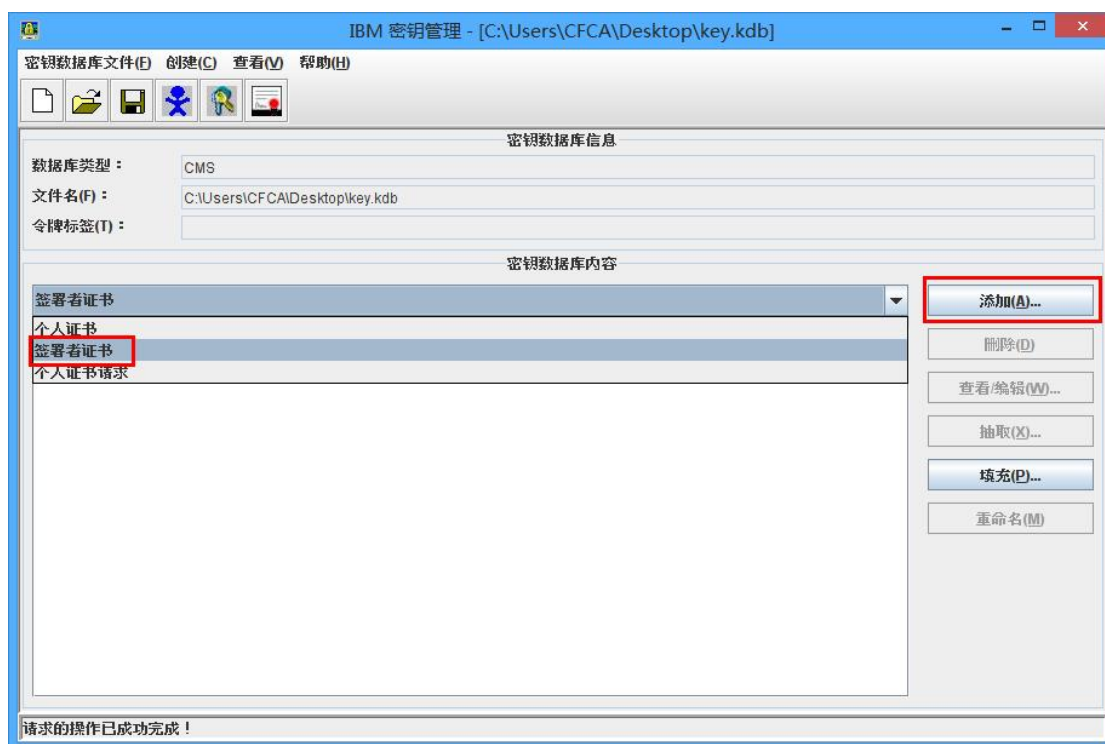
打开 iKeyman，新建一个 KDB 文件。



输入密码，可将密码存储到文件中。

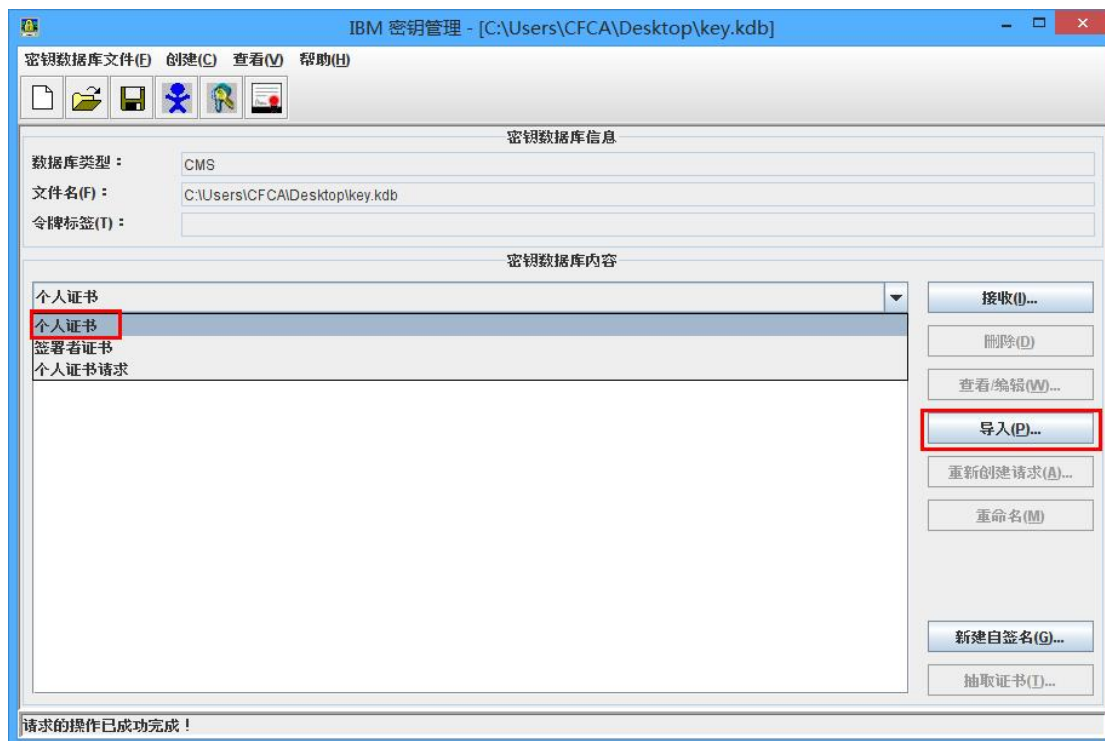


选择签署者证书，点击“添加”按钮。

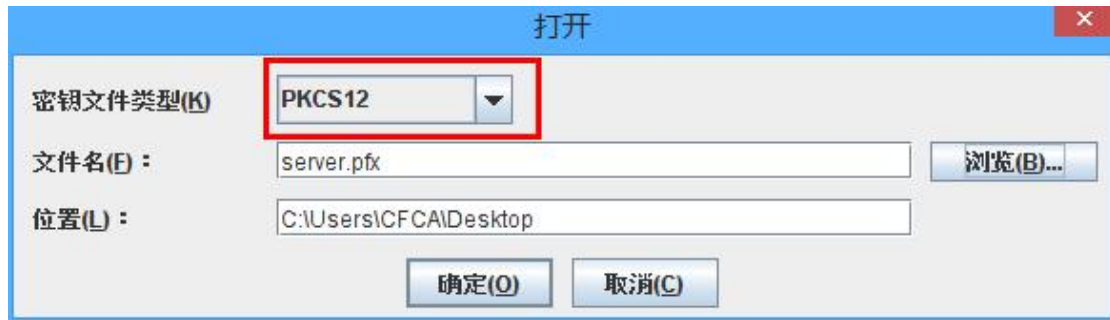


依次将根证书和中级证书导入。

选择个人证书，点击“导入”按钮。



选择 PKCS12 类型，选择 PFX 文件。



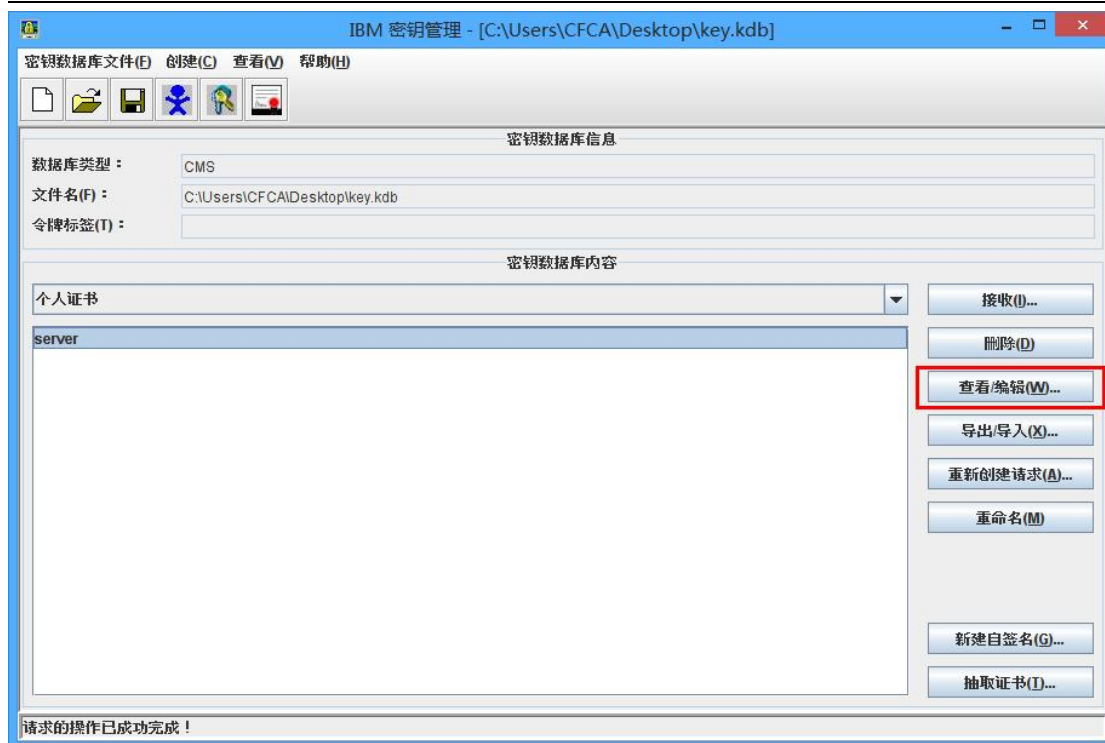
输入 PFX 密码。



输入标签名称。



导入成功后，查看个人证书。



证书信息中，勾选“将此证书设置为缺省证书”。



而后保存 KDB 文件即可。

2.5.8 KYR 格式证书制作

KYR 格式证书制作，依赖 IBM Domino Server 软件，使用 KyrTool 命令行工具进行操作。另需要 IBM Domino Server 升级到 9.0.1FP3 及以上，方能支持 sha256 算法证书。

由于 IBM Domino Server 为付费软件，故 KYR 格式证书需在用户环境中完成转换，本文档提供转换方法：

准备工作：

- 1、 已将服务器升级到 9.0.1 以上版本并成功安装 Kyrtool 的命令行工具



kyrtool.zip （将 kyrtool 工具放入 C:\Program Files\IBM\Domino 目录）

操作步骤：

- 1、 进入 Domino 目录

```
cd C:\Program Files\IBM\Domino
```

- 2、 检查 KyrTool 工具是否已就绪

```
kyrtool -h
```

KyrTool 就绪示例

```
C:\Program Files\IBM\Domino>kyrtool -h
KyrTool v1.0

kyrtool [=path/to/notes.ini] command [subcommand] [flags]

Commands:
  create          Create a new keyring file
  delete          Delete a root in a keyring file
  import          Import into a keyring file
  show            Show information about a keyring file
  verify <path>   Verify the content of a PEM import file

Use 'kyrtool [command] -h' to view help for each command.

The keyring password is stored in the STH file and will be
automatically read when using an existing keyring file.
```

- 3、 创建 kyr 文件，并设置密码

```
kyrtool create -k "D:\server.kyr" -p password
```

- 4、 将私钥文件，证书，中级证书，根证书，按照自上至下的顺序，合并为 server.txt 文件，完成后可通过以下命令检查是否正确

```
kyrtool =notes.ini verify D:\server.txt
```

检查结果


```
C:\Program Files\IBM\Domino>kyrtool =notes.ini verify D:\server.txt  
  
KyrTool v1.0  
  
Successfully read 2048 bit RSA private key  
INFO: Successfully read 3 certificates  
INFO: Private key matches leaf certificate  
INFO: IssuerName of cert 0 matches the SubjectName of cert 1  
INFO: IssuerName of cert 1 matches the SubjectName of cert 2  
INFO: Final certificate in chain is self-signed
```

5、生成最终证书文件

```
kyrtool =notes.ini import all -k D:\server.kyr -i D:\server.txt
```

2.6 证书部署方法

证书部署方式，请优先咨询提供 Web 应用软件的软件或者硬件厂商，本章节提供了部分 Web 应用软件部署证书的方式，仅供参考。

2.6.1 Apache 证书配置

Apache 使用 KEY 和 CRT 格式的证书，证书制作方式请参考“2.4.3 使用 OpenSSL 工具制作证书”。

将中级证书和根证书打开，依次将其代码复制到文本文件中（包括“-----BEGIN CERTIFICATE-----”和“-----END CERTIFICATE-----”），并保存成 cfca.crt。如下：

```
-----BEGIN CERTIFICATE-----  
中级证书编码  
-----END CERTIFICATE-----  
  
-----BEGIN CERTIFICATE-----  
根证书编码  
-----END CERTIFICATE-----
```

将服务器证书文件 server.key 和 server.crt，以及证书链文件 cfca.crt，配置在 Apache 中。

用文本编辑器打开 Apache 根目录下的 conf/httpd.conf 文件，去掉下述两行的注释符号#。

```
#LoadModule ssl_module modules/mod_ssl.so  
  
#Include conf/extra/httpd-ssl.conf
```

用文本编辑器打开 Apache 根目录下的 conf/extra/httpd-ssl.conf 文件，修改以下内容：

```
<VirtualHost 127.0.0.1:443>

    DocumentRoot "/var/www/html"

    ServerName

    SSLEngine on

    SSLProtocol all -SSLv2 -SSLv3

    SSLCertificateFile server.crt 路径

    SSLCertificateKeyFile server.key 路径

    SSLCertificateChainFile cfca.crt 路径

</VirtualHost>
```

其中：

禁用 SSLv2、SSLv3 协议：SSLProtocol all -SSLv2 -SSLv3

公钥文件：SSLCertificateFile server.crt 路径

私钥文件：SSLCertificateKeyFile server.key 路径

证书链文件：SSLCertificateChainFile cfca.crt 路径

上述设置完成过后，重新启动 Apache。

可选： 设置 HTTP 请求自动跳转 HTTPS

修改 httpd.conf 文件

在 httpd.conf 文件中的<VirtualHost *:80> </VirtualHost>中间，添加以下重定向代码。

```
RewriteEngine on
RewriteCond %{SERVER_PORT} !^443$
RewriteRule ^(.*)$ https://%{SERVER_NAME}$1 [L,R]
```

2.6.2 Tomcat 证书配置

Tomcat 使用 JKS 格式的证书，证书制作方式请参考“2.4.3 使用 Keytool 工具制作证书”。

将服务器证书文件（server.jks），配置在 Tomcat 中。

文本编辑器打开 Tomcat 安装目录下 conf 目录中的 server.xml 文件，更新以下内容。

Tomcat8.5 以下配置

```
<Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"
    maxThreads="150" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS"
    keystoreFile="jks 路径"
    keystorePass="jks 密码"
    truststoreFile="jks 路径"
    truststorePass="jks 密码" />
```

其中：

SSL 访问端口：port="443"

禁用 SSLv2、SSLv3 协议：sslProtocol="TLS"

证书文件：keystoreFile="jks 路径"

证书密码：keystorePass="jks 密码"

信任证书链文件：truststoreFile="jks 路径"

信任证书链密码：truststorePass="jks 密码"

Tomcat8.5 及以上配置：

```
<Connector port="443" protocol="org.apache.coyote.http11.Http11NioProtocol"
    maxThreads="150" SSLEnabled="true" clientAuth="false">
    <SSLHostConfig>
        <Certificate
            certificateKeystoreFile="jks 路径"
            certificateKeyAlias="jks 证书别名"
            certificateKeystorePassword="jks 密码"
            type="RSA" />
        </SSLHostConfig>
        //其他站点复制多个 SSLHostConfig
    </Connector>
```

其中：

SSL 访问端口：port="443"

证书文件：certificateKeystoreFile ="jks 路径"

证书别名：certificateKeyAlias="jks 证书别名"

证书密码：certificateKeystorePassword ="jks 密码"

配置完成后，重新启动 Tomcat。

可选： 开启 HTTP 强制跳转 HTTPS。

配置 web.xml 文件，在文件</welcome-file-list>后添加以下内容：

```
<login-config>
  <!-- Authorization setting for SSL -->
  <auth-method>CLIENT-CERT</auth-method>
  <realm-name>Client Cert Users-only Area</realm-name>
</login-config>
<security-constraint>
  <!-- Authorization setting for SSL -->
  <web-resource-collection >
    <web-resource-name >SSL</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

2.6.3 Nginx 证书配置

Nginx 使用 KEY 和 CRT 格式的证书，证书制作方式请参考“2.4.3 使用 OpenSSL 工具制作证书”。

将服务器证书、中级证书和根证书打开，依次将其代码复制到文本文件中（包括“-----BEGIN CERTIFICATE-----”和“-----END CERTIFICATE-----”），并保存成 server.crt。如下：

```
-----BEGIN CERTIFICATE-----
服务器证书编码
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
中级证书编码
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
根证书编码
-----END CERTIFICATE-----
```

将服务器证书文件 `server.key` 和 `server.crt`，配置在 Nginx 中。

如果是单向 SSL，用文本编辑器打开 Nginx 根目录下 `conf/nginx.conf` 文件，更新以下内容：

```
server {
    listen 443;

    server_name 127.0.0.1;

    ssl on;

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;

    ssl_certificate server.crt;

    ssl_certificate_key server.key;
}
```

由于版本问题，配置文件可能存在不同的写法。例如：Nginx 版本为 `nginx/1.15.0` 以上请使用 `listen 443 ssl` 代替 `listen 443` 和 `ssl on`。

其中：

启用 SSL 功能：`ssl on`

禁用 SSLv2、SSLv3 协议：`ssl_protocols TLSv1 TLSv1.1 TLSv1.2`

公钥文件：`ssl_certificate server.crt` 路径

私钥文件：`ssl_certificate_key server.key` 路径

上述设置完成过后，重新启动 Nginx。

如果是双向 SSL，用文本编辑器打开 Nginx 根目录下 conf/nginx.conf 文件，更新以下内容：

```
server {  
    listen 443;  
    server_name 127.0.0.1;  
    ssl on;  
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;  
    ssl_certificate server.crt;  
    ssl_certificate_key server.key;  
    ssl_client_certificate ca.crt;  
    ssl_verify_client on;  
    ssl_verify_depth 2;  
}
```

其中：

启用 SSL 功能：ssl on

禁用 SSLv2、SSLv3 协议：ssl_protocols TLSv1 TLSv1.1 TLSv1.2

公钥文件：ssl_certificate server.crt 路径

私钥文件：ssl_certificate_key server.key 路径

证书链文件：ssl_client_certificate ca.crt 路径

启用双向 SSL：ssl_verify_client on

证书链深度：ssl_verify_depth 2 如果客户端使用 CFCA 证书，则该项必须为 2

上述设置完成过后，重新启动 Nginx。

可选： 设置 HTTP 请求自动跳转 HTTPS。

在需要跳转的 HTTP 站点下添加以下 rewrite 语句，实现 HTTP 访问自动跳转到 HTTPS 页面。

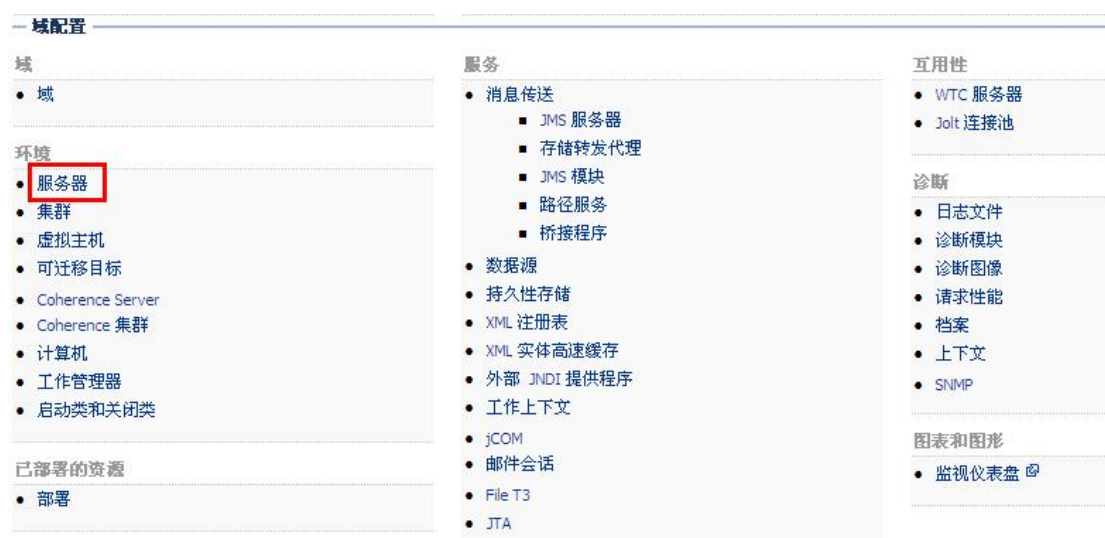
```
server {  
    listen 80;  
    server_name localhost;    #将 localhost 修改为您证书绑定的域名，例如：  
    www.example.com。
```

```
rewrite ^(.*)$ https://$host$1 permanent; #将所有 http 请求通过
rewrite 重定向到 https。
location / {
index index.html index.htm;
}
}
```

2.6.4 Weblogic 证书配置

Weblogic 使用 JKS 格式的证书，证书制作方式请参考“2.4.2 使用 Keytool 工具制作证书”。

打开 Weblogic 控制台，进入“服务器”。



选择部署的服务器。



在“一般信息”中，“启用 SSL 监听端口”。

AdminServer 的设置

配置 协议 日志记录 调试 监视 控制 部署 服务 安全 注释

一般信息 集群 服务 密钥库 SSL 联合服务 部署 迁移 优化 超载 健康状况监视 服务器启动 Web 服务

保存

使用此页可以配置该服务器的一般功能, 例如默认网络通信。
[查看 JNDI 树](#)

名称: AdminServer 此服务器实例的字母数字式名称。 [更多信息...](#)

计算机: (None) 将要运行此服务器的 WebLogic Server 主机 (计算机)。 [更多信息...](#)

集群: (Standalone) 该服务器所属的集群, 或 WebLogic Server 实例组。 [更多信息...](#)

监听地址: 此服务器用于监听传入连接的 IP 地址或 DNS 名。 [更多信息...](#)

☒ 启用监听端口 指定是否可以通过默认的非 SSL 监听端口访问此服务器。 [更多信息...](#)

监听端口: 7001 此服务器用来监听常规 (非 SSL) 传入连接的默认 TCP 端口。 [更多信息...](#)

☒ 启用 SSL 监听端口 指示是否可以通过默认的 SSL 监听端口访问服务器。 [更多信息...](#)

SSL 监听端口: 7443 此服务器监听 SSL 连接请求所使用的 TCP/IP 端口。 [更多信息...](#)

☐ 启用客户机证书代理 指定 HttpClusterServlet 是否代理特殊标头中的客户机证书。 [更多信息...](#)

Java 编译器: javac 供此服务器上所有需要编译 Java 代码的应用程序使用的 Java 编译器。 [更多信息...](#)

在“密钥库”页面，配置服务器证书（server.jks）。其中：

密钥库选择“定制标识和定制信任”；

密钥库输入 server.jks 的路径和密码；

信任密钥库输入 server.jks 的路径和密码；

输入完成后，保存。

AdminServer 的设置

配置 协议 日志记录 调试 监视 控制 部署 服务 安全 注释

一般信息 集群 服务 **密钥库** SSL 联合服务 部署 迁移 优化 超载 健康状况监视 服务器启动 Web 服务

保存

密钥库可以确保私有密钥和信任证书颁发机构 (CA) 的安全存储和管理。在此页中, 您可以查看和定义各种密钥库配置。这些设置有助于管理消息传输的安全。

密钥库: 定制标识和定制信任 [更改](#) 查找服务器的标识和信任密钥库时应该使用哪些配置规则? [更多信息...](#)

— 标识 —

定制标识密钥库: c:\server.jks 标识密钥库的路径和文件名。 [更多信息...](#)

定制标识密钥库类型: jks 密钥库的类型。此项一般为 JKS。 [更多信息...](#)

定制标识密钥库密码短语: 定制标识密钥库的加密密码短语。如果为空或空值, 打开密钥库时将不需要密码短语。 [更多信息...](#)

确认定制标识密钥库密码短语:

— 信任 —

定制信任密钥库: c:\server.jks 定制信任密钥库的路径和文件名。 [更多信息...](#)

定制信任密钥库类型: jks 密钥库的类型。此项一般为 JKS。 [更多信息...](#)

定制信任密钥库密码短语: 定制信任密钥库的密码短语。如果为空或空值, 打开密钥库时将不需要密码短语。 [更多信息...](#)

确认定制信任密钥库密码短语:

保存

在“SSL”页签，配置 SSL 选项。其中：

标识和信任设置，选择“密钥库”；

私有密钥输入密钥别名和密码；

输入完成后，保存。



配置完成后，激活 Weblogic 更改，重新启动 Weblogic 服务。

注： 全球服务器证书为 SHA256 算法的，所以 weblogic 版本必须为 10.3.3 或者更高版本，且这些版本的 weblogic 必须勾选“使用 JSSE SSL”。参考如下两个图：

1. SHA as HASH ALgorithm : If while signing the Certificate, signature hash algorithm used by CA is SHA256 (to find Algorithm, click certificate and then Details) then this is supported only on WebLogic 10.3.3 or higher version (for prior version of WebLogic use SHA1). For WebLogic 10.3.3 or higher with SHA256, select option Use JSSE SSL in SSL tab



2.6.5 IBM Http Server 证书配置

IBM Http Server 使用 KDB 格式的证书，证书制作方式请参考“2.4.4 使用 iKeyman 工具制作证书”。

将制作好的 kdb、rdb、sth 文件放在同一个目录下，而后在 httpd.conf 文件中配置。

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so

Listen 443

<VirtualHost 127.0.0.1:443>

    ServerName 127.0.0.1

    SSLEnable

    SSLClientAuth required

    Keyfile "key.kdb 路径"

    SSLStashfile "key.sth 路径"

</VirtualHost>

SSLDisable
```

其中，KeyFile 所指定的为证书数据库路径，SSLStashfile 为密码文件路径。

配置完成后，重启启动 IBM HTTP Server。

2.6.6 JBoss 证书配置

JBoss 使用 JKS 格式的证书，证书制作方式请参考“2.4.2 使用 Keytool 工具制作证书”。

将服务器证书文件（server.jks），配置在 JBoss 中。

用文本编辑器打开 Jboss 安装目录下 server/default/deploy/jbossweb.sar 目录中的 server.xml 文件，更新以下内容。

```
<Connector protocol="HTTP/1.1" SSLEnabled="true"
  port="443" address="{jboss.bind.address}"
  scheme="https" secure="true" clientAuth="false"
  keystoreFile="jks 路径"
  keystorePass="jks 密码"
  truststoreFile="jks 路径"
  truststorePass="jks 密码"
  sslProtocol = "TLS"/>
```

其中：

SSL 访问端口：port="443"

证书文件：keystoreFile="jks 路径"

证书密码：keystorePass="jks 密码"

信任证书链文件：truststoreFile="jks 路径"

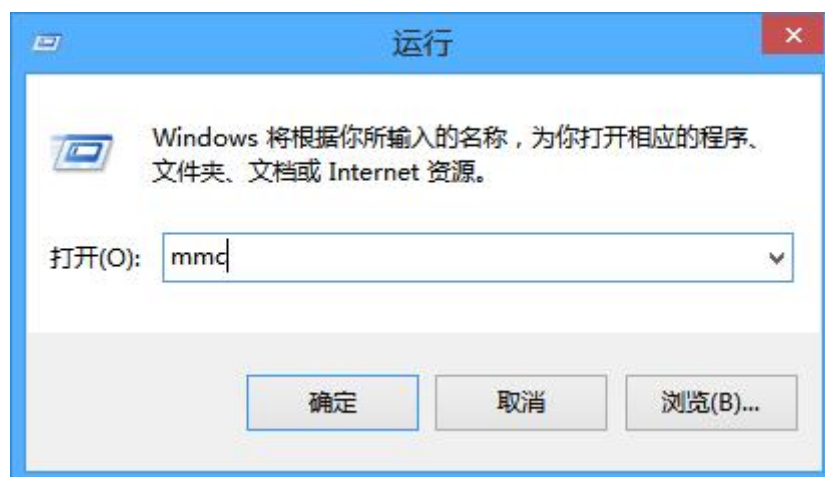
信任证书链密码：truststorePass="jks 密码"

配置完成后，重新启动 JBoss。

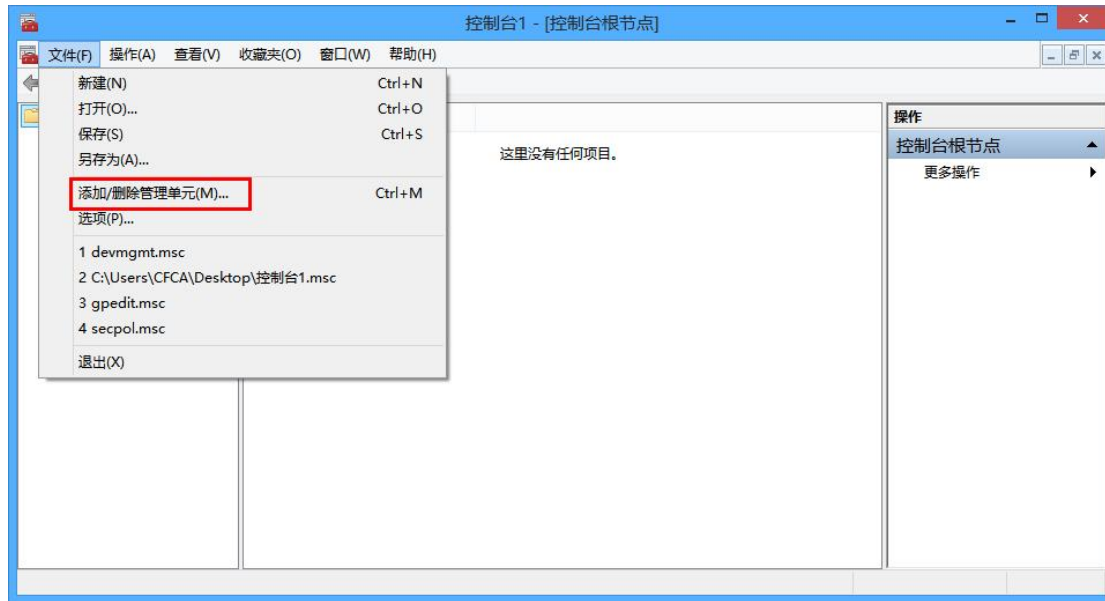
2.6.7 IIS 证书配置

IIS 可以直接使用 PFX 格式的证书文件，PFX 证书制作方式请参考“2.5 证书格式转换”。

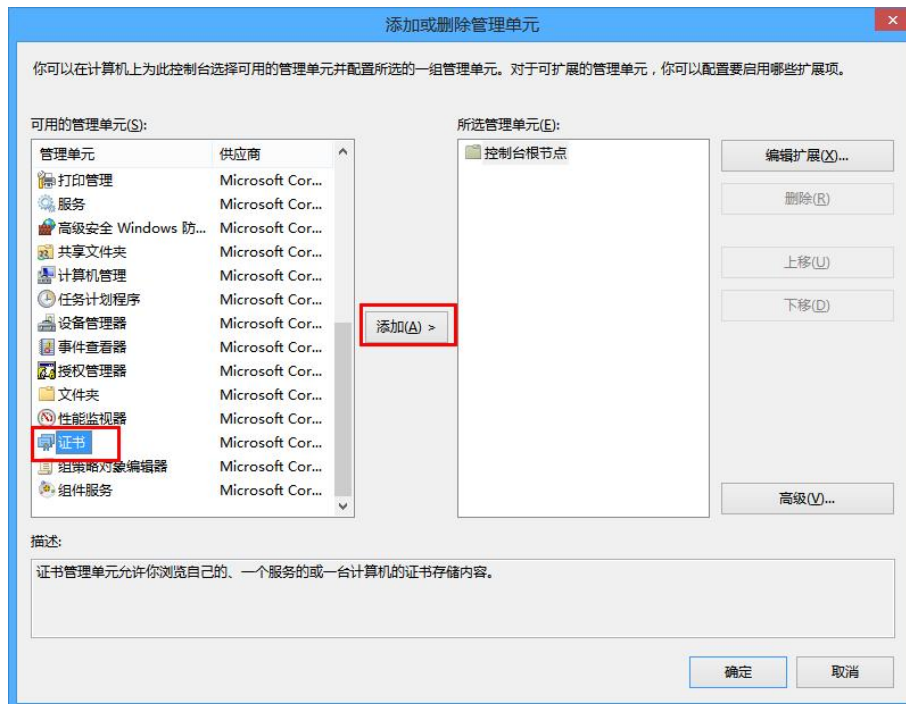
在运行框中输入 MMC，进入管理控制台。



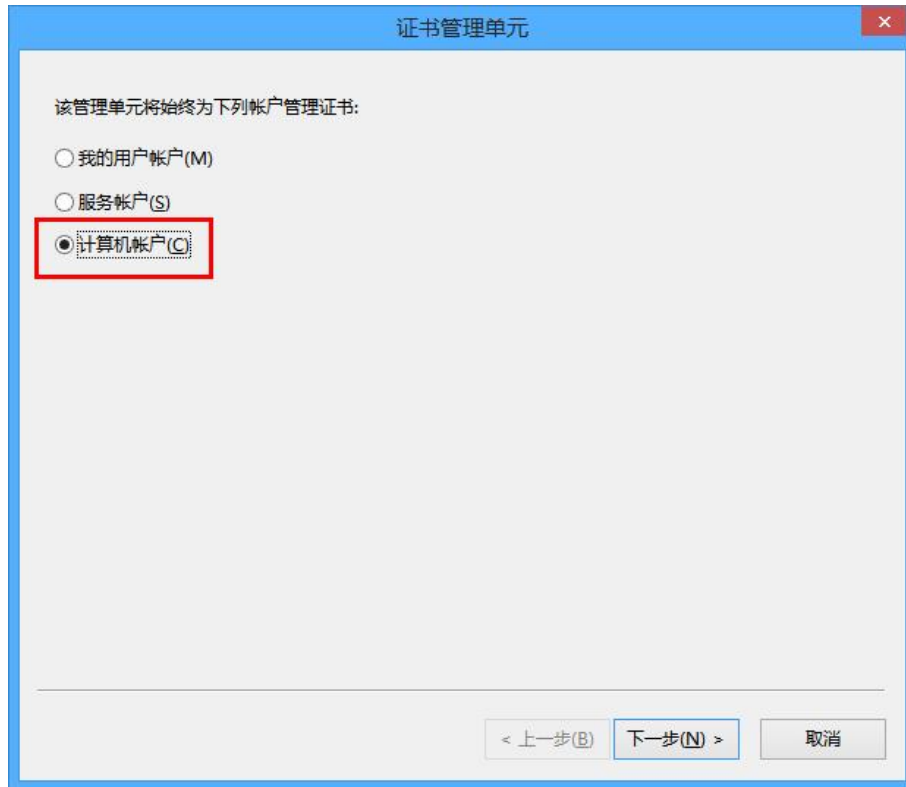
添加删除管理单元。



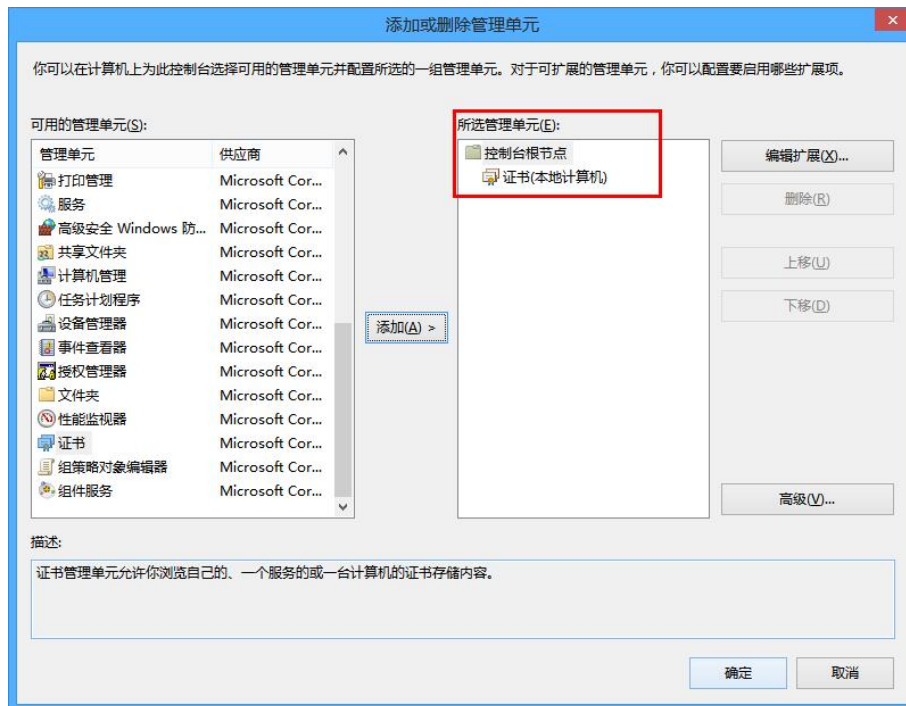
添加证书。



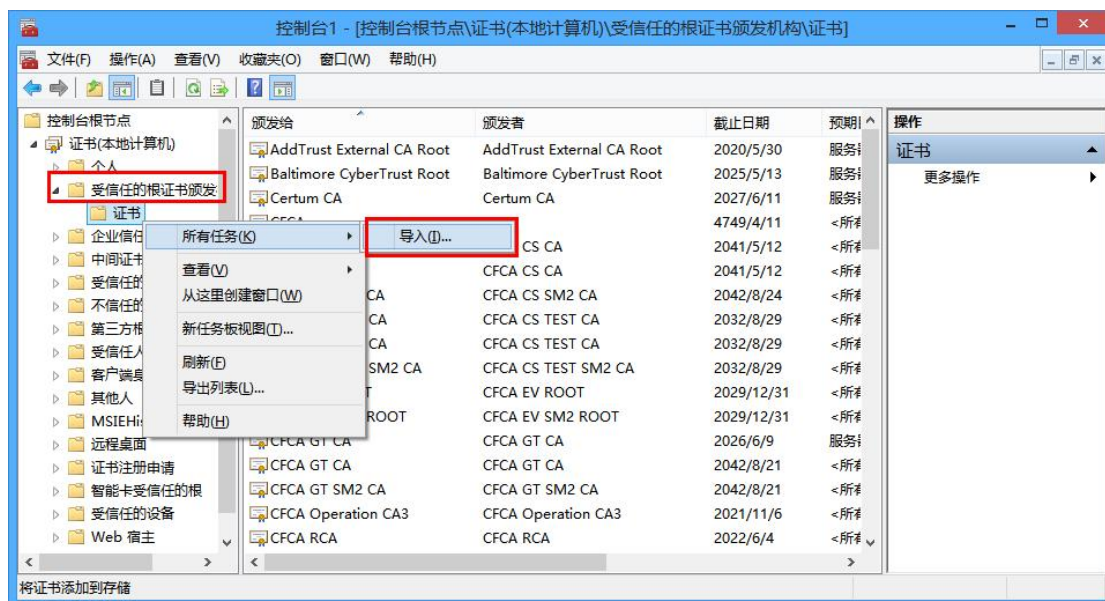
选择计算机账户。



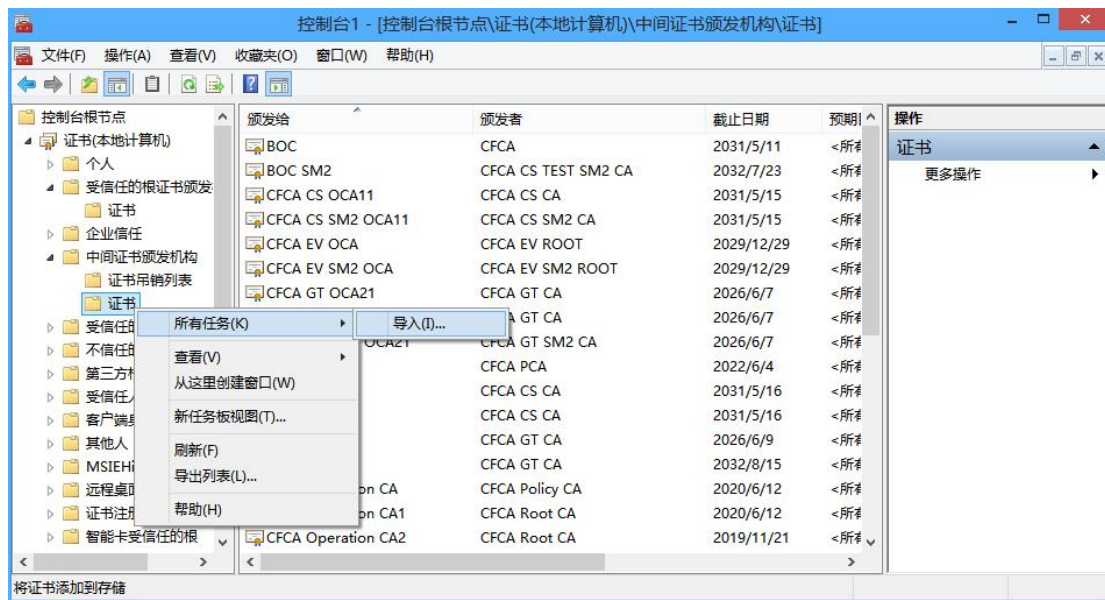
添加之后，确定。



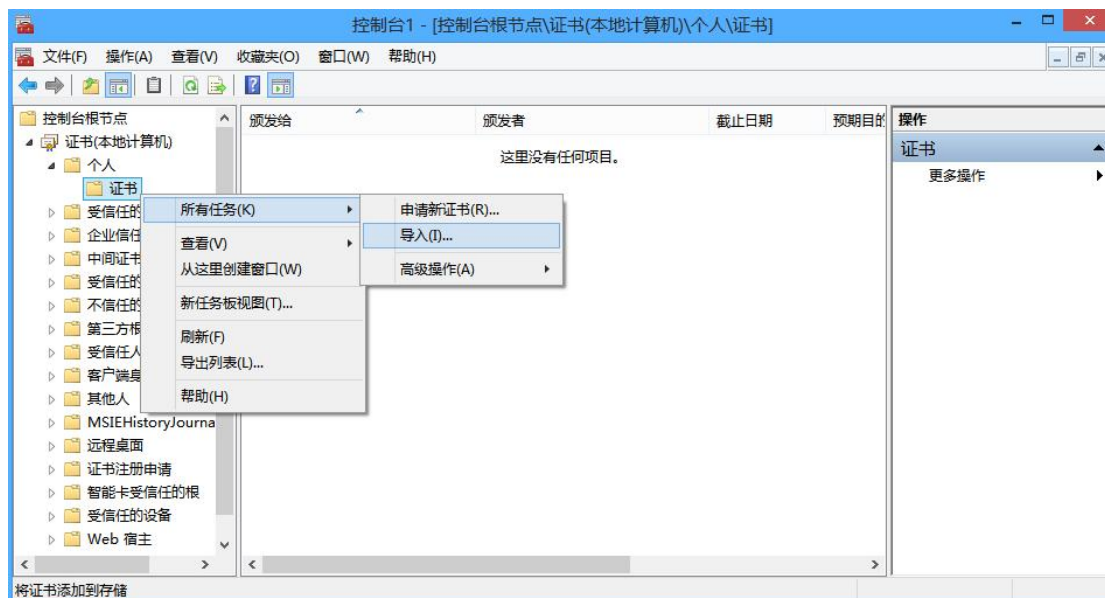
在受信任的根证书颁发机构中，导入根证书。



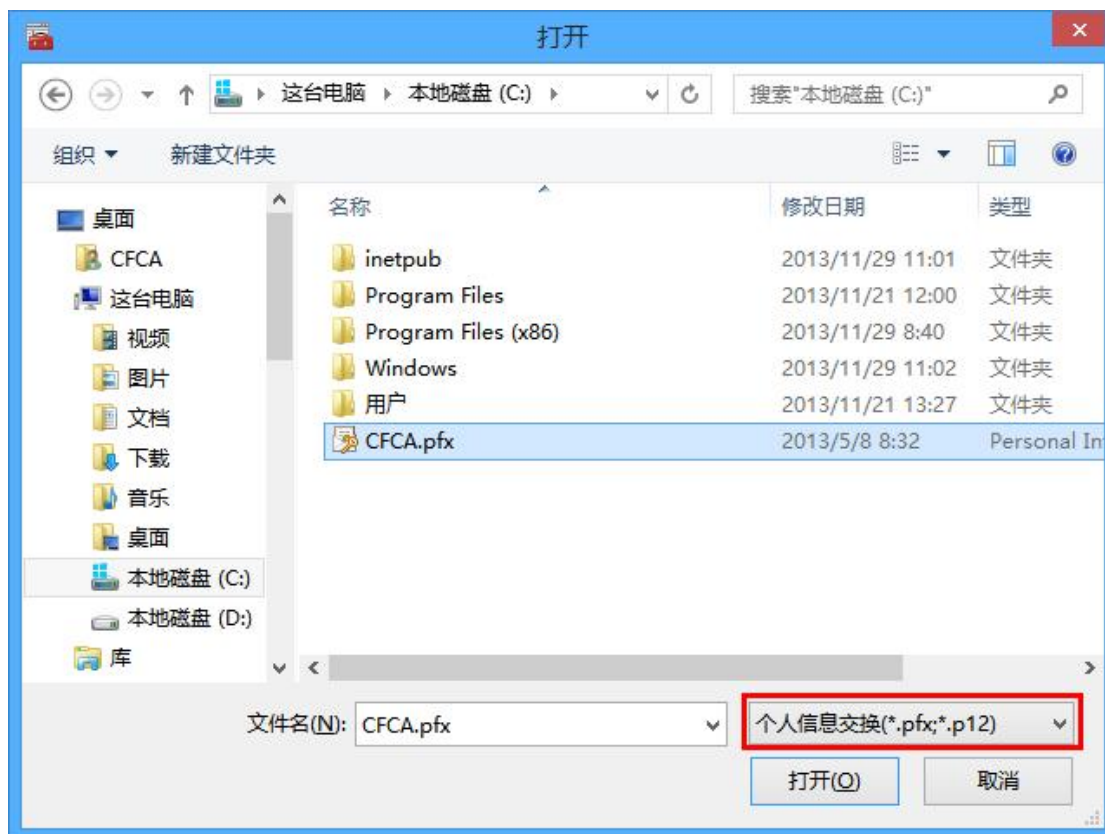
在中级证书颁发机构中，导入中级证书。



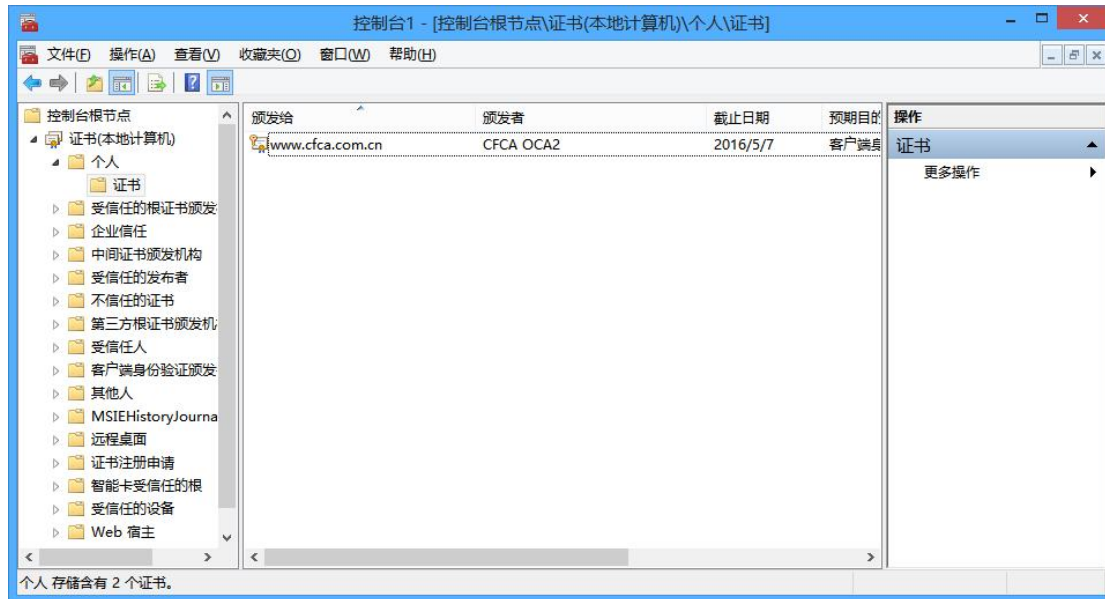
在个人证书中，导入 PFX 证书文件。



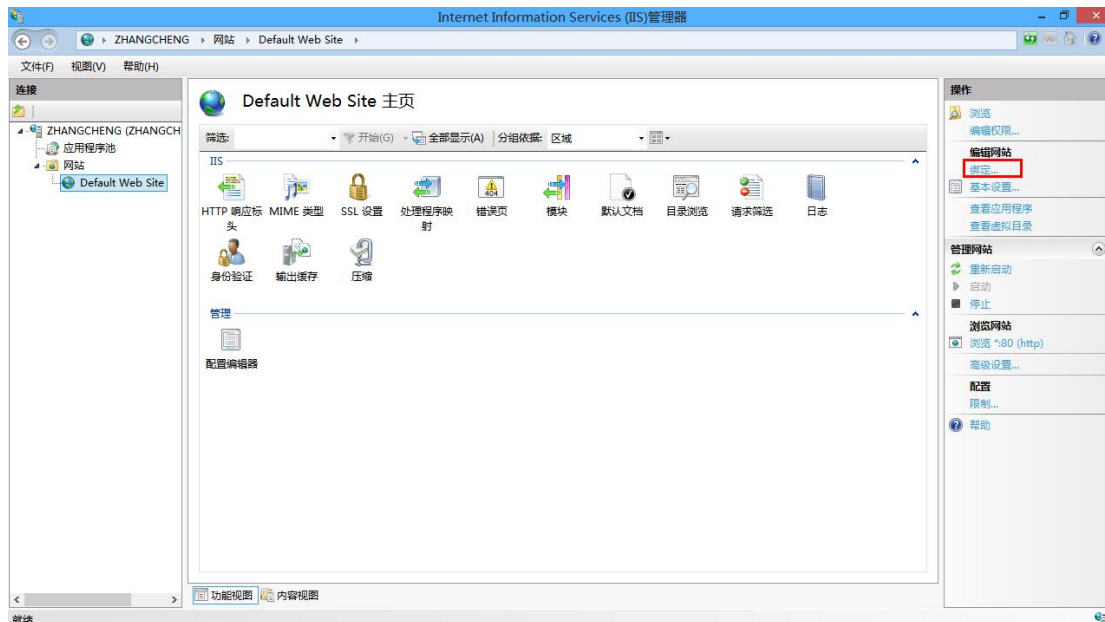
导入时，选择个人信息交换（PFX、P12）。



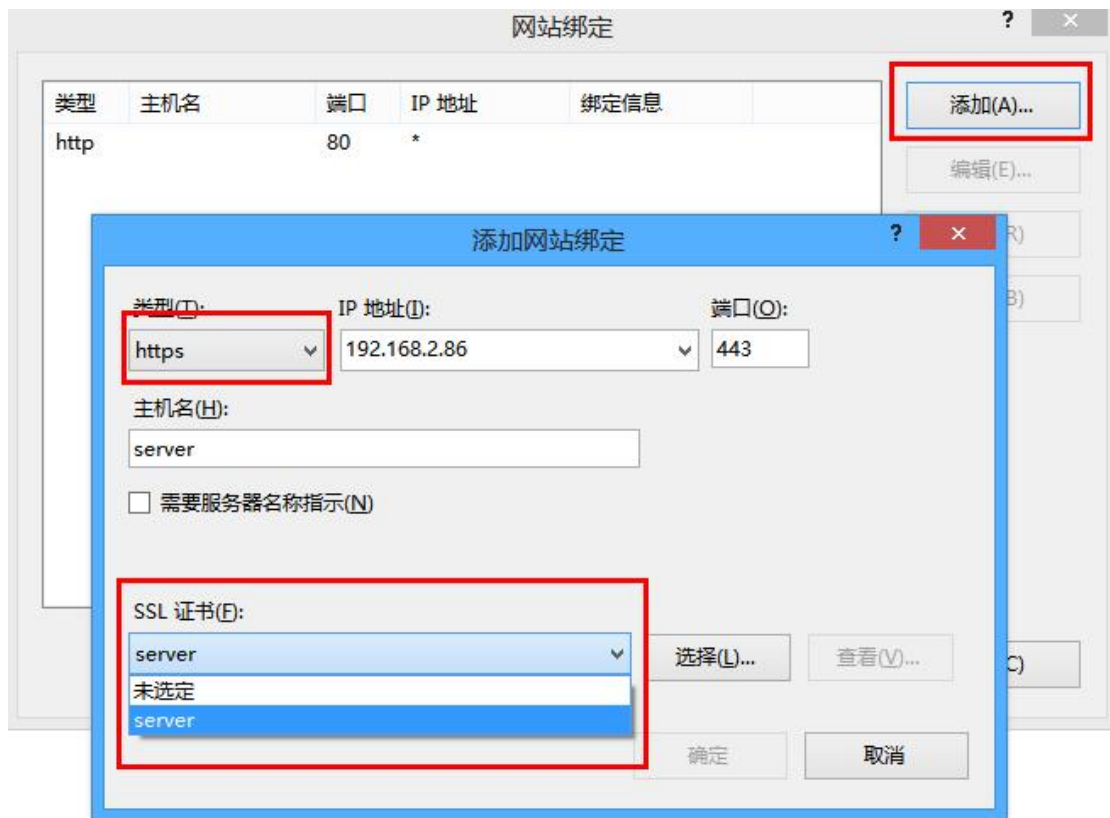
导入完成。



在 IIS 管理控制台，选择站点，点击“绑定”。



点击“添加”，选择“https”，选择 SSL 证书，点击确定。



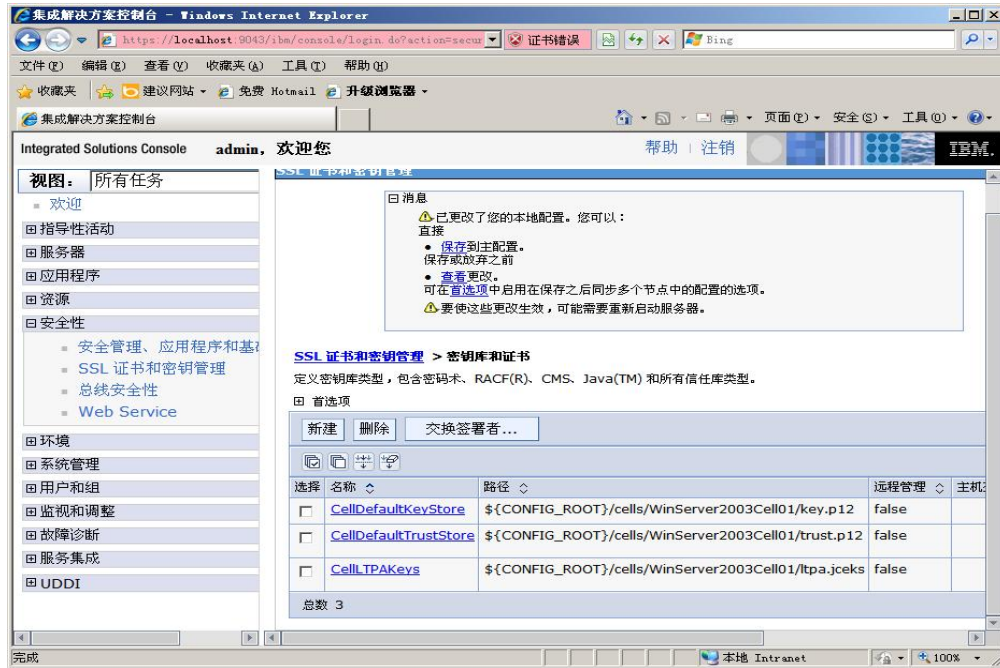
重新启动 IIS 即可。

2.6.8 Websphere 证书配置

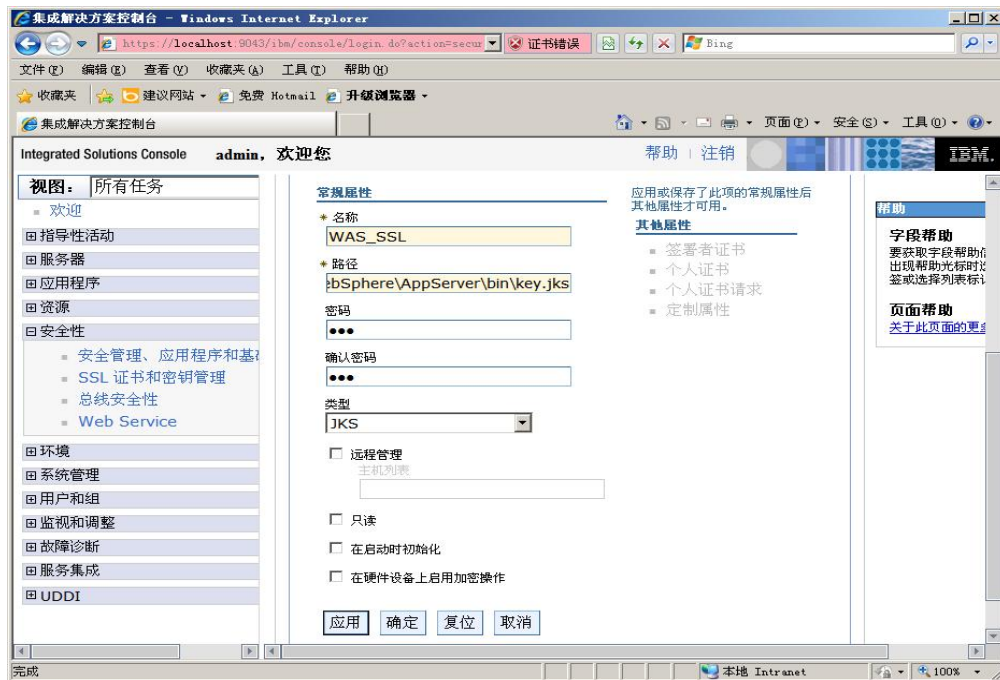
Websphere 使用 JKS 格式的证书，证书制作方式请参考“2.4.2 使用 Keytool 工具制作证书”。

将准备好的 JKS 文件放在适当的目录中，如 Websphere 主目录\AppServer\bin 中。

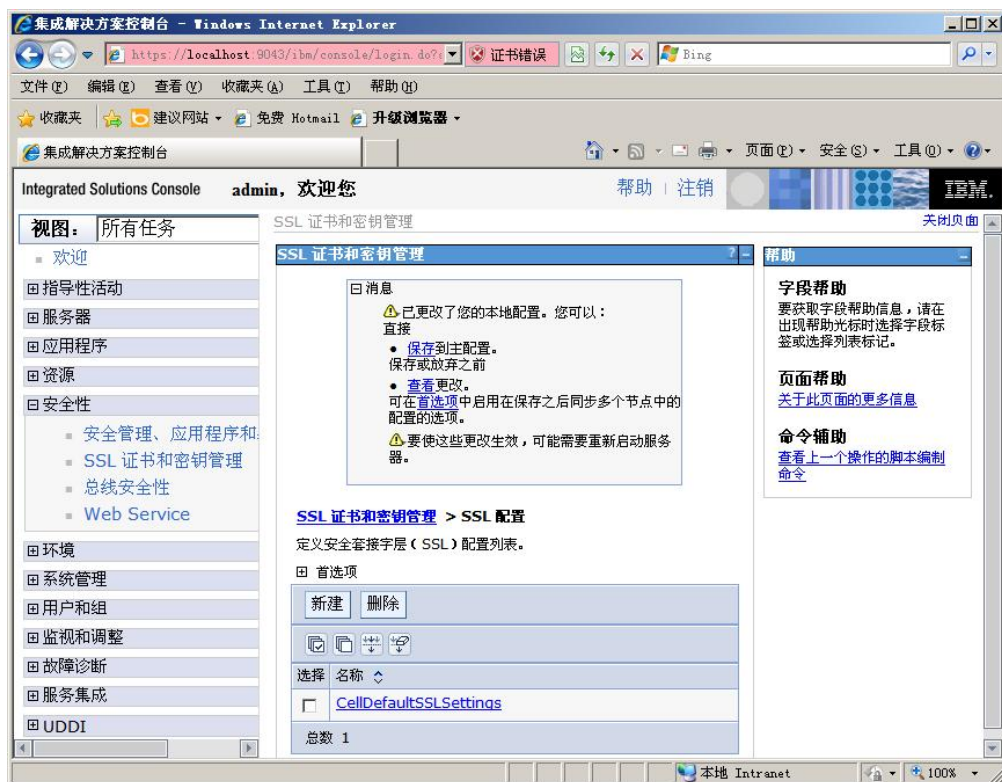
然后打开 WebSphere 管理控制台：



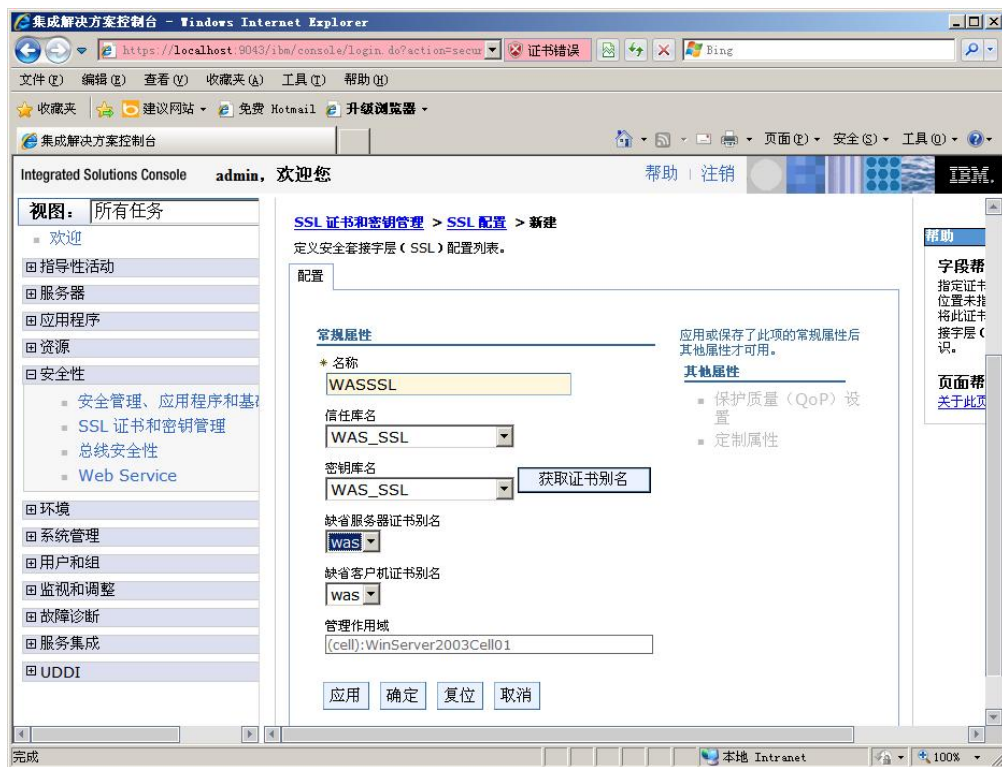
选择“SSL 证书和密钥管理”->“密钥库和证书”->“新建”，



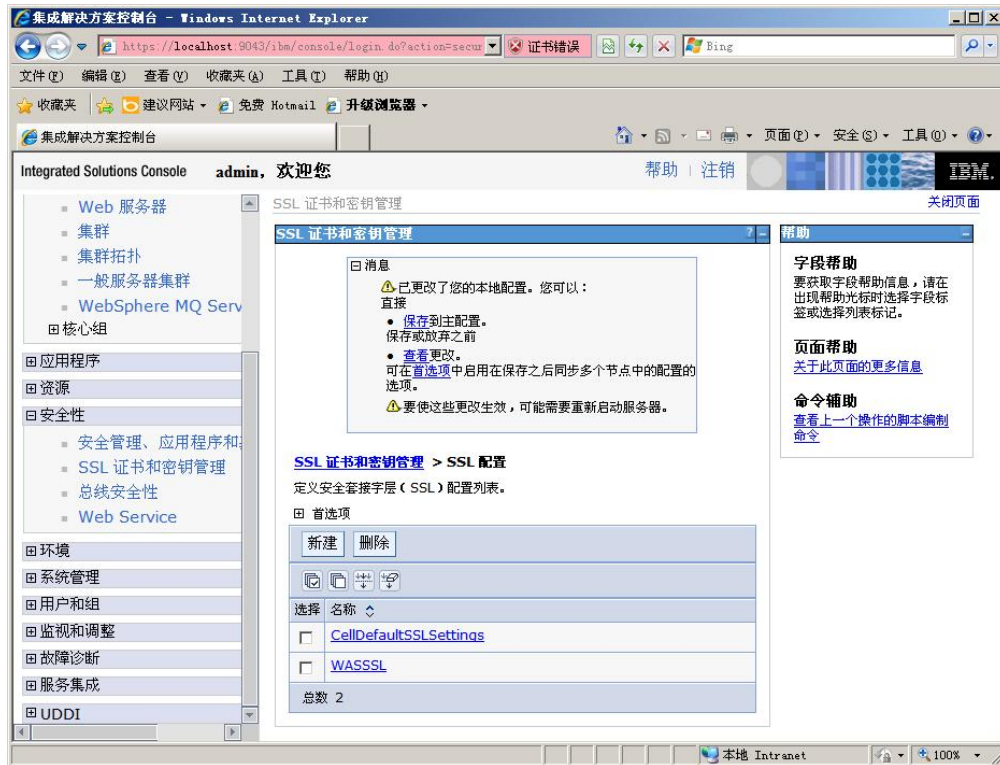
输入 JKS 文件信息，并单击“应用”。



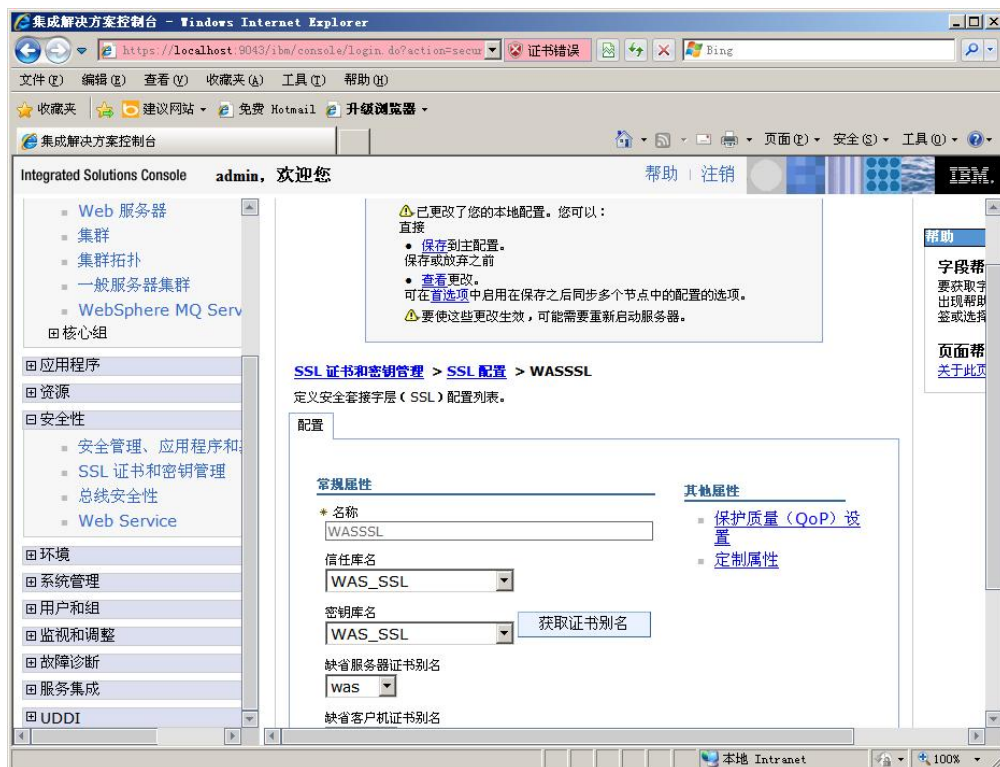
然后选择“SSL 证书和密钥管理”->“SSL 配置”->“新建”，



输入“名称”，选择刚才创建的“信任库名”和“密钥库名”后，单击“获取证书别名”，然后单击“应用”。



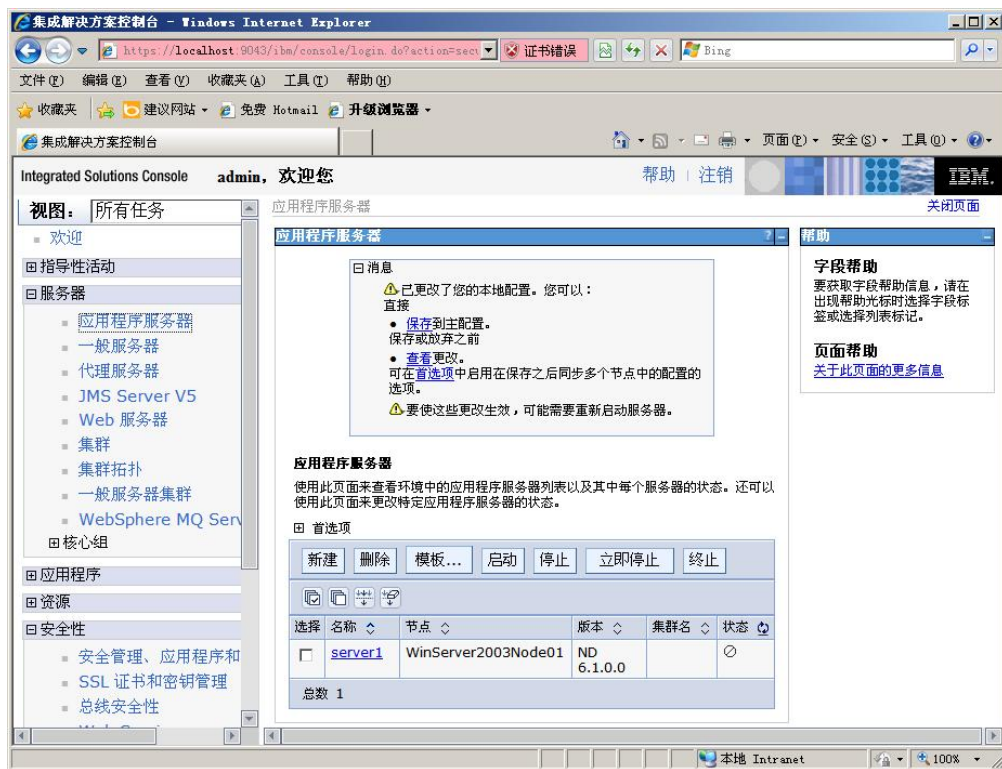
再次进入“SSL 证书和密钥管理”->“SSL 配置”，选择刚才创建的配置“WASSSL”，



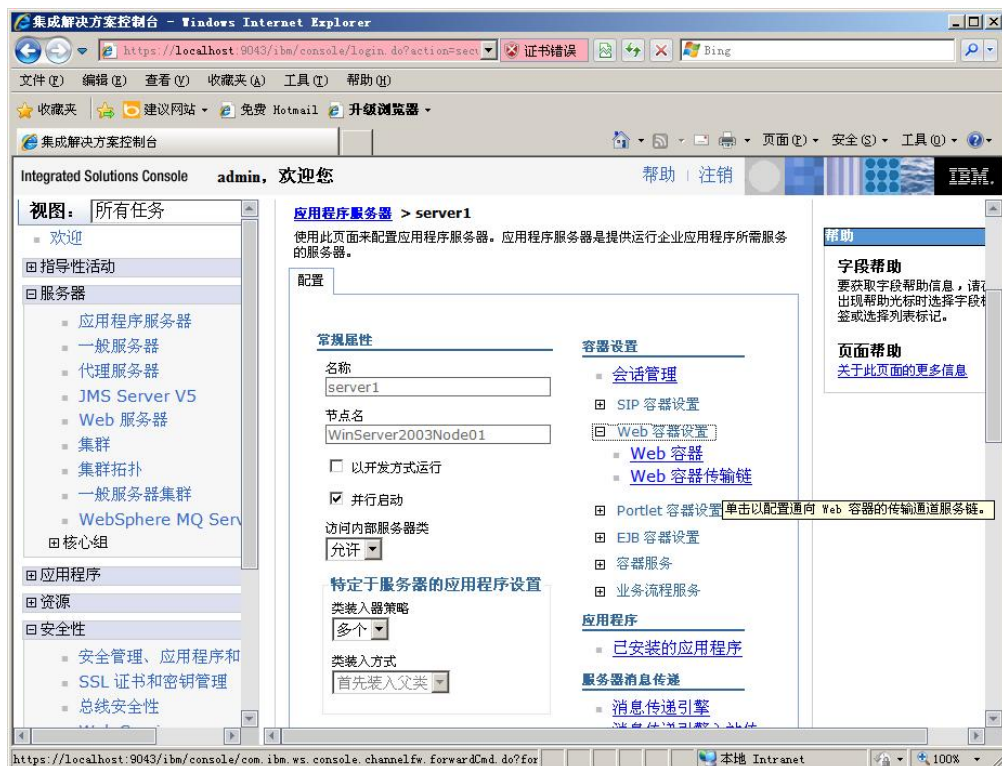
选择“保护质量(QoP)设置”，



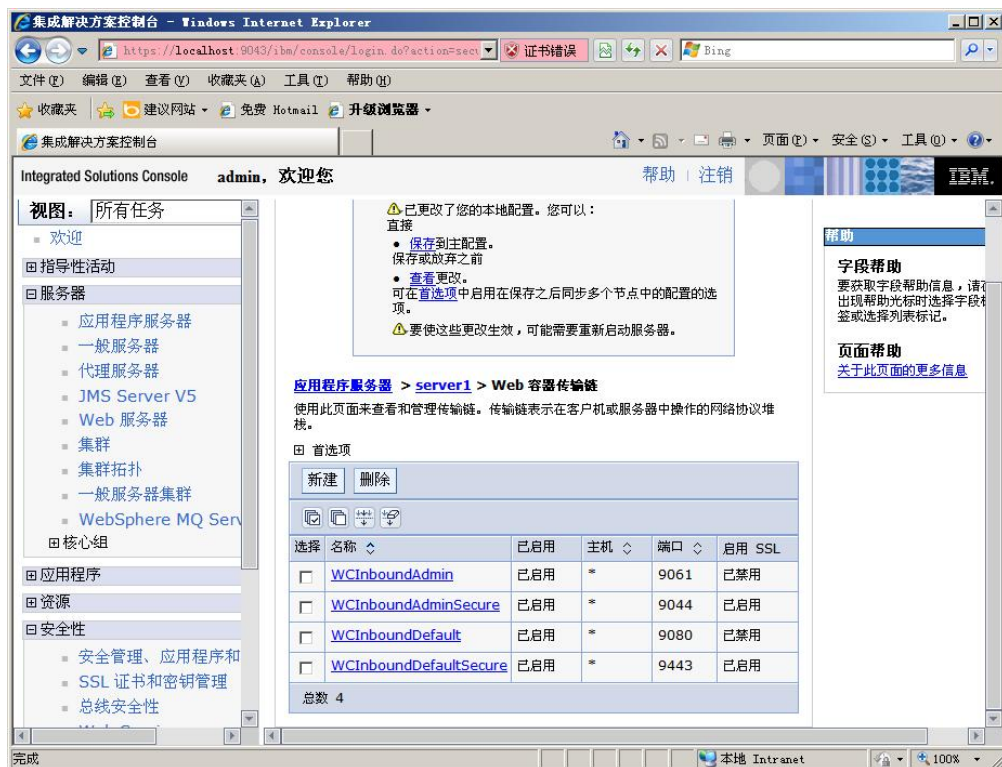
在“客户机认证”中选择“必需的”，然后单击“应用”。



进入“服务器”->“应用程序服务器”，选择“server1”，



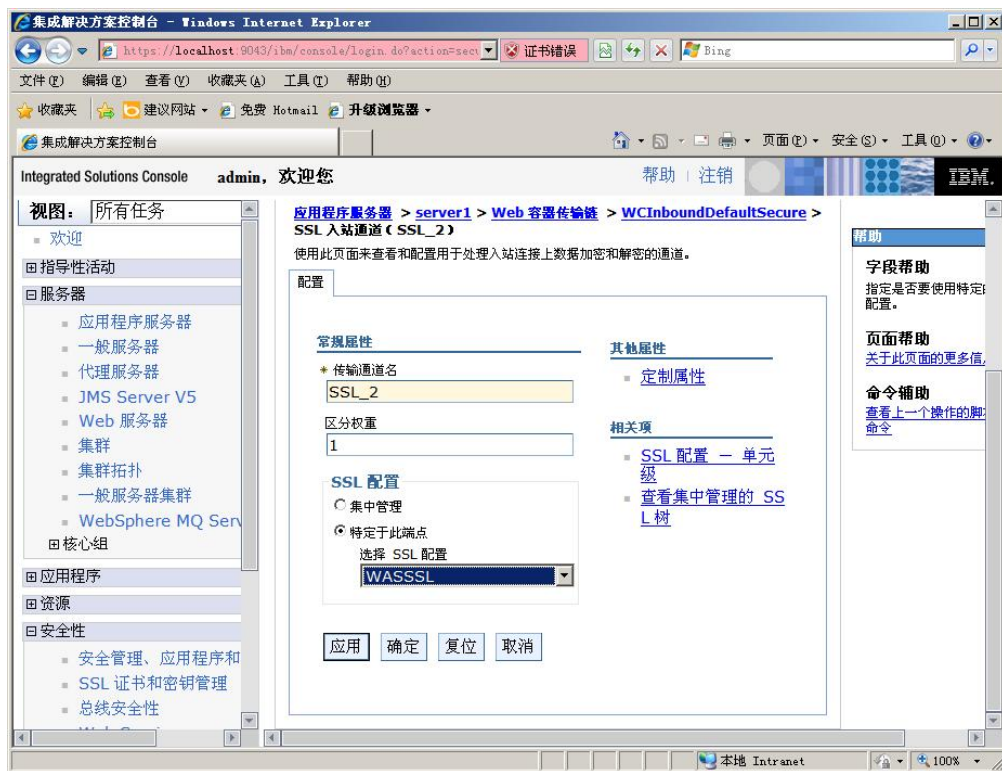
选择“Web 容器设置”->“Web 容器传输链”，



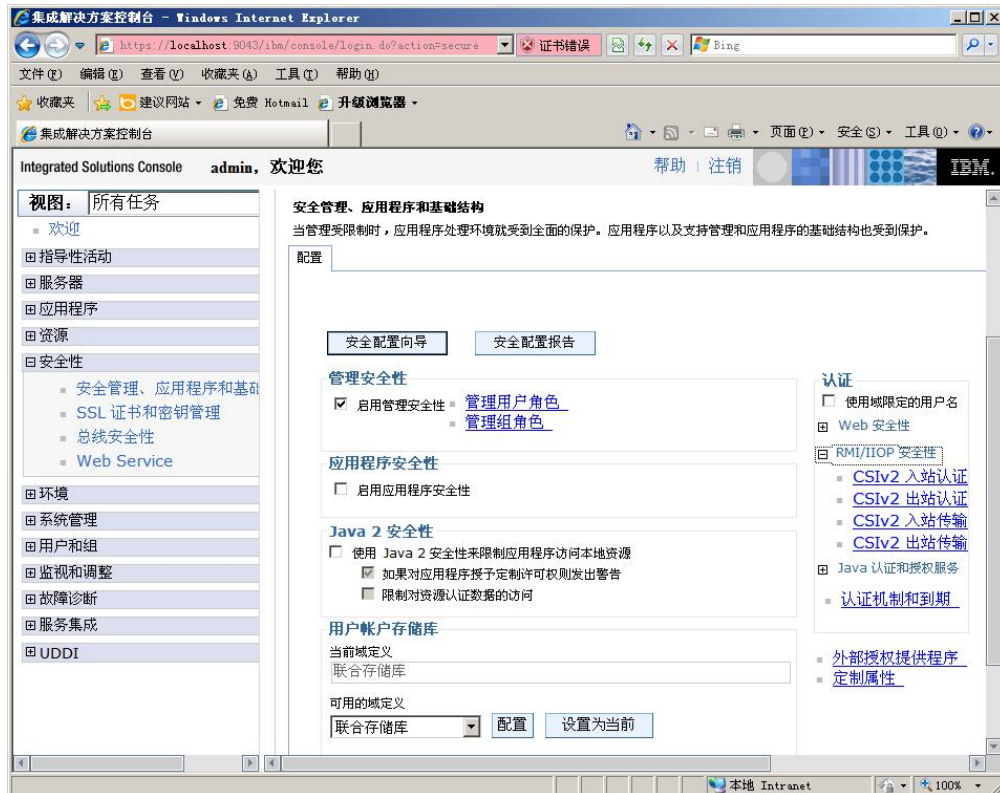
选择“WCInboundDefaultSecure”，



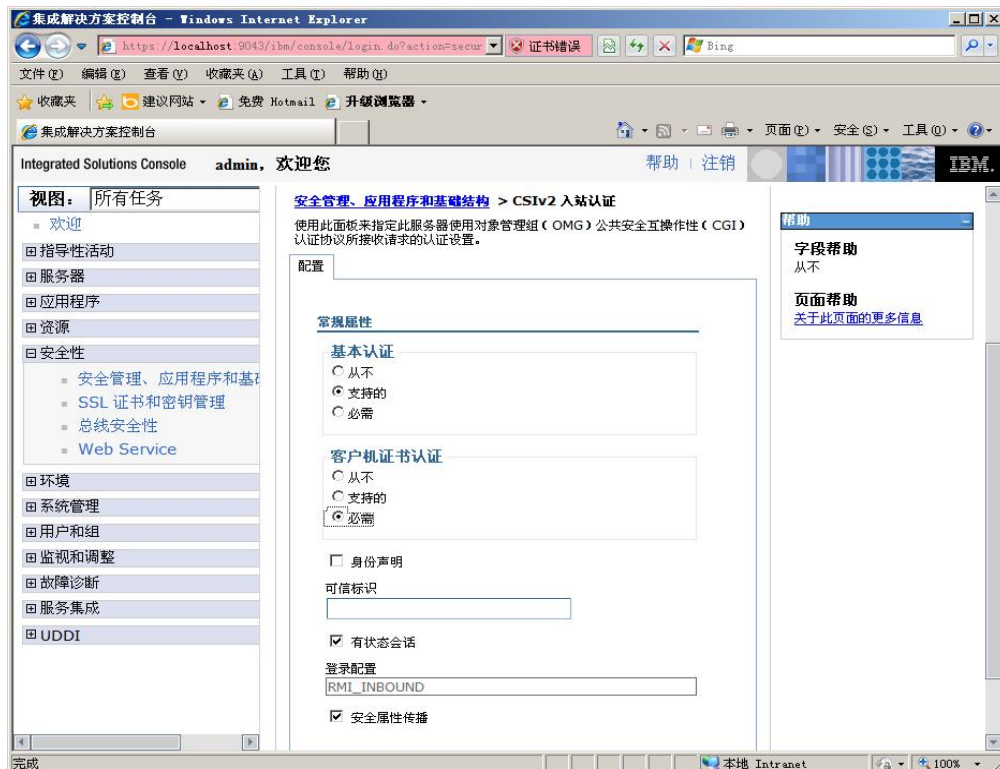
选择“SSL 入站通道(SSL_2)”。



在“SSL 配置”中，选择“特定于此端点”，然后在“选择 SSL 配置”中，选择刚才创建的 SSL 配置“WASSSL”，单击“应用”。

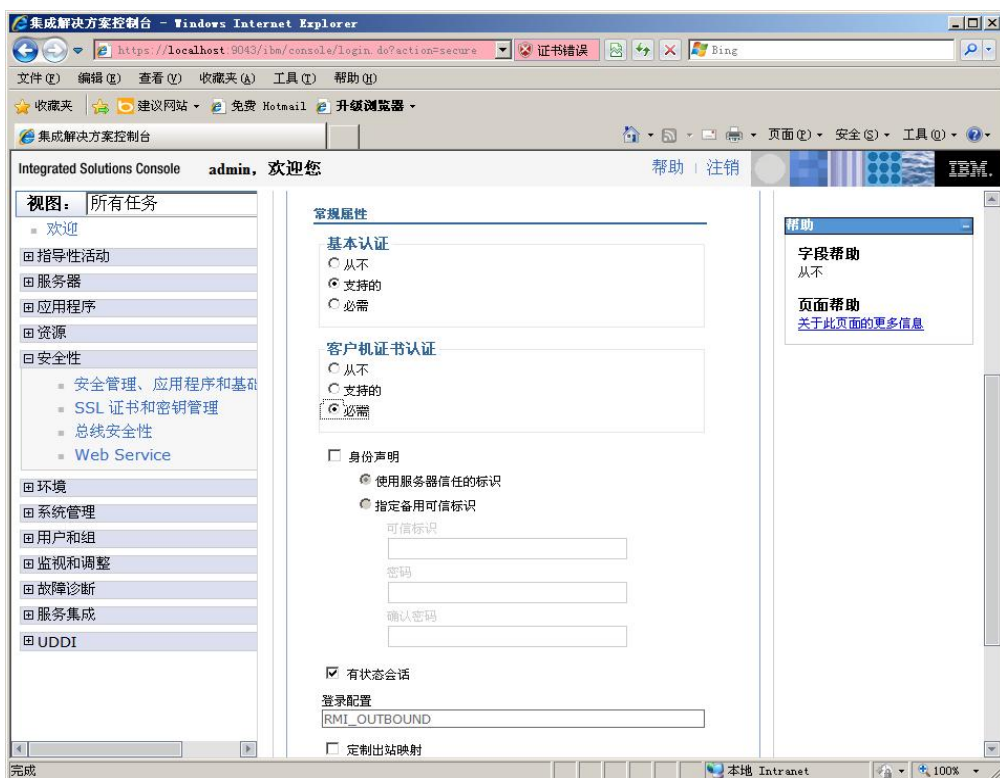


进入“安全性”->“安全管理、应用程序和基础结构”->“认证”->“RMI/IIOP 安全性”，选择“CSiv2 入站认证”，



在“客户机证书认证”中，选择“必需”，单击“应用”。然后再选择“CSiv2

出站认证”，



在“客户机证书认证”中，选择“必需”，单击“应用”。配置完成。

2.6.9 IHS+WAS 证书配置

安装 IHS 和 WAS，

安装 WAS 插件，

将准备好的 kdb 和 sth 文件复制到适当的目录中，如 C:\bin 中，KDB 文件的生成方法请参考 2.4.4，

打开 IHS 的 conf 目录中的 httpd.conf 文件，在文件末尾加入如下内容：

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
```

```
<IfModule mod_ibm_ssl.c>
```

```
Listen 443
```

```
<VirtualHost *:443>
```

```
    ServerName 192.168.17.128
```

```
    SSLEnable
```

```
    SSLClientAuth Required
```

```
</VirtualHost>
```

```
</IfModule>
```

KeyFile "c:\bin\key.kdb"

SSLDisable

重启服务，配置完成！

2.6.10 F5 设备证书配置

F5 设备具体配置请以厂商手册为准，如下内容为参考：导入证书公钥如果是导入已经存在的域，则根据之前其他 F5 上的命名规则填写名称，如果为新建则使用如下命名规则，域名_ssl_版本和根证书_域名_版本，例如：login_ssl_v3 和 parent_login_v3，Import Type 选择“certificate”，找到 server.cer 公钥，选择“Import”

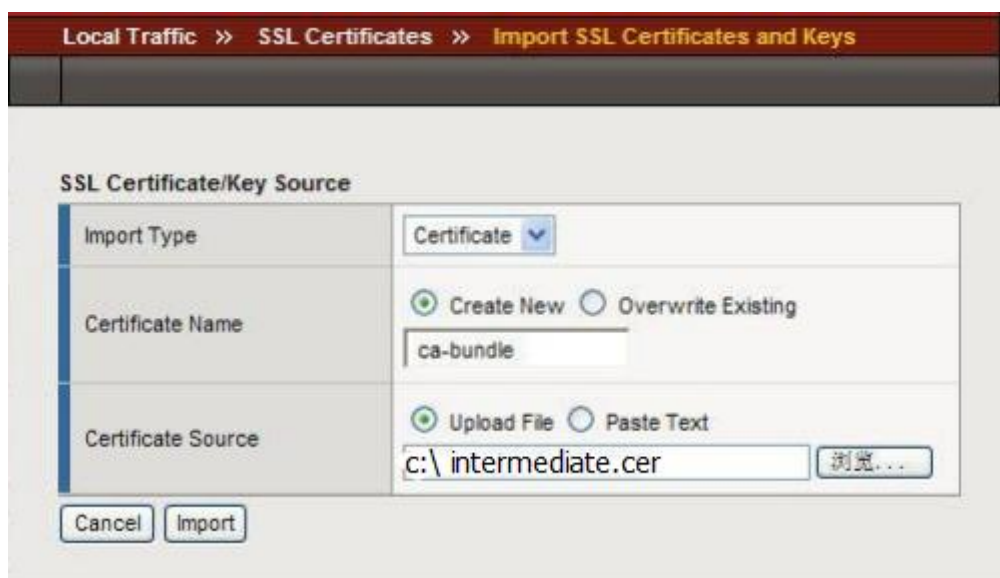
SSL Certificate/Key Source	
Import Type	Certificate
Certificate Name	<input checked="" type="radio"/> Create New <input type="radio"/> Overwrite Existing <input type="text"/> 填写证书名称
Certificate Source	<input checked="" type="radio"/> Upload File <input type="radio"/> Paste Text <input type="button" value="选择文件"/> 未选择文件

Cancel Import

导入证书的私钥 Key 文件为生成 csr 时生成的文件(第一步中下载压缩包内容 key 文件)，如果是导入已经存在的域，则根据之前其他 F5 上的命名规则填写名称，如果为新建则必须与证书名称相同，例如：证书名称为 login_ssl_v3，key 的名称也与证书名相同，Import Type 选择“key”，找到 server.key 私钥，选择“Import”

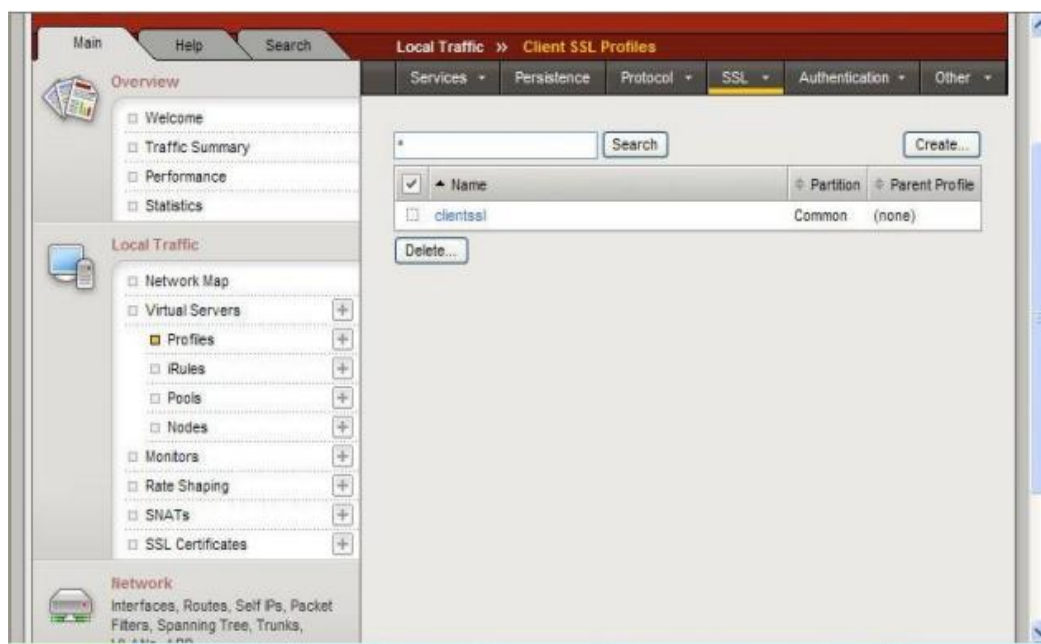


导入 CA 中级证书 选择 Local Traffic-> SSL Certificates 在 SSL Certificate List 主界面点击右上角“Import”,证书邮件保存的 intermediate.cer 使用“Certificate”方式导入。

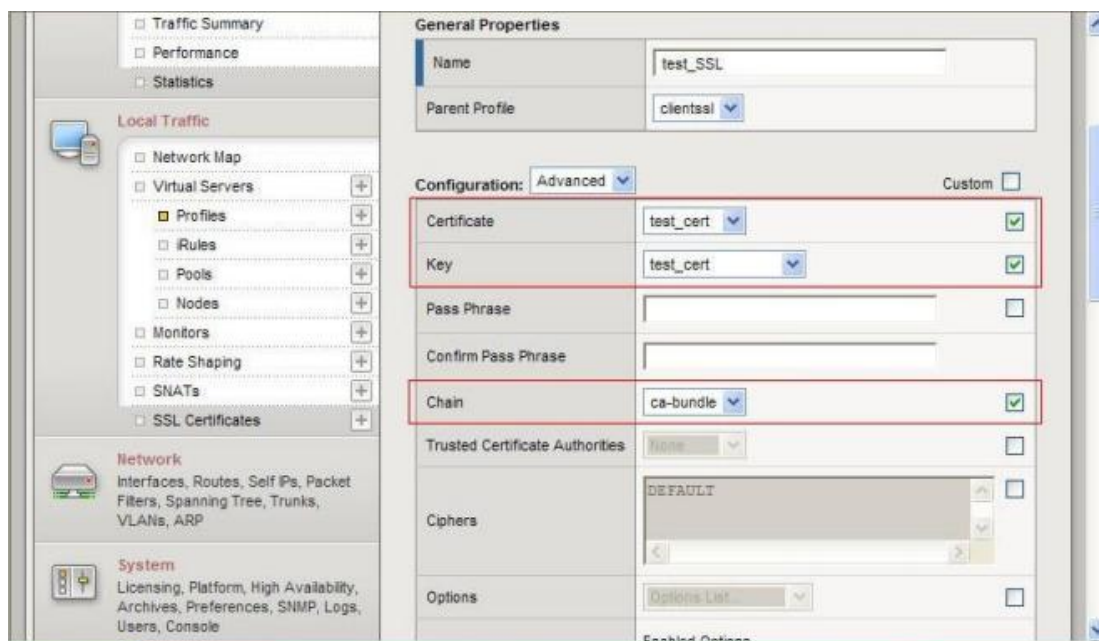


导入成功后，F5 将自动识别导入的证书为 Certificate Bundle。

配置服务器证书 选择“Local Traffic”-“Virtual Servers”-“Profiles” 选择“Profile”中，“SSL”下的“Client”进入“Client SSL Profile”设置 如果您需要为站点配置一个全新的 SSL 证书，则您需要新建一个 Client SSL Profile。如果您需要为一个已有证书的站点更新服务器证书，则仅需点击已存在的 Profile，进行编辑更新操作即可。



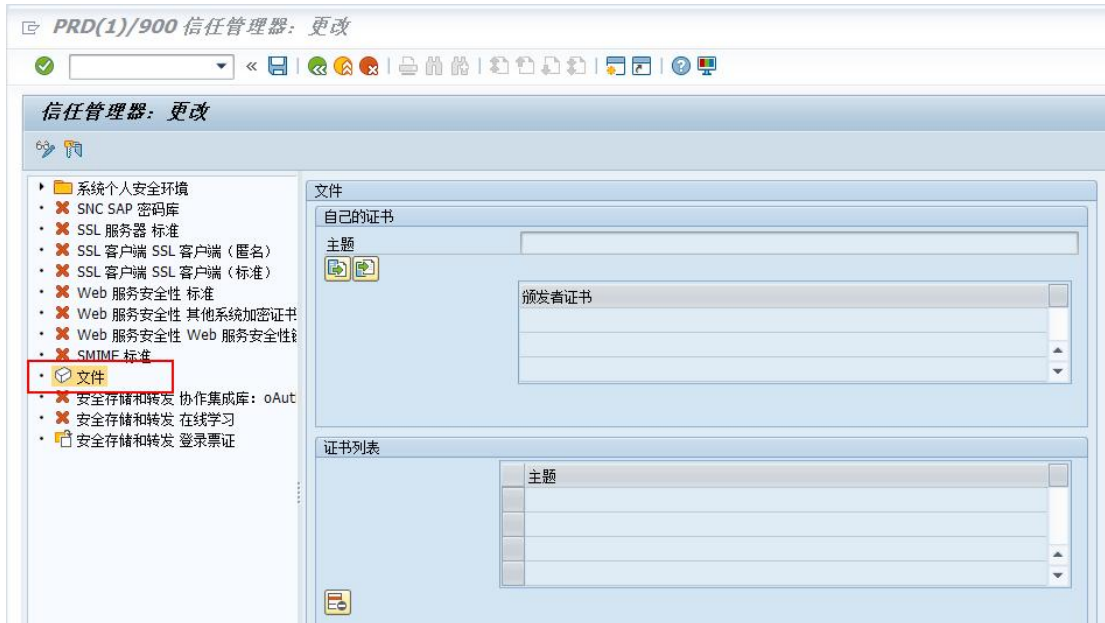
在新建的 Profile 中，选择当前 Profile 所使用的证书（Certificate）、私钥（key），以及在 Chain 处，设置与 该证书应用相关联的证书链（之前导入的中级 CA 证书）。完成后，选择“Update”保存



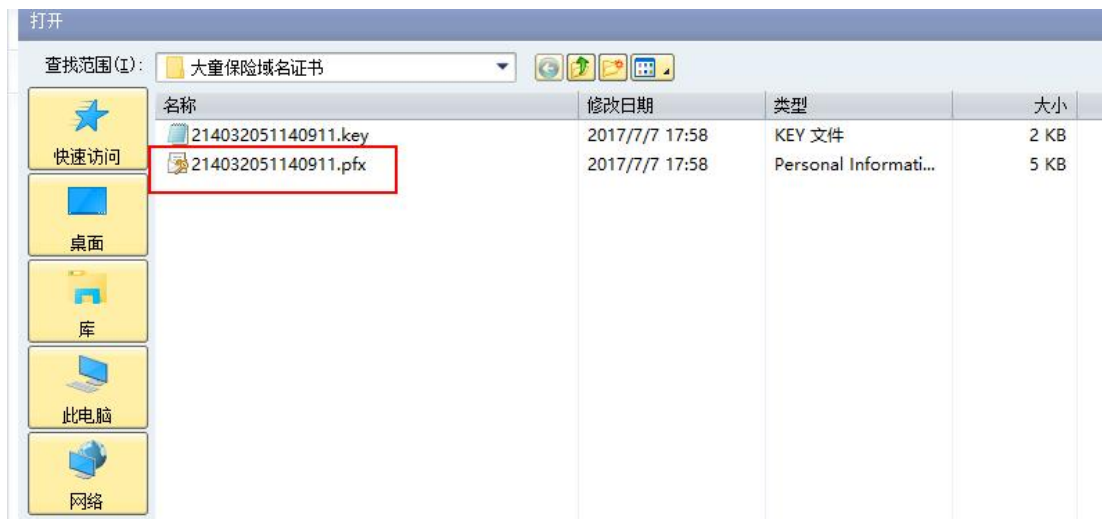
在证书成功配置后，需要创建一个 443 端口的 Virtual Server，并加载上面的 Client SS Profile 对应该站点 启用 SSL 证书。

2.6.11 SAP 证书配置

导入个人信息交换文件，事务码 STRUST，双击文件节点



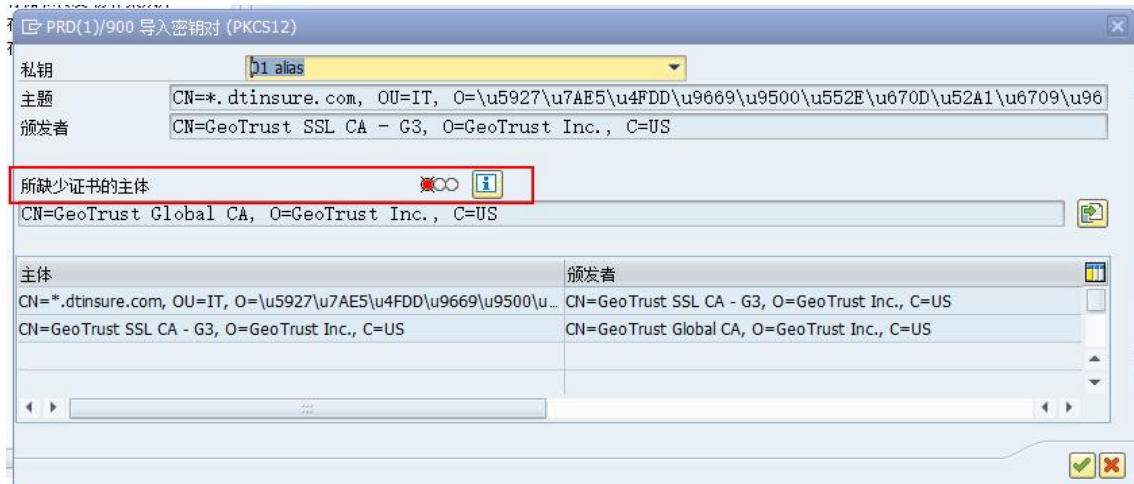
选择 pfx 个人信息交换文件



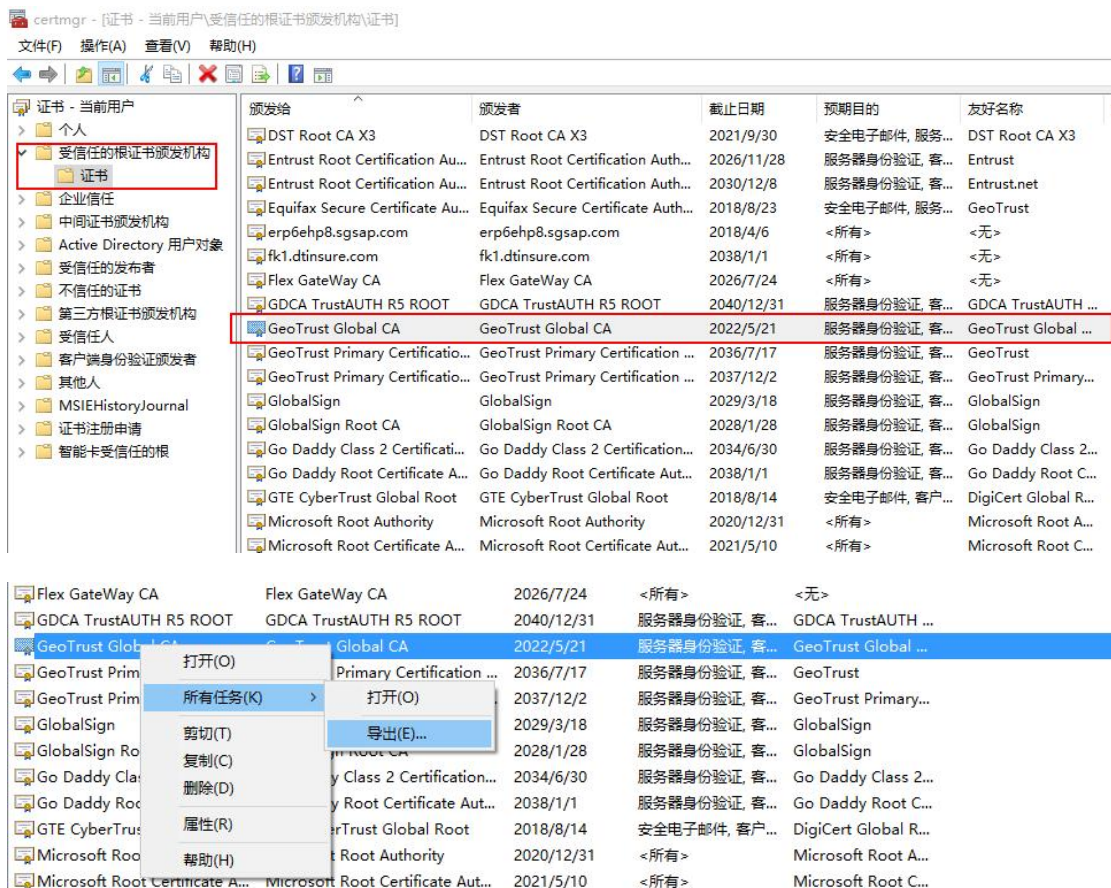
输入 pfx 密码



显示如下，会提示缺少证书主体

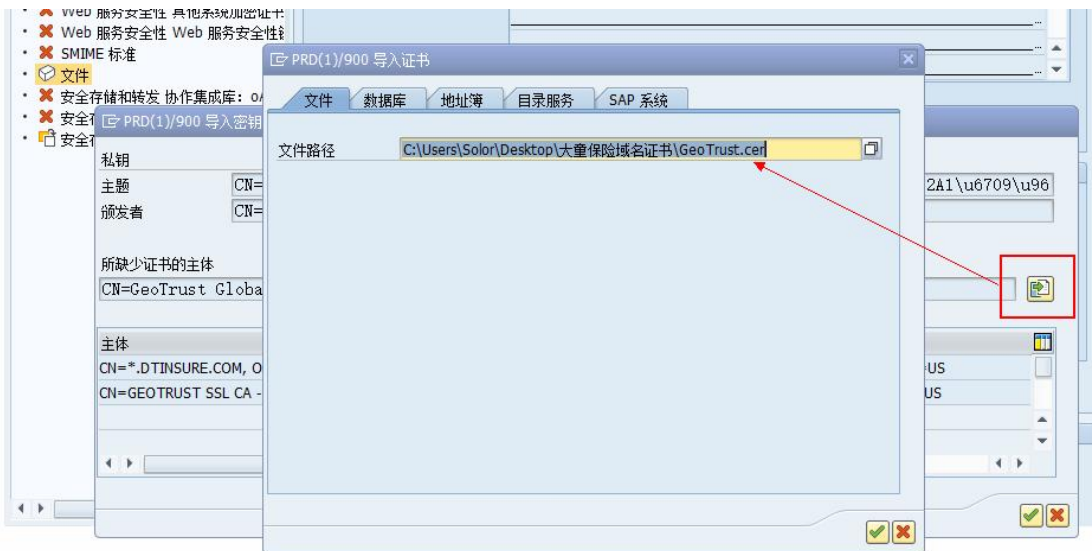


certmgr.msc 进入证书管理器，导出主体证书

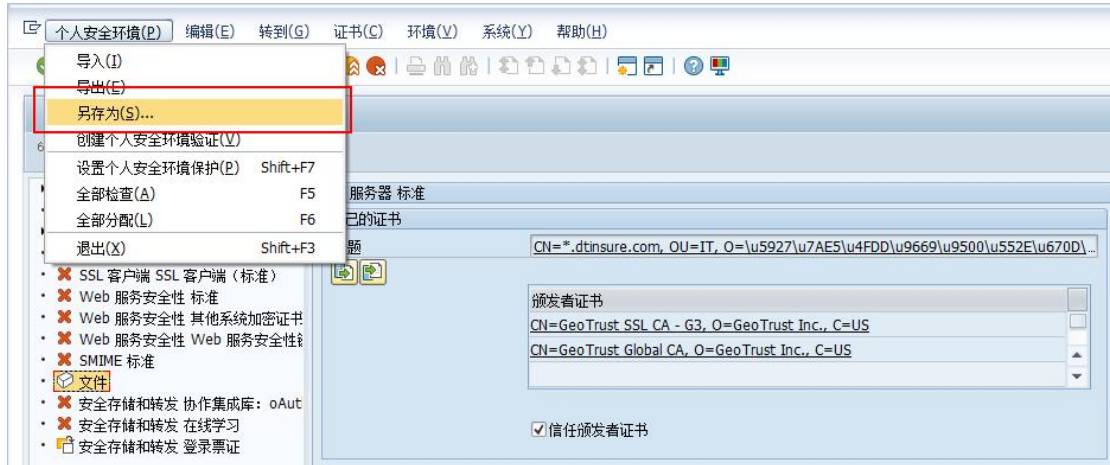




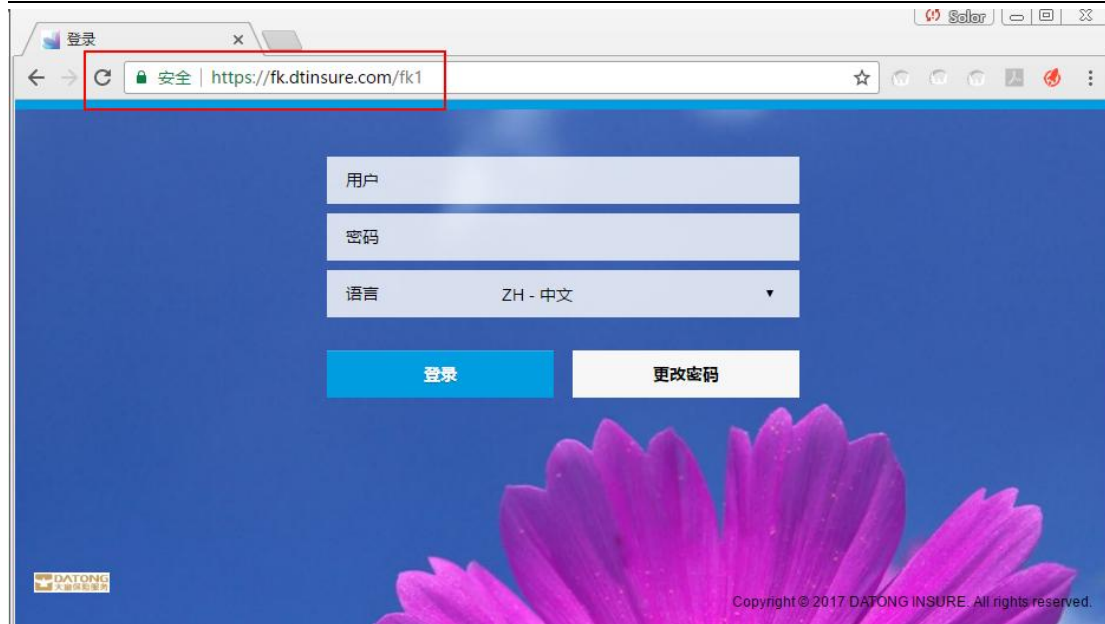
导入主体证书



另存为服务器个人安全文件

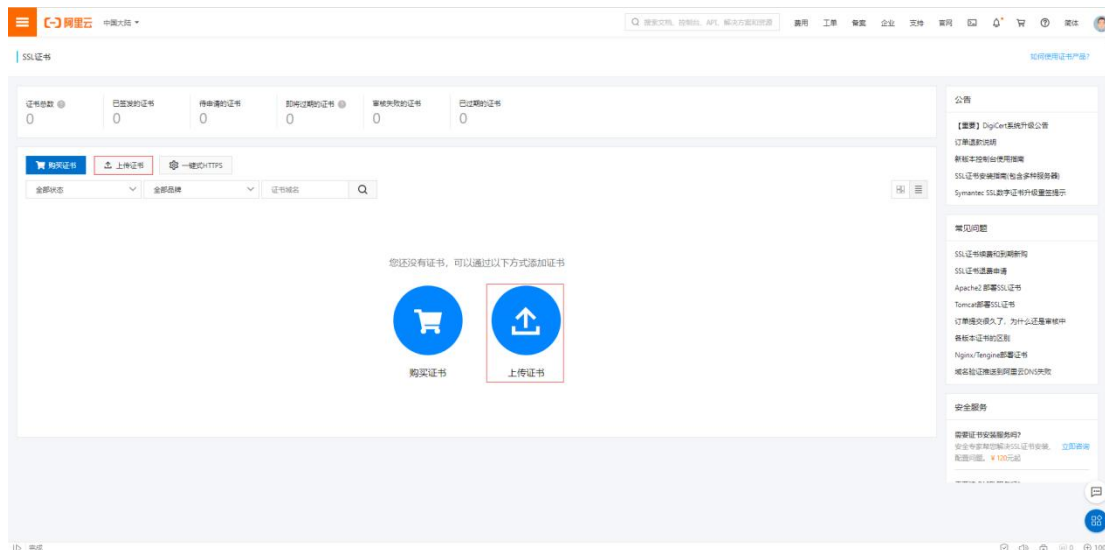


浏览器输入域名测试，显示证书安全，则证书导入成功



2.6.12 阿里云通用证书配置

1、进入 SSL 证书管理页面 <https://www.aliyun.com/product/cas>，点击 SSL 证书管理台进入证书管理页面。



2、点击上传证书按钮，输入证书名称，证书文件。证书私钥，点击确认完成证书上传。

上传证书



● 证书名称:

测试1

名称仅支持英文字母、数字、下划线、中线

● 证书文件:

```
Eq4UB11/JCVO++LRSLtDNbRJ0IRW8jcCqmHbXWBakRmew8kmrnDGouJj+HbsHRmt
XXwkxfXHy5jXqHChy333m66+9wSzfl6bZBriwleMjZMgzPtgsqgJ5t4ofu+GDaLx
a+O0gi0CE2Qhc4dv1tquZZYJ45UwdCg2Yrls/MbRK0eYhobtuRRIW9CRtv6LCH+z
NGq2wvxSEFqN5ljl5/LRb/FU5Ptlnm9QjqwlPnZbNs32i7YWvkG0CjI2CaR+bbx
A+IW3PFch5b5jb5aEHsYMs3SD8yvCtyZXs2Snw9XoA==
-----END CERTIFICATE-----
```

● 证书私钥:

```
rYnLsB1oaoGHWIE+/bt6xQlxJIsPeJv0TBrUGXEBmp2DZ38xRA6GAdnbajUjJMOv
rYJrhGGDGWVAwvmfT5dbCfLj31X8OANuWOVfdqTLtN/nVa6s+h/wUbol7xheshz1
bTXWfQKBgAMtcwvWYkyJNvVe3cNTSckUkx2aDx58H1H7YkzBM0yJt/7XO93yG8A
tY/CM0AktC68DCOMN6uiXUASkmKamdQUhe73XVt2tYgQXg9T6XJReXgRGgD6LFu
KQOfkFakOYIshR/8Zfzvtx51q5Rha1YXR2Ry1bR4sMW/ZP159qug
-----END RSA PRIVATE KEY-----
```

为了更好的保护您的证书数据安全，上传的证书不支持下载！

确定

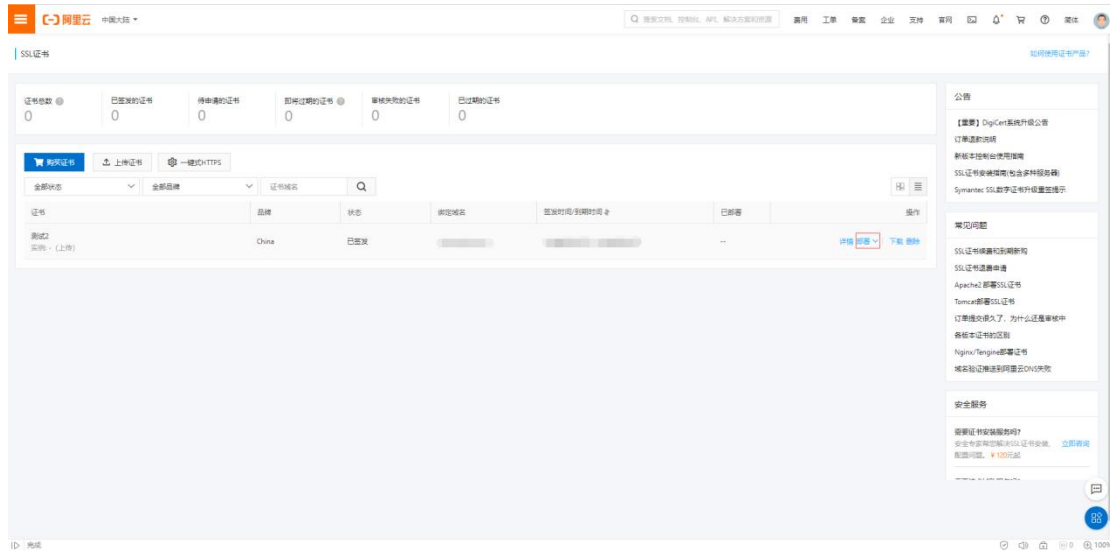
取消

证书名称：名称仅支持英文字母、数字、下划线、中线

证书文件：公钥证书，即 CFCA 邮件中：单位名称.cer 文件内容

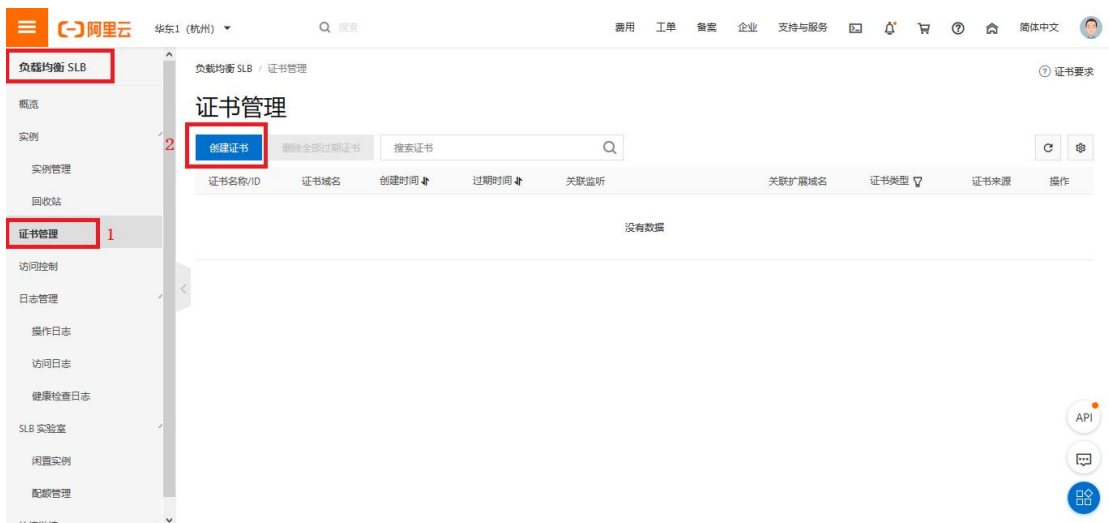
证书私钥：生成 csr 时产生的 key 文件。如果有密码，需要去除密码后上传 pem 文件。如果是 jks 或者 pdf 文件，见本文档 3.5 章节证书格式转化。

3、选择证书，点击部署根据需求部署到对应网站即可

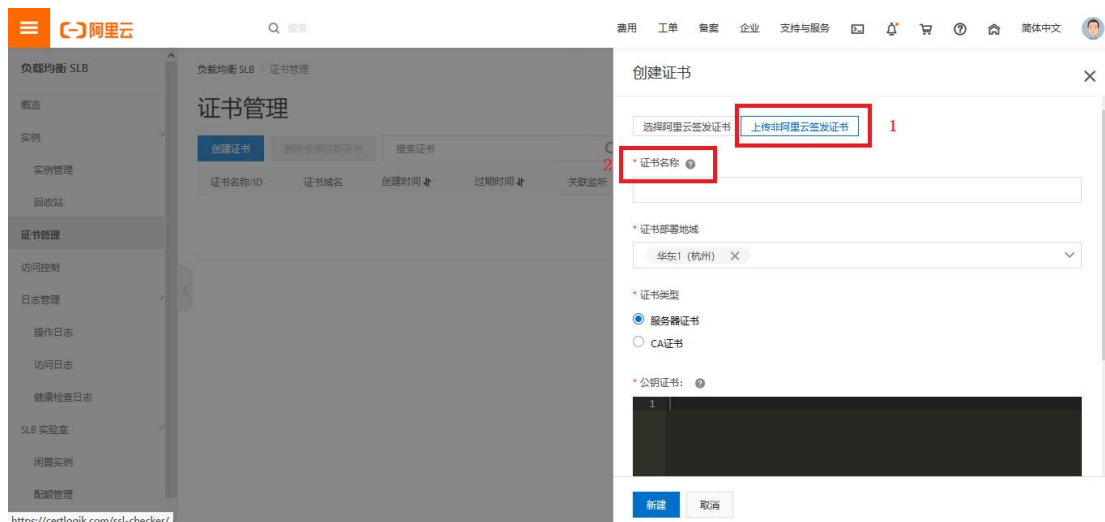


2.6.13 阿里云 SLB 证书配置

- 1、以下地址进入 SLB: <https://slbnew.console.aliyun.com/slb>。
- 2、负载均衡 SLB—证书管理—创建证书



- 3、上传非阿里云签发证书—填写证书名称



4、证书类型选择—服务器证书，填写公钥证书内容及私钥



说明：

公钥证书：此处包括 SSL 证书及中级证书，格式如下：

-----BEGIN CERTIFICATE-----

SSL 公钥证书(BASE64 编码)—CFCA 邮件中：单位名称.cer 文件内容

-----END CERTIFICATE-----

!!!中间不可有空行!!!

-----BEGIN CERTIFICATE-----

中级公钥证书(BASE64 编码)—CFCA 邮件中：CFCA_OV_OCA.cer (OV 证书添加此文件内容)
CFCA_EV_OCA.cer (EV 证书添加此文件内容)

-----END CERTIFICATE-----

私钥：与 CSR 同时产生保存的.key 文件，格式如下：

-----BEGIN RSA PRIVATE KEY-----

证书私钥(BASE64 编码)

-----END RSA PRIVATE KEY-----

5、证书类型选择—CA 证书



CA 证书：CFCA 根证书，格式如下

-----BEGIN CERTIFICATE-----

CA 公钥证书(BASE64 编码) -- CFCA 邮件中：CFCA_EV_ROOT.cer 文件内容

-----END CERTIFICATE-----

6、完成证书上传

7、进行实例管理，依次：实例管理—监听配置向导



8、配置 HTTPS



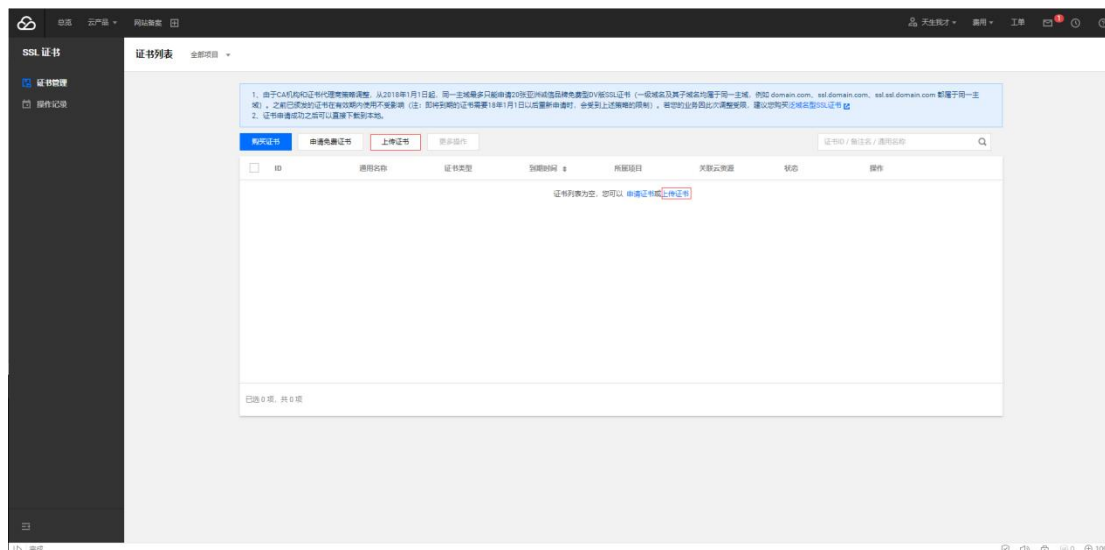
9、选择上传的证书



依次完成：后端服务器—健康检查—配置审核等操作

2.6.14 腾讯云证书配置

1、进入 SSL 证书管理页面 <https://console.cloud.tencent.com/ssl>



2、点击上传证书按钮，输入证书，私钥证书，点击确认完成证书上传。

上传证书

上传后腾讯云将保管您第三方申请的SSL证书，安全可靠。

证书

```
a+0UgiUCEZQnc4dvItquZZYJ45UwdCgZYris/MDRKDeYnobtURKIW9CRt
v6LCH+z
NGq2wvxSEFqN5ljcl5/LRb/FU5PtInm9QjqwIPnZbNs32i7YWvkG0Cjl2C
aR+bbx
A+IW3PFch5b5jb5aEHsYMs3SD8yvCtyZXs2Snnw9XoA==
---END CERTIFICATE---
```

请输入证书内容（包含证书链）

私钥

```
BTXWfQKBgAMtcwVvYkyJNVVe3cNTScKcURXZaDX58HtH/YR2BMOyJt
/7XO93yG8A
tY/CM0AktC68DCOMN6uiXIJASkmKamdQUhe73XVt2tYgQXg9T6XJRe
XgRGgD6LFu
KQOfKfakOYlshR/8Zfzvtx51q5Rha1YXR2Ry1bR4sMW/ZP159qug
---END RSA PRIVATE KEY---
```

所属项目

默认项目

上传

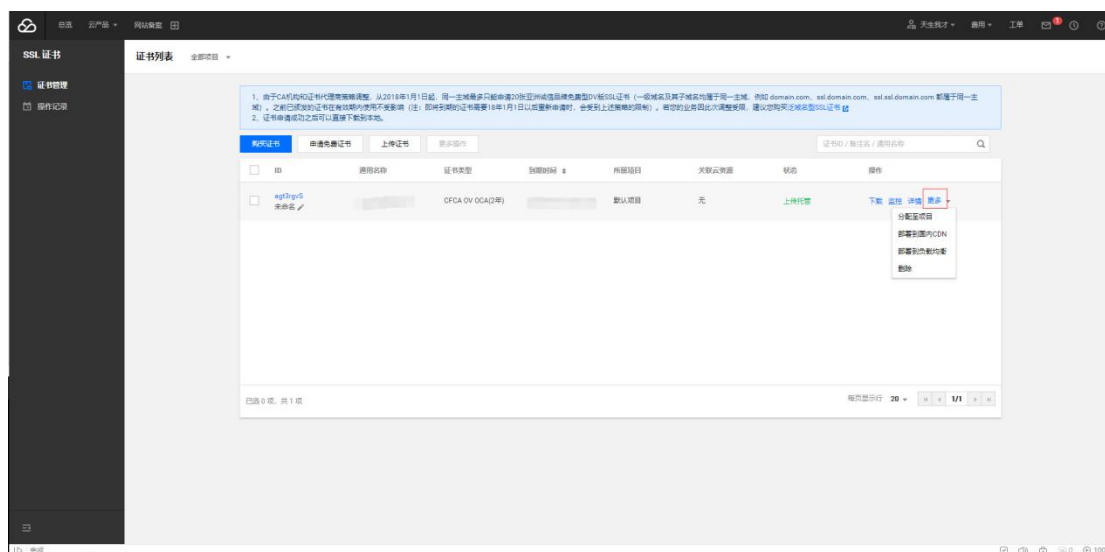
取消

证书：证书公钥，即 CFCA 邮件中：单位名称.cer 文件内容

证书私钥：生成 csr 时产生的 key 文件。如果有密码，需要去除密码后上传

pem 文件。如果是 jks 或者 pdf 文件，见本文档 3.5 章节证书格式转化。

3、选择证书，点击更多根据需求部署到对应网站即可



2.6.15 金蝶 Apusic 证书配置 1

Apusic 应用服务器默认配置下不支持双向认证，要支持 SSL 双向认证，需按照其中一种方式进行修改

(一) 采用密钥库文件的方式

修改 Apusic-AS-6.0-OS-Independent/domains/mydomain/config 目录下的 apusic.conf 文件：

```
<SERVICE
  CLASS="com.apusic.net.Muxer">
    <ATTRIBUTE NAME="Port" VALUE="6888"/>
    <ATTRIBUTE NAME="Backlog" VALUE="50"/>
    <ATTRIBUTE NAME="Timeout" VALUE="300"/>
    <ATTRIBUTE NAME="MaxWaitingClients" VALUE="200"/>
    <ATTRIBUTE NAME="WaitingClientTimeout" VALUE="5"/>
    <ATTRIBUTE NAME="SSLEnabled" VALUE="True"/>
    <ATTRIBUTE NAME="SecurePort" VALUE="6889"/>
    <ATTRIBUTE NAME="MutualAuthPort" VALUE="6887"/>
    <ATTRIBUTE NAME="NeedClientAuth" VALUE="True"/>
    <ATTRIBUTE NAME="KeyStore" VALUE="config/test.jks"/>
    <ATTRIBUTE NAME="KeyPassword" VALUE="000000"/>
    <ATTRIBUTE NAME="KeyStoreType" VALUE="JKS"/>
    <ATTRIBUTE NAME="TrustStore" VALUE="config/trust.jks"/>
    <ATTRIBUTE NAME="TrustStorePassword" VALUE="000000"/>
    <ATTRIBUTE NAME="TrustStoreType" VALUE="JKS"/>
</SERVICE>
```

其中：

SSLEnabled：是否启用 SSL；

MutualAuthPort：双向 SSL 端口；

NeedClientAuth：是否需要客户端证书；

KeyStore、KeyPassword、KeyStoreType：服务器证书路径、密码、类型；

TrustStore、TrustStorePassword、TrustStoreType：信任证书路径、密码、类型。

访问管理控制台：<https://hostname:6887/admin>，状态栏上的小锁处于闭合状态，且证书为有效状态，表示您已经成功地与服务器建立了要求客户端验证的 SSL 安全连接。

.....

(二) 证书文件配置证书方式

1. 在对应的域(如 mydomain)目录创建目录，如 cal
2. 把相关的文件 (ca-cert.cer, server2.cer, serverkey) 拷贝到 cal 目录下
3. 修改 Muxer 服务，如果采用证书文件的方式，示例配置如下：<SERVICE CLASS="com.apusic.net.Muxer">

```
<ATTRIBUTE NAME="SSLEnabled" VALUE="True"/>
<ATTRIBUTE NAME="SecurePort" VALUE="6889"/>
<ATTRIBUTE NAME="MutualAuthPort" VALUE="6887"/>
<ATTRIBUTE NAME="NeedClientAuth" VALUE="True"/>
<ATTRIBUTE NAME="ServerCertificateKey" VALUE="cal/server.key"/>
<ATTRIBUTE NAME="ServerCertificateChain"
VALUE="cal/server2.cer;cal/ca-cert.cer"/>
<ATTRIBUTE NAME="TrustCertificates" VALUE="cal/ca-cert.cer"/>.....
</SERVICE>
```

2.6.16 金蝶 Apusic 证书配置 2

1、拷贝证书。进入 jetty 服务器目录下的 etc，新建存放 jks 格式证书的目录，并复制 jks 格式证书至当前目录。

```
# pwd
/opt/jetty9222/etc
# mkdir cert
# cd cert/
# cp ../../../../keys/jetty.jks .
# ls
jetty.jks
```

```
root@iZ8vbe7yd3ei8tupg44md8Z:/opt/jetty9222/etc# pwd
/opt/jetty9222/etc
root@iZ8vbe7yd3ei8tupg44md8Z:/opt/jetty9222/etc# mkdir cert
root@iZ8vbe7yd3ei8tupg44md8Z:/opt/jetty9222/etc# cd cert/
root@iZ8vbe7yd3ei8tupg44md8Z:/opt/jetty9222/etc/cert# cp ../../../../keys/jetty.jks .
root@iZ8vbe7yd3ei8tupg44md8Z:/opt/jetty9222/etc/cert# ls
jetty.jks
```

2、编辑 jetty 服务器目录中的 etc 中的 jetty-ssl.xml，设置证书相关参数（

```
<?xml version="1.0"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN" "http://www.eclipse.org/jetty/configure_9_0.dtd">

<!-- Configure a TLS (SSL) Context Factory -->
<!-- This configuration must be used in conjunction with jetty.xml -->
<!-- and either jetty-https.xml or jetty-spdy.xml (but not both) -->
<!-- ----->
<Configure id="sslContextFactory" class="org.eclipse.jetty.util.ssl.SslContextFactory">
  <Set name="KeyStorePath"><Property name="jetty.base" default="." /></Property name="jetty.keystore" default="etc/cert/jetty.jks"/></Set>
  <Set name="KeyStorePassword"><Property name="jetty.keystore.password" default="214362464370691"/></Set>
  <Set name="TrustStorePath"><Property name="jetty.base" default="." /></Property name="jetty.truststore" default="etc/cert/jetty.jks"/></Set>
  <Set name="TrustStorePassword"><Property name="jetty.truststore.password" default="214362464370691"/></Set>
  <Set name="EndpointIdentificationAlgorithm"></Set>
  <Set name="NeedClientAuth"><Property name="jetty.ssl.needClientAuth" default="false"/></Set>
  <Set name="WantClientAuth"><Property name="jetty.ssl.wantClientAuth" default="false"/></Set>
  <Set name="ExcludeCipherSuites">
    <Array type="String">
      <Item>SSL_RSA_WITH_DES_CBC_SHA</Item>
      <Item>SSL_DHE_RSA_WITH_DES_CBC_SHA</Item>
      <Item>SSL_DHE_DSS_WITH_DES_CBC_SHA</Item>
      <Item>SSL_RSA_EXPORT_WITH_RC4_40_MD5</Item>
      <Item>SSL_RSA_EXPORT_WITH_DES40_CBC_SHA</Item>
      <Item>SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA</Item>
      <Item>SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA</Item>
    </Array>
  </Set>
</Configure>

<!-- ----->
<!-- Create a TLS specific HttpConfiguration based on the -->
<!-- common HttpConfiguration defined in jetty.xml -->
```

3、编辑 jetty 服务器目录中的 etc 中的 jetty-https.xml，配置 https 所使用的 443 端口。

```
<Call id="httpsConnector" name="addConnector">
  <Arg>
    <New class="org.eclipse.jetty.server.ServerConnector">
      <Arg name="server"><Ref refid="Server" /></Arg>
      <Arg name="acceptors" type="int"><Property name="ssl.acceptors" default="-1"/></Arg>
      <Arg name="selectors" type="int"><Property name="ssl.selectors" default="-1"/></Arg>
      <Arg name="factories">
        <Array type="org.eclipse.jetty.server.ConnectionFactory">
          <Item>
            <New class="org.eclipse.jetty.server.SslConnectionFactory">
              <Arg name="next">http/1.1</Arg>
              <Arg name="sslContextFactory"><Ref refid="sslContextFactory"/></Arg>
            </New>
          </Item>
          <Item>
            <New class="org.eclipse.jetty.server.HttpConnectionFactory">
              <Arg name="config"><Ref refid="sslHttpConfig"/></Arg>
            </New>
          </Item>
        </Array>
      </Arg>
      <Set name="host"><Property name="jetty.host" /></Set>
      <Set name="port"><Property name="https.port" default="443"/></Set>
      <Set name="idleTimeout"><Property name="https.timeout" default="30000"/></Set>
      <Set name="soLingerTime"><Property name="https.soLingerTime" default="-1"/></Set>
      <Set name="acceptorPriorityDelta"><Property name="ssl.acceptorPriorityDelta" default="0"/></Set>
      <Set name="selectorPriorityDelta"><Property name="ssl.selectorPriorityDelta" default="0"/></Set>
      <Set name="acceptQueueSize"><Property name="https.acceptQueueSize" default="0"/></Set>
    </New>
  </Arg>
</Call>
```

4、编辑 jetty 服务器目录中的 start.ini 文件，按需求更改端口号，并设置启动加载 jetty-https.xml, jetty-ssl.xml

```
jetty.port=80
jetty.dump.stop=
etc/jetty-ssl.xml
etc/jetty-https.xml
```

5、重启 jetty，验证 https 访问是否正常。

附录一、CFCA 全球信任证书（SSL 证书）申请表及授权书

申请表

CFCA 全球信任服务器证书申请表						
证书申请信息	申请日期		证书数量		证书期限	
	业务类型	<input type="checkbox"/> 新申请 <input type="checkbox"/> 更新 <input type="checkbox"/> 吊销				
	证书类型	OV 证书	<input type="checkbox"/> 单域名 OV 服务器证书		<input type="checkbox"/> 通配符 OV 服务器证书	
			<input type="checkbox"/> 多域名 OV 服务器证书			
		EV 证书	<input type="checkbox"/> 单域名 EV 服务器证书			
			<input type="checkbox"/> 多域名 EV 服务器证书			
	域 名					
		注： 1、多域名证书，默认以 CSR 中填写的域名为证书主域名，其他作为备用域名 2、通配符证书，适用以*开头的域名，例如*.domain.com 3、IP 类型只限于申请公网 IP，且只可申请 OV 单域名或者多域名（多个 IP）证书 4、该处直接填写域名即可，不需要添加 http://或者 https://				
域名验证方式	<input type="checkbox"/> 邮箱验证		<input type="checkbox"/> DNS 验证	<input type="checkbox"/> 文件验证		
	注： 1、采用邮箱验证方式时，请确保 whois 隐私保护关闭，whois 中管理员邮箱可用（若开启隐私保护，我方无法查询明确的管理员邮箱，则默认向 admin、administrator、webmaster、hostmaster、postmaster 开头的域名邮箱发送验证邮件，例如 admin@domain.com 形式，请确认上述邮箱可正常回复邮件后，再选择此种验证方式） 2、采用 DNS 或文件验证方式时，需要域名管理员在域名解析服务商处的域名管理系统操作，记录值或文件 CFCA 会发送至本表格中经办人邮箱，按照邮件提示操作					
申请企业/机构信息区（以下信息全部填写，不可留白）						
机构信息	机构名称 (中文全称)					
	机构证件类型	<input type="checkbox"/> 企业营业执照 <input type="checkbox"/> 组织机构代码证 <input checked="" type="checkbox"/> 其它，请注明： 统一社会信用代码				
	机构证件号码				联系电话	

	联系地址					邮政编码	
申请 经办人	姓名		职务		电子邮件		
	证件类型		证件号		联系电话		
申请 确认人	姓名		职务		电子邮件		
	证件类型		证件号		联系电话		
申请 声明	本人/机构授权本表格中经办人办理证书申请相关事宜，并承诺以上信息资料真实、有效。本人/机构已认真阅读并同意遵守中金金融认证中心有限公司（CFCA）网站（ http://www.cfca.com.cn ）发布的《数字证书服务协议》、《全球信任体系电子认证业务规则（CPS）》中规定的相关义务。						
	申请机构盖章					日期	
	备注						
<p>申请材料说明：</p> <p>1、申请表（加盖企业公章或带有公司名称字样的部门公章） *</p> <p>2、CSR（CSR 中信息需要与申请表中一致） *</p> <p>3、机构证件复印件</p> <p>4、经办人身份证复印件</p> <p>4、加盖公章的域名证书（若选择邮箱、DNS 或者文件验证方式，此文件不需要提供）</p> <p>5、公网 IP 证明函（仅当以公网 IP 申请时需要，运营商出具的加盖公章的公网 IP 分配证明文件）</p>							

授权书

授 权 书

兹授权_____*****有限公司_____, 企业统一社会
信用代码_____91110000*****_____, 办理中金金融认证中心
有限公司（CFCA）证书申请事宜。

授权期限：_____2023_____年_____10_____月_____1_____日
至_____2024_____年_____10_____月_____1_____日

单位名称： *****政务服务中心

日期： 2023 年 9 月 30 日

附录二、CFCA 域名验证方式

目前 CFCA 支持邮箱验证、DNS 验证、文件验证三种域名验证方式，本文介绍常用三种方法。

注意事项：域名验证记录值有效期为 15 天，自生成时开始计算。请务必在 15 天内完成配置，如超时未进行配置或验证未通过，请联系 CFCA 工作人员，重新申请域名验证记录值并配置。

方法一：DNS 验证

下文介绍 SSL 证书 DNS 验证在各主流域名注册商下的域名解析方法，仅供参考，具体以各注册商实际为准。CFCA 会将 DNS 记录值发送到证书申请经办人邮箱，请留意查收。

DNS 验证注意事项：

当申请的域名不为主域名（如：domain.com），为二级域名时（如：www.domain.com），主机记录值需更新为：“_cfcachallenge.host.二级域名前缀”，即：_cfcachallenge.host.www

阿里云操作示例：

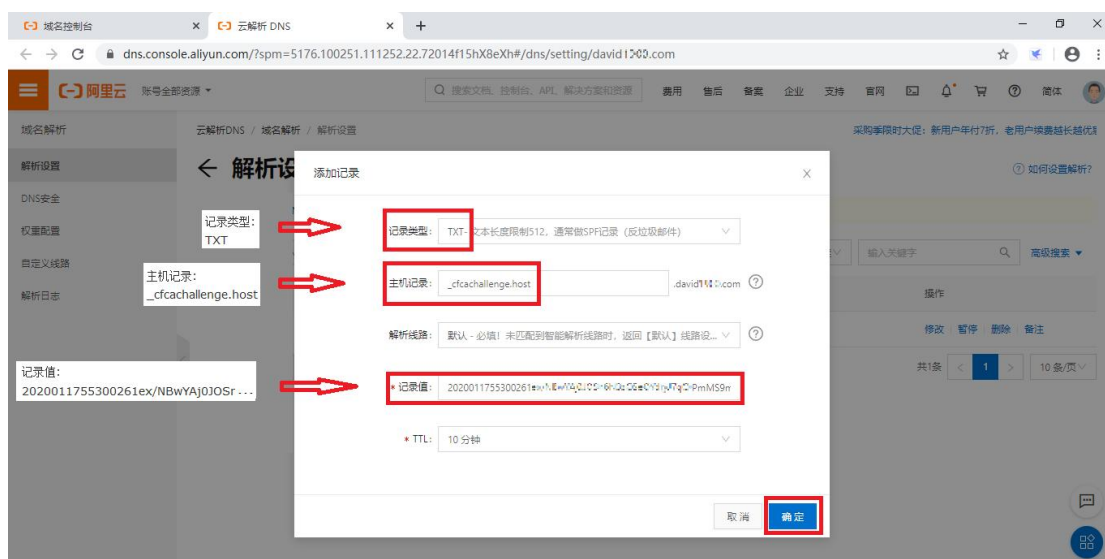
（1）登陆域名管理控制台，查看【域名列表】，单击操作栏的【解析】，进入域名解析页面：



（2）单击【添加记录】



(3) 添加记录类型为 TXT 的 DNS 记录，单击【确定】完成添加

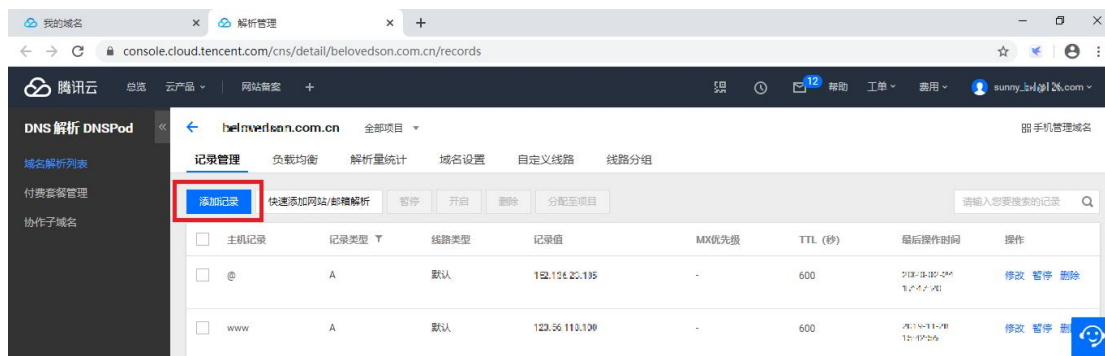


腾讯云操作示例：

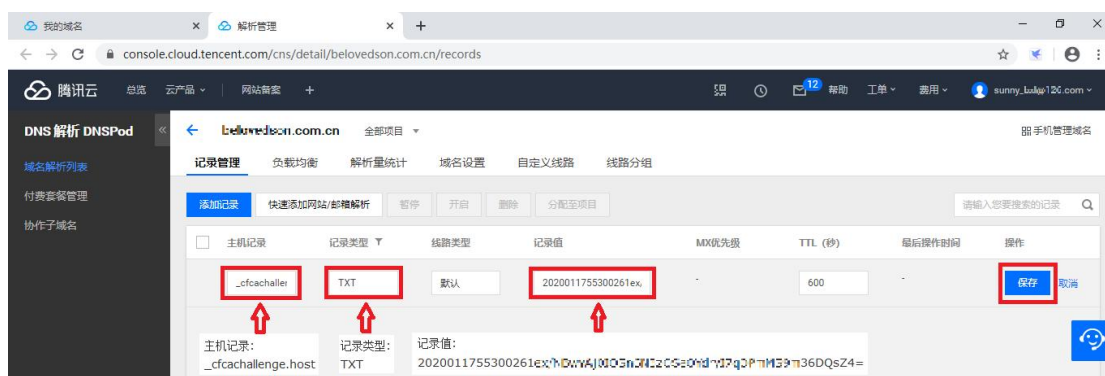
(1) 登陆域名管理控制台，查看【我的域名】，单击操作栏的【解析】，进入域名解析页面：



(2) 单击【添加记录】



(3) 添加记录类型为 TXT 的 DNS 记录，单击【保存】完成添加



新网操作示例：

将记录类型选择为 TXT 记录，在主机记录中输入邮件中提供的主机记录字段信息，不包括网址信息，在记录值中输入邮件中的记录值字段信息，点击添加



方法二：文件验证

选择文件方式验证后，CFCA 会发送记录值至证书申请经办人邮箱：

操作步骤

1、创建文件：

本地创建名称为“**cfcafileauth.txt**”的 TXT 文件，将邮件中“文件内容”字段，

复制到上述文件，保存（请不要增加空格等其他多余信息）；

2、创建目录：

在站点根目录下创建 `/.well-known/pki-validation` 子目录，然后将 `cfcafileauth.txt` 文件上传至该目录；

注：

(1) 第一层目录是带点的隐藏目录，Windows 下命令为：`mkdir .well-known`

```
Microsoft Windows [版本 10.0.17134.1099]  
(c) 2018 Microsoft Corporation。保留所有权利。
```

```
C:\Users\thinkpad>cd C:\inetpub\wwwroot
```

```
C:\inetpub\wwwroot>mkdir .well-known
```

(2) 如果您的站点由于某种原因无法创建隐藏目录，请选择 DNS 验证方式

3、域名解析至服务器

4、配置检测：

配置好之后，可通过浏览器访问地址，如正常输出配置的记录值，则表示配置成功。

(1) HTTP 配置检测：`http://您的域名/.well-known/pki-validation/cfcafileauth.txt`

(2) HTTPS 配置检测：`https://您的域名/.well-known/pki-validation/cfcafileauth.txt`

若申请 `*.domain.com` 类型的通配符证书时，访问检测地址为：

(1) HTTP 配置检测：`http://domain.com/.well-known/pki-validation/cfcafileauth.txt`

(2) HTTPS 配置检测：`https://domain.com/.well-known/pki-validation/cfcafileauth.txt`

注意事项：

(1) HTTP、HTTPS 任选其一验证通过即可，HTTP 方式需使用 80 端口，HTTPS 方式需使用 443 端口；

(2) 文件验证需要直接响应 200 状态码和文件内容，不支持任何形式的跳转。


方法三： 邮箱验证

邮箱验证，即通过 Whois 查询域名注册时预留的邮箱，CFCA 向该注册邮箱发送 SSL 证书申请确认信息，若 CFCA 受到确认邮件，则可证明该邮箱被合法持有人控制，验证通过后可为其颁发服务器证书。


采用邮箱验证方式时，请确保 whois 隐私保护关闭，whois 中管理员邮箱可正常回复邮件（若开启隐私保护，我方无法查询明确的管理员邮箱，则默认向 `admin`、`administrator`、`webmaster`、`hostmaster`、`postmaster` 开头的域名邮箱发送验证邮件，例如 `admin@domain.com` 形式，请确认上述邮箱可正常回复邮件后，再选择此种验证方式。

Whois 邮箱查询地址:

<https://www.whois.com/whois/>

 **Domain Information**

Domain:	████████.cn
Registrar:	北京新网数码信息技术有限公司
Registered On:	1999-05-18
Expires On:	2022-06-18
Status:	ok
Name Servers:	████████.nsod.net ████████.nsod.net

 **Registrant Contact**

Organization:	████████████████████
Email:	████████████████████

若Domain、ExpiresOn、Organization信息公开可查，未超过有效期，上述信息与申请表中信息一致，则可以不再重复做域名验证，无需提交其他域名验证材料

附录三、CFCA 全球信任根证书获取方式

Windows Vista、Windows 7

Windows Vista 以及更高版本的操作系统，Windows 通过根证书自动更新机制分发 CFCA 全球信任根证书。即，用户访问含有 CFCA 全球信任证书的网站、读取含有 CFCA 全球信任证书的安全电子邮件、执行含有 CFCA 全球信任证书代码签名的 ActiveX 控件及可执行程序时，Windows 证书链验证程序访问 Microsoft 根证书信任列表，自动下载 CFCA 全球信任根证书，并将其安装在用户 Windows 受信任根证书颁发机构存储区。整个过程自动完成，用户不会看到任何安全性对话框或警告。有关 Windows Vista、Windows 7 根证书更新的详细技术信息，请访问以下网站：

[http://technet.microsoft.com/en-us/library/cc749331\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc749331(WS.10).aspx)

Windows XP

Windows XP 含有更新根证书组件（控制面板——添加或删除程序——添加/删除 Windows 组件），当用户访问含有 CFCA 全球信任证书的网站、读取含有 CFCA 全球信任证书的安全电子邮件、执行含有 CFCA 全球信任证书代码签名的 ActiveX 控件及可执行程序时，更新根证书组件将联系的微软 Windows Update 站点检查根证书信任列表，自动下载 CFCA 全球信任根证书，并将其安装在用户 Windows 受信任根证书颁发机构存储区。有关 Windows XP 根证书更新的详细技术信息，请访问以下网站：

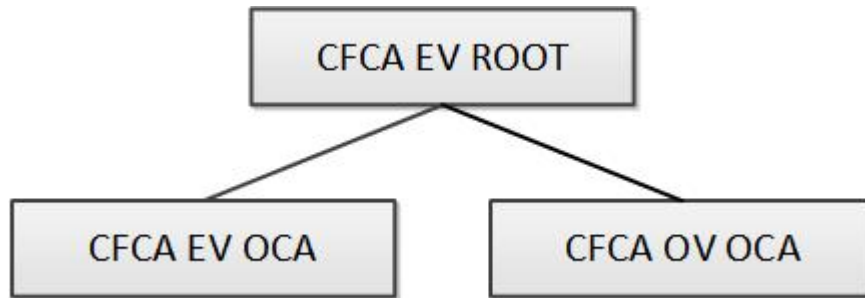
<http://technet.microsoft.com/en-us/library/bb457160.aspx>

Firefox 浏览器

CFCA EV 根证书已经内置在 Firefox 浏览器中，包括 Windows、Linux、Mac OS X、Android 等平台。用户升级到 Firefox v38.0 及以上版本，即可获取 CFCA EV 根证书。

附录四、CFCA 全球信任证书链

CFCA 全球服务器证书体系如下：



主题：CN = CFCA EV ROOT

O = China Financial Certification Authority

C = CN

序列号：18 4a cc d6

有效期：2012 年 8 月 8 日 11:07:01——2029 年 12 月 31 日 11:07:01

摘要算法：SHA256

密钥长度：RSA（4096Bits）

-----BEGIN CERTIFICATE-----

```

MIIFjTCCA3WgAwIBAgIEGEm1jANBgkqhkiG9w0BAQsFADBWMQswCQYDVQQGEwJD
TjEwMC4GA1UECgwnQ2hpbmEgRmluYW5jaWFsIEN1cnRpZm1jYXRpb24gQXV0aG9y
aXR5MRUwEwYDVQQDDAxDRkNBIEVWIFJPT1QwHhcNMTIwODA4MDMwNzAxWhcNMjky
MjEwMDMwNzAxWjBWMQswCQYDVQQGEwJDTjEwMC4GA1UECgwnQ2hpbmEgRmluYW5j
aWFsIEN1cnRpZm1jYXRpb24gQXV0aG9yaXR5MRUwEwYDVQQDDAxDRkNBIEVWIFJP
T1QwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDXXWvNED8fBVnVBU03
sQ7smCuOFR36k0sXgiFxEFLXUWRwFsJVaU20FW2fvwwbwuCjZ9YMrM8irq93VCpL
TIpTUUnrD7i7es3ElweldPe6hL6P3KjzJIx1qqx2hp/Hz7KDVRM8Vz3IvHWOX6Jn5
/Z0kVIBMuTrSqy5J35DNuF++P96hyk0glCXohC1Tt7GIH//62pCfCqktQT+x8Rgp
7hZZLDRJGqG16iIOgNyejLi6mhNbiyWZXvKWfry4t3uMCz7zEasxGPrb382KzRz
EpR/38wmnvFyXVB1WY9ps4deMm/DGIq1lY+wejfeWkU7xzbb72fROdOXW3NiGUgt
  
```


hxwG+3SYIElZ8AXSG7Ggo7cbcNOIabla1jj0Ytwli3i/+Oh+uFzJlU9fpy25IGvP
a931DfSCt/SyZi4QKPaXWnuWFO8BGS1sbn85WAZkgwGDg8NNkt0yxoeK+NkWzqot
aK8KgWU6cMGbrU1tVMoQLUuFG70A5nBFDWteNfB/07ic5ARwiRIIk9oKmSJgamNg
TnYGmE69g60dWIo1hdLHZR4tjsbftsbfh4oEIRUpdPA+nJCdDC7xi j5aqgwJHsfV
PKPt18MeNPo4+Qg048BdK4PRVmrJtqhUUy54Mmc9gn900PvhtgVguXDbjgv5E1hv
cWAQUhC5wUEJ73IfZzF4/5YFjQIDAQABo2MwYTAfBgNVHSMEGDAWgBTj/i39KNAL
tbq2osS/BqoFjJP7LzAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBBjAd
BgNVHQ4EFgQU4/4t/SjQC7W6tqLEvwaqBYyT+y8wDQYJKoZIhvcNAQELBQADggIB
ACXGumvrh8vegjmWPfBEp2uEcwPenStPuiB/vHiyz5ewG5zz13ku9Ui20vsXiObT
ej/tUxPQ4i9qecsAIyjmHjdXNYmEwnZPNdatZ8PQQaIxfFu2Bq41gt/UP+TqhdL
jOztUmCypAbqTuv0axn96/Ua4CUqmtzHQtb3yHQFhDmVodYLO6Qn+gjYXB74BGBS
ESgoA//vU2YApUo0FmZ8/Qmkrp5nGm9BC2sGE5uPhnEFtC+NiWYzKXZUmhH4J/qy
P5Hgzg0b8zAarb8iXRvTvyUFTeGSGn+ZnzxEk8rUQE1sgIfXBDrDMI1ID1b4pd19
xIsNER9Tyx6yF7Zodlrg1MvIB6710i60N7fQAUtDKXeM0ZePglr4UeWJoBjnaH9d
Ci77o0cOPaYjesYBx4/IXr9tgFa+iiS6M+qf4TIRnvHST4D2G0Cv0J4RUH1zEhLN
5mydLIhyPDCBBpEi6lmt2hkuIsKnuYyH4Ga8cyNfIWRjgEj1oDwYPZTISEEdQLpe
/v5W0aHIz16eGWRGENoXkbcFgKyLmZJ956LYBws2J+dIeWCKw9cTXPhyQN9Ky8+Z
AAoACxGV21ZFA4gKn2fQ1XmxqI1AbQ3CekD6819kR5LLU7m7Wc5P/dAVUwHY3+vZ
5nbv0C070615s9UCKc2Jo5YPSjXnTkLAdc0Hz+Ys63su

-----END CERTIFICATE-----

主题: CN = CFCA EV OCA

O = China Financial Certification Authority

C = CN

序列号: 00 b4 cf 94 32 66

有效期: 2012 年 8 月 8 日 14:06:31——2029 年 12 月 29 日 14:06:31

摘要算法: SHA256

密钥长度: RSA (2048Bits)

-----BEGIN CERTIFICATE-----

MIIFTjCCAzagAwIBAgIGALTP1DJmMAOGCSqGS1b3DQEBCwUAMFYxCzAJBgNVBAYT
AkNOMTAwLgYDVQQKDCdDaG1uYSBGaW5hbmNpYWwgQ2VydG1maWNhdG1vbiBBdXRo
b3JpdHkxFTATBgNVBAMMDENGQ0EgRVYgUk9PVDAeFw0xMjA4MDgwNjA2MzFaFw0y
OTeyMjkWnJA2MzFaMFUxCzAJBgNVBAYTAkNOMTAwLgYDVQQKDCdDaG1uYSBGaW5h
bmNpYWwgQ2VydG1maWNhdG1vbiBBdXR0b3JpdHkxZDASBgNVBAMMCONGQ0EgRVYg
TONBMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA020MsGFxFQIPMKVI
oRa09rHNX41xbq8jhnbdKOMDVbxfGa3b8QTKxMcmx1R1ULfsaie0cIlaR10AUcJP
QH9ftekzh4T287xqsEAydYQHf77arWQ5nY3fR9RcoBq9pTCQbqw49S6/jHA5oPQa
EoKbFOG8zfVKp5PrckSufHMqyKo/Ez2UYT+gut36j4GYpAABuV6PbusPpjufsn9B
r9+xqgyz8ubSp1W1lqS1vQUQBhAJAH+a3NMhD0i1laGfTdWbF485a5Ni1MFGqJBa
/kLVEYwG4aoKdV9vG/NFSOLKz3QVnB7bkrLjTkuGN/zQJP0daJ3CGAzmn+Cr2ujt
XOfAYwIDAQABo4IBITCCAR0wOAYIKwYBBQUHAQEELDAqMCgGCCsGAQUFBzABhho
dHRwOi8vb2NzcC5jZmNhLmNvbS5jb19vY3NwMB8GA1UdIwQYMBaAFOP+Lf0o0Au1
uraixL8GqgWMk/svMA8GA1UdEwEB/wQFMAMBAf8wRAYDVROgBD0wOzA5BgRVHSA
MDEwLWYIKwYBBQUHAQEELDAqMCgGCCsGAQUFBzABhhoDHRwOi8vb2NzcC5jZmNh
LmNvbS5jb19vY3NwMB8GA1UdIwQYMBaAFOP+Lf0o0Au1uraixL8GqgWMk/svMA8
GA1UdEwEB/wQFMAMBAf8wRAYDVROgBD0wOzA5BgRVHSAAMDEwLWYIKwYBBQUHA
QEELDAqMCgGCCsGAQUFBzABhhoDHRwOi8vb2NzcC5jZmNhLmNvbS5jb19vY3Nw
MB8GA1UdIwQYMBaAFOP+Lf0o0Au1uraixL8GqgWMk/svMA8GA1UdEwEB/wQFMAM
BAf8wRAYDVROgBD0wOzA5BgRVHSAAY2EvU1NBL2NybDEuY3JsMA4GA1UdDwEB/w
QEAwIBBjAdBgNVHQ4EFgQUVQVji3MyVbR9d3rNH60kWxsBFd8QwDQYJKoZIhvcNAQELBQADggIBAMmFEIoCE9UNmb2BYYhT
RV12kNVucP6t683BaFTgJizIJw/ebvvTdWNTycyP5MQF1HKrIYwjvF09Rfw8+yIs
sT3JFYiqsLBswvaMr3AIuA2mTnmasvZFe6P19qitzTRkz+TL6TFailrtznzudsvn2
SeVbRix+6CsyNNMoPsRHTeZAEpkB7J3vh+ZAiv3gsIXtjtz5Y1iWWRZiPemJ/qEf
W2hDONB+T6lGcEXHDi9dIkWcC/jFT4XPM64pagAz9gEGZg1PzFBE8QMxiwaDA0ea
G010e/HW4wJl04ZzOELqZGJL1YhQ8AkBYR95NEtR9j5bWK98Lznykl dk2MDLBD2m
rIfmkVjMwEj4A8E1MXsLnWXXg41NN6gjUm2/IudK0aGqniPs5SZrN3604B3NzsaZ
dLznHH5H0+aksurjgme8RAG0A20AnRG3VXBWrxud7tOKDINLs+mxY7IR+xVZ2cwG
Cer8HnAVfKpJrbdq7vyJJkIpC1l+mLHaGgvv3IqiU4rrrl1E3NYjKG4Fk2MiYvZg
10KXA8t1YsLt8I/RcNmC2TvjZHYVE3tanbGw53TRGFk2Vq68X0kvoo0ardihrKqg
qc0gUvouORuvSqTlkQizTFH6FTUt3xuuED4dnn5N/1ijcDt0N315ovoyHOVcYiO4
drCN96LHiUoiSfYODmpXG2t1

-----END CERTIFICATE-----

主题: CN = CFCA OV OCA

O = China Financial Certification Authority

C = CN

序列号: 00 f9 df 6a df f5 64 be a6 8b 82

有效期: 2015 年 3 月 25 日 10:02:56——2029 年 12 月 25 日 10:02:56

摘要算法: SHA256

密钥长度: RSA (2048Bits)

-----BEGIN CERTIFICATE-----

MIIFdCCA2SgAwIBAgILAPnfat/1ZL6mi4IwDQYJKoZIhvcNAQELBQAwwjELMAkG
A1UEBhMCQ04xMDAuBgNVBAoMJONoaW5hIEZpbmFuY2lhbCBDZXJ0aWZpY2F0aW9u
IEF1dGhvcml0eTEVMBMGGA1UEAwwMQ0ZDQSBFViBST09UMB4XDTE1MDMyNTAyMDI1
Nl0XDTI1MTIyNTAyMDI1Nl0wVTELMakGA1UEBhMCQ04xMDAuBgNVBAoMJONoaW5h
IEZpbmFuY2lhbCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTEUUMB1GA1UEAwwLQ0ZD
QSBPViBPQ0EwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQN14xTy0bH
zkaeyACeq6ryfxxG5zZT1fCL41mw7sk6SVm0KNfE60Gf7W6orksrFVIbIMK+VrYp
+aYyhScq8EJT9xXBgXK2HqtpaDGOec1spJvcs+rXn9t1T789NBp3i5U+nLE9M1bR
CHSx3Hzu8p7Aeq1lou+8nZ2egaVbWFL1zC1JENupSSI9Yjbefhb06y/TVxQ0x4Zt
zwPwLcd8NUtSruldolxPbhQeCZNJMPq1GKMxhd5pDwY4mCKxDeraqhTNXu19Aef3
qyi2Ic9EXmdNPARKZJU2XTJ9FJ+DE+ChaiVfJ/VwQfM0eG1Bn/SAAav54jBmRnec
PeD6YfpuiJ8vAgMBAAGjggFKMIIBRjA4BggrBgEFBQcBAQQsMCowKAYIKwYBBQUH
MAGGHGh0dHA6Ly9vY3NwLmNmY2EuY29tLmNuL29jc3AwHwYDVROjBBgwFoAU4/4t
/SjQC7W6tqLEvwaqBYyT+y8wDwYDVROTAQH/BAUwAwEB/zBEBgNVHSAEPTA7MDkG
BFUdIAAwMTAvBggrBgEFBQcCARYjaHR0cDovL3d3dy5jZmNhLmNvbS5jb191cy91
cy0xMi5odG0wOgYDVROfBDMwMTAvO2gK4YpaHR0cDovL2Nybc5jZmNhLmNvbS5j
bi91dnJjYS9SU0EvY3JsMS5jcmwwDgYDVROPAQH/BAQDAgEGMBOGA1UdDgQWBRRm
s+/7VJWH6ay111au5n3t0tBD0TANBgNVHSUEIDAeBggrBgEFBQcDAgYIKwYBBQUH
AwQGCCsGAQUFBwMBMA0GCSqGSIb3DQEBCwUAA4ICAQDKER8qcBmZGOG8G0J670VW

OSg3UovOoc7/xz2mE+enyEcSwn/0QrL8C5DSA6nMvBMrCWEytYPofGQUXTwt1u78
GLxYNn3A/RtzcZJ+/BXIhoe3aOT4tQ+2s9vrFRfIXs4CkmqHhfYSvArokdayYmBd
78psIwS5LCUzGKS7y8UmAgoxiy7RtrVt5clwvJyeuYk1Z1l8MN1szPrmAb4HS/D
qnB+0qdhFGvf0yv7lg6/wlIAkN84cH1KNC3JvyFHaCIAyhTPgjUayUvBKFK7XwN9
utIX12L3IZX7zfxGS/J9+ZeNwyblQKmd/MKyJu9Ak6+ZMLLgjlCFkihJIn9Ur8M
2KQigz7YPDVIJj0tS7lj0QVGh88LPUnQ1fBY7RwagficS/xclIOaXhoyWzg7EcQ
/T1/04FkpMqKu0reaI5NExjAT8cKizyY2wc00XKIYri3Ewnbm+00IaYYaiQRGUR6
pzFFKxdFMbStCtI40bN+A9tB7cnBCW4vz3sAJd/OgmLF38XTa+/km3c1nQ0fhCGs
6kx2heN/DgFAc+P7ld0bo/kgGQtR6tr02gyXCFWnLMtT0+CoNOY0o3T+LbEqYeKL
W7p29G9sgHgoqLFibWNMSKG1QvevkhjUMOD/g48f/nMSYsbU++yEaLv jvRHbb50N
IPkcE28TRhQQKmDKI+DRig==
-----END CERTIFICATE-----

附录五、SHA 摘要算法介绍

安全哈希算法（Secure Hash Algorithm）主要适用于数字签名标准（Digital Signature Standard DSS）里面定义的数字签名算法（Digital Signature Algorithm DSA）。对于长度小于 2^{64} 位的消息，产生一个消息摘要。消息摘要可以用来验证数据的完整性。

SHA 家族的五个算法，分别是 SHA-1、SHA-224、SHA-256、SHA-384，和 SHA-512，后四者并称为 SHA-2。支持 SHA2 的操作系统包括：Windows 8.1、Windows 8、Windows 7、Windows Server 2012 R2、Windows Server 2012、Windows Server 2008 R2、Windows Server 2008、Windows Vista、Windows Server 2003 R2、Windows 2003 Server SP2、32 位 Windows XP SP3、64 位 Windows XP SP2。

由于 SHA1 摘要算法存在杂凑冲撞攻击，随着计算机运算能力越来越强，其安全性受到越来越严重的威胁。微软、谷歌等陆续发布了弃用 SHA1 摘要算法的时间表。微软方面，要求 CA 机构 2016 年之后不能再签发新的 SHA1 摘要算法的 SSL 站点证书。2017 年之后，Windows Vista、Windows Server 2008 及以上版本操作系统将无法访问 SHA1 摘要算法 SSL 证书的网站。谷歌方面，Chrome 浏览器将对 SHA1 摘要算法 SSL 证书的站点提示安全警告。



附录六、常见问题

1、CFCA 全球信任 SSL 证书支持的操作系统和浏览器

Windows 平台浏览器 100%支持，包括但不限于：Internet Explorer、Google Chrome、Mozilla Firefox、Opera，以及 360、搜狗、遨游、QQ、UC、猎豹、百度等国产浏览器；

Windows Phone 平台浏览器 100%支持；

Andriod（Android 6.0 Marshmallow）平台 100%支持；

Linux 平台浏览器 100%支持；

Mac OS（10.12.1 及更新版本）平台浏览器 100%支持，包括但不限于：Safari、Google Chrome、Mozilla Firefox 等；

IOS（10.1 及更新版本）平台浏览器 100%支持；

浏览器 操作系统	Internet Explorer	Mozilla Firefox	Google Chrome	Apple Safari
Windows	√	√	√	——
Unix/Linux	——	√	√	——
Mac OS 10.12.1 及更新 版本	——	√	√	√
iOS 10.1 及更 新版本	——	√	√	√
Andriod 6.0 Marshmallow	——	√	√	——
Windows Phone	√	√	√	——

√：表示该浏览器完全支持 CFCA 全球信任 SSL 证书；

×：表示该浏览器不完全支持 CFCA 全球信任 SSL 证书，当浏览器访问含有 CFCA 全球信任 SSL 证书的网站时，会有不受信任提示；

——：表示该浏览器不支持此操作系统。

2、Windows XP SP2 操作系统使用 CFCA EV SSL 证书

CFCA EV SSL 证书采用 SHA256 摘要算法，而 Windows XP SP2 操作系统并不支持该算法。在 Windows XP SP2 操作系统中，使用 IE 浏览器无法访问含有 CFCA EV SSL 证书的网站（包括使用其他 CA 机构 SHA256 摘要算法 SSL 证书的网站）。可以将操作系统升级到 Windows XP SP3 及以上版本，即可正常访问。

此外，火狐（Firefox）、谷歌（Chrome）等浏览器不依赖操作系统，浏览器本身支持 SHA256 摘要算法。因而可以在 Windows XP SP2 操作系统上使用这些浏览器访问含有 CFCA EV SSL 证书的网站。

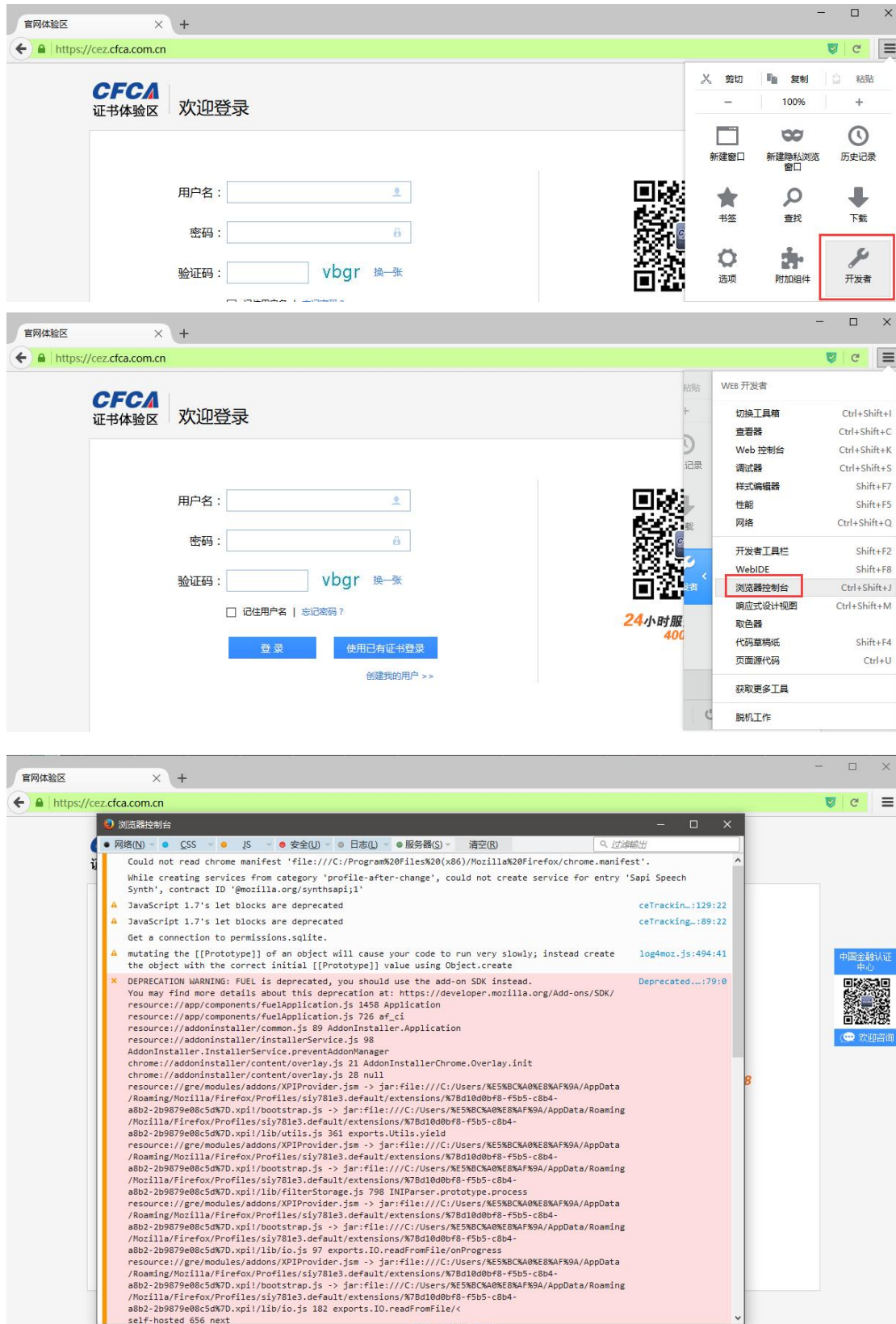
3、通过 HTTPS 访问，页面弹出警告“是否只查看安全传送的网页内容”

当网页包括经加密传送的 HTTPS 内容和未经加密传送的 HTTP 内容时，IE 会弹出警告询问用户是否允许接受未经加密的内容。



可以在“工具”——“Internet 选项”——“安全”——“自定义级别”——“显示混合内容”设置为“启用”，即可不再弹出该提示。

一般来说，当 HTTPS 页面引用外部 HTTP 链接时，会提示此内容不安全。可以通过 Firefox 的 Web 控制台，或者 Chrome 的 JavaScript 控制台查看到具体的报错代码行，并可以参考相关提示修改页面代码。



4、部署 CFCA 站点认证标识

办理 CFCA EV SSL 证书、CFCA OV SSL 证书的网站均可以在其网站页面上嵌入 CFCA 站点认证标识，用户点击该标识，可以跳转到 CFCA 站点认证页面，CFCA 将对网站的相关信息予以认证说明，增强网站的可信度。

部署 CFCA 站点认证标识，需将站点认证图标和以下链接嵌入网站页面上。



<https://evwebverify.cfca.com.cn/WebVerify/webVerifyServlet?domain=网站域名>

注意：1、网站域名必须为完整的域名，如：www.cfca.com.cn

2、该 URL 和图标必须放在对应域名的页面上，如果 URL 中的域名和网页的域名不一致则会认证失败；

3、如果网站既有 https 页面，也有 http 页面，则 https 页面嵌入的 URL 为“https://”，http 页面嵌入的 URL 为“http://”。

5、Chrome、Firefox 提示“SSL 收到了一个弱临时 Diffie-Hellman 密钥”



Chrome、Firefox 等最新版本的浏览器，对客户端浏览器和网站服务器之间的密钥算法有较高要求，不允许客户端浏览器和网站服务器之间使用相对较弱的密钥算法。该问题需要调整 Web 应用服务器的相关配置，限定客户端浏览器和网站服务器之间使用较高强度的密钥。

常用的 Web 应用服务器配置密钥算法的方式如下：

Apache:

在 httpd-ssl.conf 配置文件中增加如下内容:

SSLCipherSuite

ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA:DHE-RSA-AES256-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:AES:CAMELLIA:DES-CBC3-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA

Nginx:

在 conf/nginx.conf 配置文件中增加如下内容:

SSLCipherSuite

ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA:DHE-RSA-AES256-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:AES:CAMELLIA:DES-CBC3-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA

Tomcat:

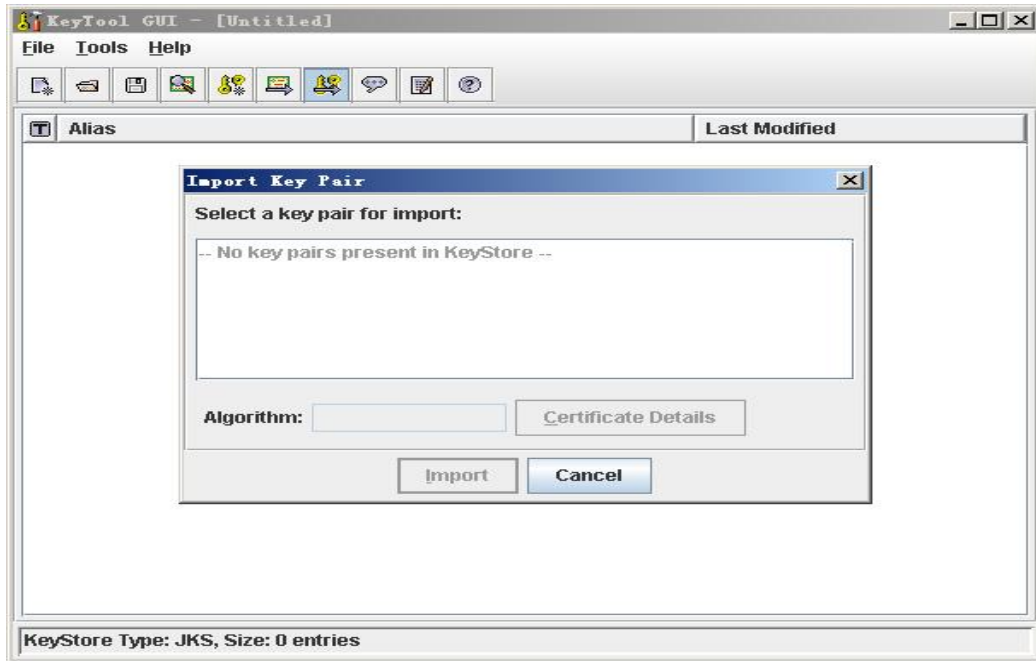
在 conf/server.xml 配置文件中增加如下内容:

```
<Connector  
    ciphers="TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_  
WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_  
ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_DHE_RSA_WITH_AES_128_GCM_  
SHA256,TLS_DHE_DSS_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_1  
28_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_SHA256,TLS_ECDHE_RSA_WITH_AE  
S_128_SHA,TLS_ECDHE_ECDSA_WITH_AES_128_SHA,TLS_ECDHE_RSA_WITH_AES_2  
56_SHA384,TLS_ECDHE_ECDSA_WITH_AES_256_SHA384,TLS_ECDHE_RSA_WITH_AE  
S_256_SHA,TLS_ECDHE_ECDSA_WITH_AES_256_SHA,TLS_DHE_RSA_WITH_AES_128  
_SHA256,TLS_DHE_RSA_WITH_AES_128_SHA,TLS_DHE_DSS_WITH_AES_128_SHA25  
6,TLS_DHE_RSA_WITH_AES_256_SHA256,TLS_DHE_DSS_WITH_AES_256_SHA,TLS_D  
HE_RSA_WITH_AES_256_SHA" />
```

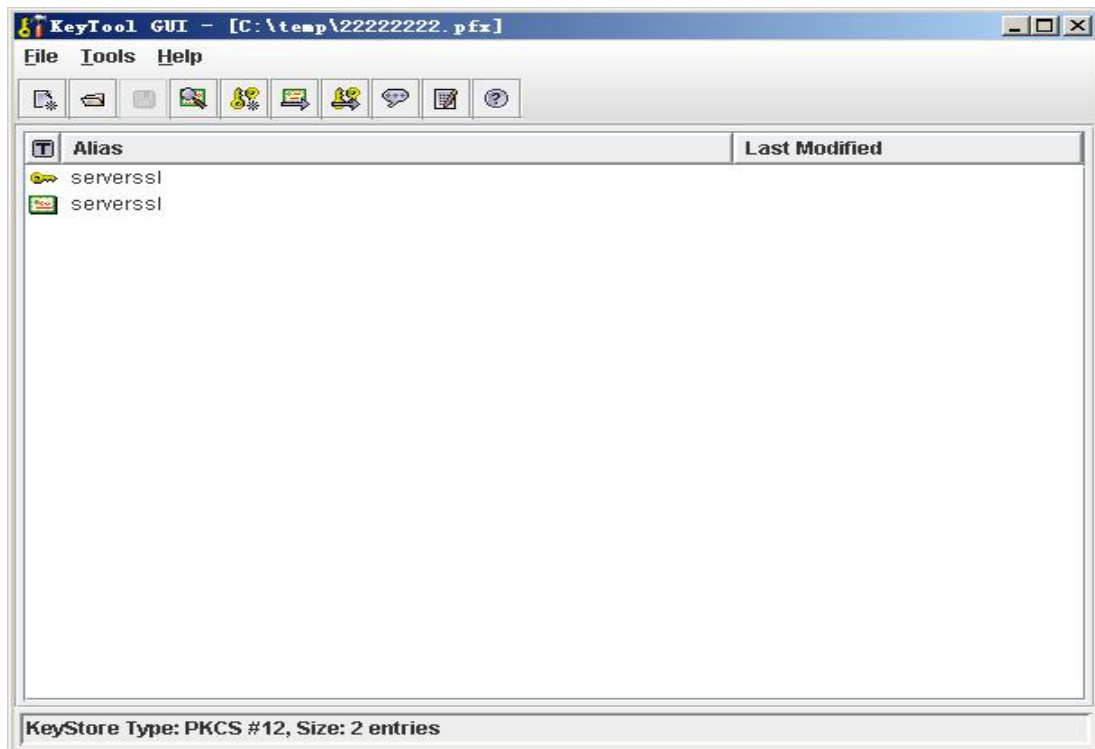
6. 客户反馈下载的服务器公钥证书总是在导入 jks 文件的时候报错 无法安装认证回复

- 确认 JKS 文件是否是当初生成 CSR 文件的证书文件
- 确认证书链是否已经导入: 注意要先后导入根证书、中级证书
- 确认证书链是否与公钥证书的证书路径一致
- 如是不相符的证书文件, 只能重新办理, 即重新产生 CSR 并且补发证书

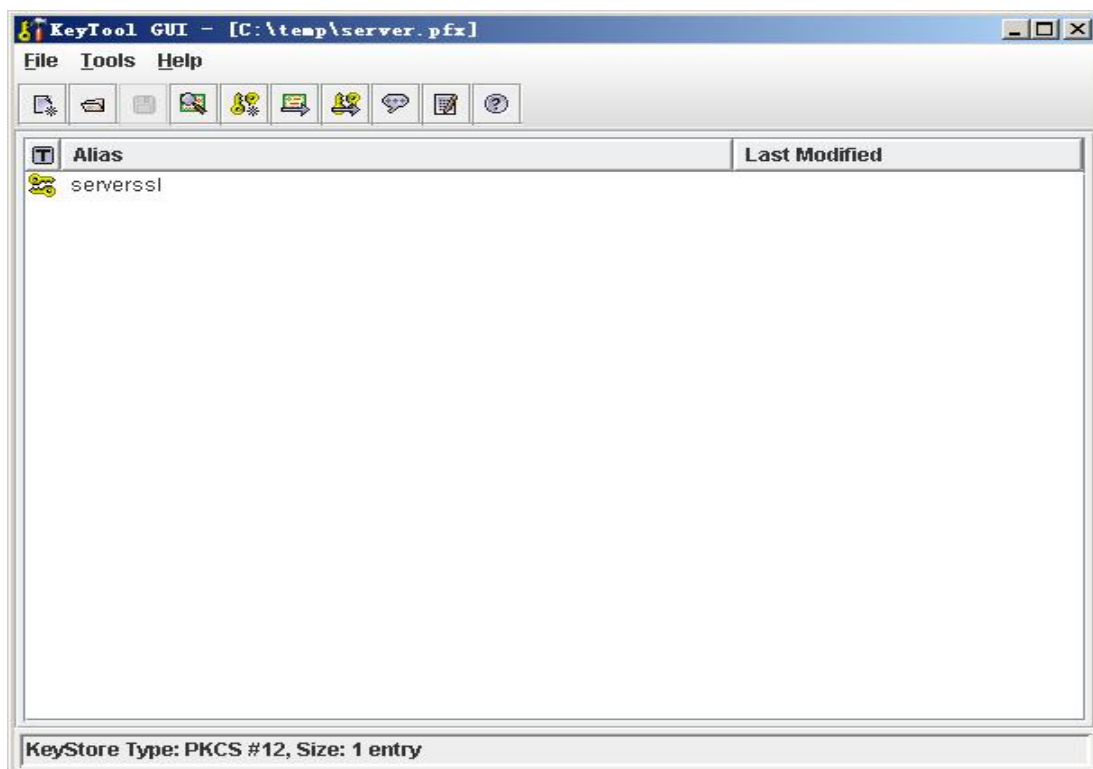
7. pfx 导入 jks 出现如下问题 “No key pairs present in KeyStore ”



问题原因：造成此问题的原因是 pfx 出现问题，在 keytool 工具打开 pfx 文件时，现象如图



如果 pfx 文件正常，那么 keytool 工具打开时，效果如下图：

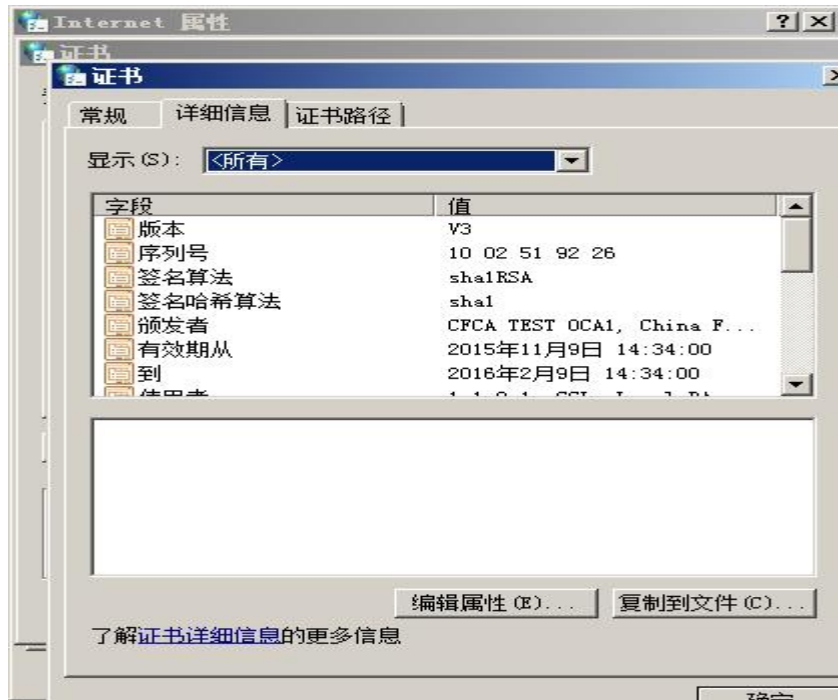


造成 pfx 出现上面另一种情况的原因在于，ie 中看到的友好名称：

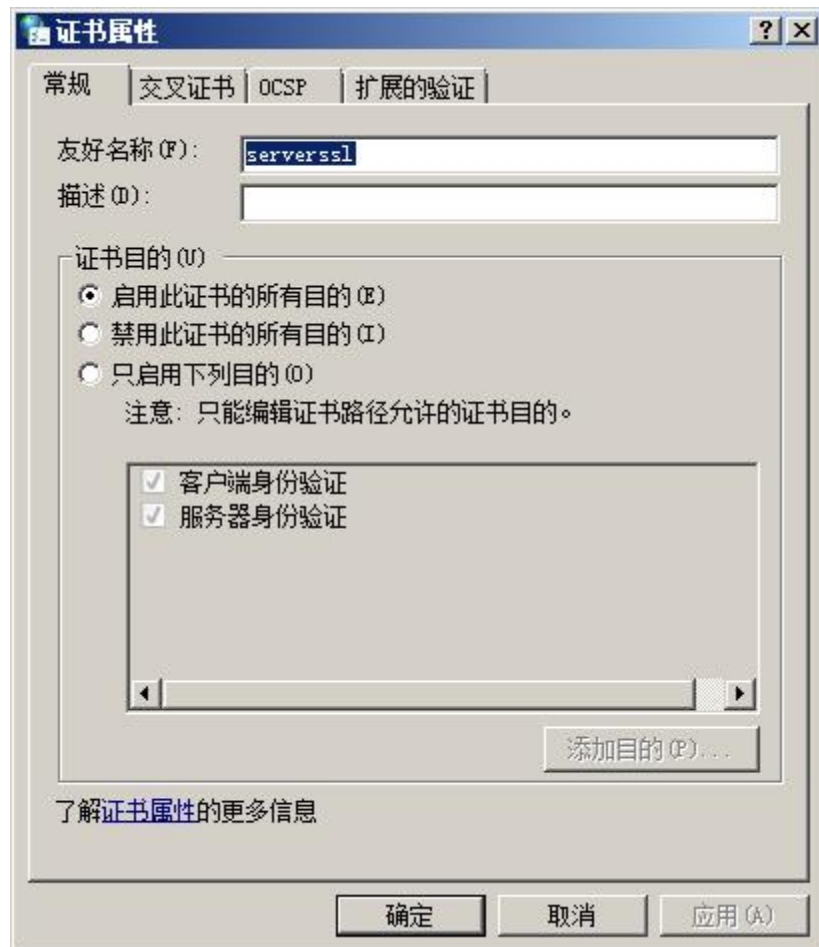


解决方法：删除友好名称

1. 双击证书，查看详细信息：



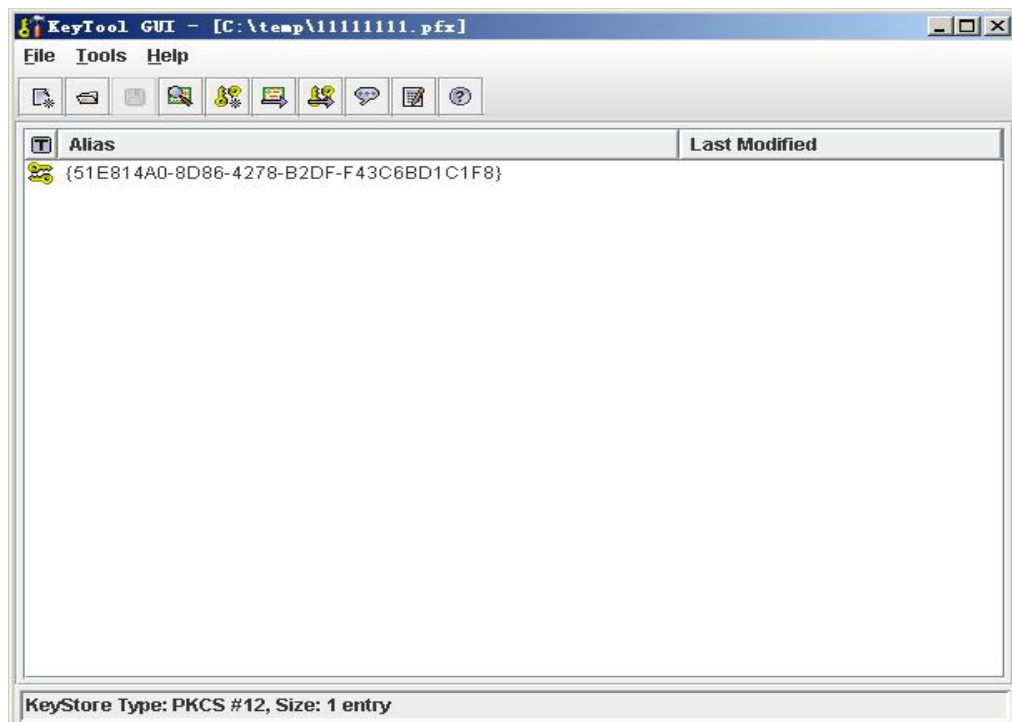
2. 点击“编译属性”



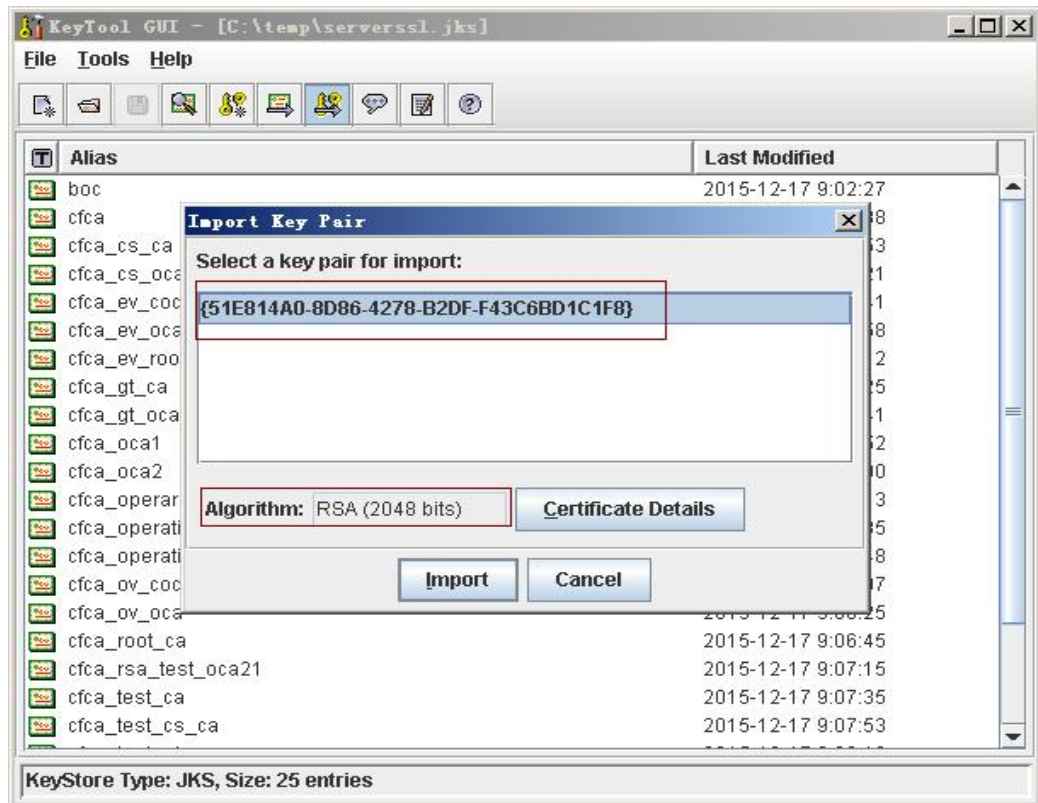
3. 将友好名称删除



4. 之后再导出成 pfx。Jks 查看，pfx 文件正常：



5. 导入 jks 时可以正常选中密钥空间



7. 客户反馈证书配置正确，依旧提出证书不可信。建议客户对服务器进行抓包，确认服务器是否将中级证书和根证书发到客户端。经常发生的情况是客户的网关或者防火墙将中级证书和根证书屏蔽导致。