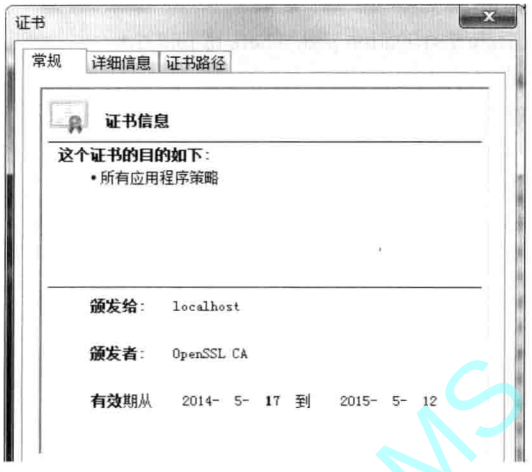

图 22-27　签发后的服务器证书

服务器证书的内容为：

-----BEGIN CERTIFICATE-----
MIICtzCCAZ+gAwIBAgIDAK71MA0GCSqGSIb3DQEBBQUAMCIxCzAJBgNVBAYTAkNO
MRMwEQYDVQQDDApPcGVuU1NMIENBMB4XDTE0MDUxNzAxNDgyOVoXDTE1MDUxMjAx
NDgyOVowITELMAkGA1UEBhMCQ04xEjAQBgNVBAMMCWxvY2FsaG9zdDCBnzANBgkq
hkiG9w0BAQEFAAOBjQAwgYkCgYEArBjZkPl2w4gBpr4/E5hStQ85PckjkaTcnNsf
/Qu7Ifr1WOxN3mahX7PJaD2qyWWproirAcvA//+KSXJYE1D0EPPy1MfZirnebQLv
d48Xk2BA4TGlMVnwI8g1AGNyiOo46f8MKdPClWcL8TjymwtNZMDQgnS23ZYZoZQJ
iCfoYPsCAwEAAaN7MHkwCQYDVR0TBAIwADAsBglghkgBhvhCAQ0EHxYdT3BlblNT
TCBHZW5lcmF0ZWQgQ2VydGlmaWNhdGUwHQYDVR0OBBYEFEh/ovg3V9omtDEz5qc3
mdLOOKTzMB8GA1UdIwQYMBaAFNtEkvYF/pPI0BoMHS63eQ518KGYMA0GCSqGSIb3
DQEBBQUAA4IBAQDF0rR3lyvT9PhRBHtrTkL+FTj0cgBtTyRe5kSs/cWwVWdu1MN6
F4bTmjwR22bjWs4P/QiRWDsEzK2neUXX5LdoJv6BwZsN39dduF40aGt5fk/lfeuJ
1rQDty90i09oiznAcd3GqXAPsvw6bQeEIfoRTG5PgZYLCqOjN4aunumyMJlM0El0
nWKt1g9yDd5HXxPd4FWQttYnfjAMAvzMQNR62ZMD5YWnaxc2Qv6sbSZaJgYvlBrO
Fz1T2l+19WiyGj6l9gZJRINBqBuVpNXJiUT4gbYIz29tpNzYDyRtMukSdtysuMv1
aZIP5oFdvh5qOerbmZdwETPaymFV33+gzwtb
-----END CERTIFICATE-----

CA 证书的内容为：

-----BEGIN CERTIFICATE-----
MIIDFzCCAf+gAwIBAgIJAMB4e3UzgUdVMA0GCSqGSIb3DQEBBQUAMCIxCzAJBgNV
BAYTAkNOMRMwEQYDVQQDDApPcGVuU1NMIENBMB4XDTE0MDUwODEzMzc1NFoXDTI0
MDUxNTEzMzc1NFowIjELMAkGA1UEBhMCQ04xEzARBgNVBAMMCk9wZW5TU0wgQ0Ew
ggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDSeJ+BsTtVNh+o1534vBhq
Oe7nWpS9nkv13OGIXdqddUlJQ2dOOnv40uM98wq/YDiLBhKgG7v0oVraZOUULI9u

pBlwWnz9FFY68WT7JxyxMdWkGb9/SFU83QkcY6HznFdeGbRLwk+7CcH1nwmfDwtO
cg0VeB2tkECUWLJ37C2MrxQrhwfgT6VuG78Wd5URY5PdshPSedAzSHEo74kdgkOx
hSSB4ghQB55YShgI5EanzHTGcnuB8yB58/IllSBug2PNFVMD9o+RKm3UfsoIgl37
8XX6AdwNmyEQae0TIvW+ZFqGQS05qauEq0sLX/CzaGRfN8OXExpLjpRTY5ZAzDzV
AgMBAAGjUDBOMB0GA1UdDgQWBBTbRJL2Bf6TyNAaDB0ut3kOdfChmDAfBgNVHSME
GDAWgBTbRJL2Bf6TyNAaDB0ut3kOdfChmDAMBgNVHRMEBTADAQH/MA0GCSqGSIb3
DQEBBQUAA4IBAQCbbRvZzYQWa1WFpIh3iQi76VoGIst1WQz71xNFxikdBiLCRPkS
+nhYnBXCV7UPesKYdeCDt4TKmYnfGQm4BdIl2glJIUENpndBWqB1lHQ9Xjr7/OfJ
+FLC6e31dmwrKRS3S4HxBWY/7vWBCHOXF2JzAZyqmgkWRG7UXb8ZppFiVXc94E8z
qgmEmQwv3AKqBzqlD/vosQswtYm7fyRn7Xj9sV4plNejKp6pBMlRr6KRnB98XYNG
N6o2nDw9kPS6KvFBkTgM8swmlh/impT0G7wU+GXe0U3toA9EarDCbhH0SJkVnPkf
BiBlexH3w5pBsO+i/PLK/ABQhQpjG+Y+FEgp
-----END CERTIFICATE-----

把服务器证书、服务器私钥、CA 证书文件复制到 Apache 的配置目录 C:/Apache2/conf/下。

22.2.2　配置 SSL 策略

对 SSL 策略的配置，同样在 httpd-ssl.conf 中实现，下面分别予以说明。

1．支持的 SSL 协议

使用 SSLProtocol 配置，可以配置支持 SSLv2、SSLv3、TLSv1、all，如果要去掉对某一协议的支持，则在协议名称前添加减号“-”，如：

```
# 启用 SSLv3 和 TLSv1 但禁用 SSLv2
SSLProtocol all -SSLv2
```

all 表示全部 SSL 协议。因为 SSLv2 协议基本上不使用了，因此可以在配置文件中禁用它。

2．使用算法

通过 SSLCipherSuite 指定在 SSL 握手时支持的算法。配置方式为：

```
SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5
```

SSLCipherSuite 指令的值是一个用冒号分隔的 OpenSSL 加密算法集字符串，用于在 SSL 握手过程中进行加密算法协商时告诉客户端允许使用哪些加密算法。

OpenSSL 加密算法集实际上是由 4 个属性组成的：

① 密钥交换算法：RSA 或 Diffie-Hellman 算法的各种变种。

② 认证算法：RSA，Diffie-Hellman，DSS 或 none。

③ 加密算法：AES，DES，Triple-DES，RC4，RC2，IDEA 或 none。

④ 摘要算法：MD5，SHA 或 SHA1。

可以指定加密算法集中每个属性的算法，也可以使用别名指定一组特定的算法集，如表 22-1 所示。

表 22-1　算法类型及选项

类型	选项	类型	选项
密钥交换算法	kRSA：纯 RSA 密钥交换 kDHr：使用 RSA 密钥的 Diffie-Hellman 密钥交换 kDHd：使用 DSA 密钥的 Diffie-Hellman 密钥交换 kEDH：临时 Diffie-Hellman 交互密钥	摘要算法	MD5：MD5 摘要 SHA1：SHA1 摘要 SHA：SHA 摘要
认证算法	aNULL：不进行认证 aRSA：RSA 认证 aDSS：DSS 认证 aDH：Diffie-Hellman 认证	别名	SSLv2：所有 SSLv2 算法 SSLv3：所有 SSLv3 算法 TLSv1：所有 TLSv1 算法 EXP：所有出口算法 EXPORT40：所有 40-位出口算法 EXPORT56：所有 56-位出口算法 LOW：所有低强度算法（非出口算法，DES） MEDIUM：所有 128-位加密算法 HIGH：所有使用 Triple-DES 或更高强度的算法 RSA：所有使用 RSA 密钥交换的算法 DH：所有使用 Diffie-Hellman 密钥交换的算法 EDH：所有使用临时 Diffie-Hellman 密钥交换的算法 ADH：所有使用匿名 Diffie-Hellman 密钥交换的算法 DSS：所有使用 DSS 认证的算法 NULL：所有不加密的算法
加密算法	eNULL：不加密 AES：AES 加密 DES：DES 加密 3DES：Triple-DES 加密 RC4：RC4 加密 RC2：RC2 加密 IDEA：IDEA 加密		

可以使用下面的语法增删算法以及确定在握手阶段协商的“算法集”优先级顺序。

① [没有标记]：向列表中增加一个算法集。

② +：在列表中的相应位置增加一个算法集。

③ -：从列表中临时删除相应的算法集（之后还可以被再次添加）。

④ !：从列表中永久删除相应的算法集（之后不可以被再次添加）。

该指令默认值为“ALL:!ADH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP”，其含义为：首先永久删除所有使用匿名 Diffie-Hellman 密钥交换的算法，然后添加使用 RC4 和 RSA 的算法，再后顺序添加高、中、低强度的算法，最后再追加所有的 SSLv2 算法和出口算法到列表结尾。

可以使用“openssl ciphers –v”命令查看所有可用的“加密算法集”。

3．客户端验证

使用 SSLVerifyClient 配置客户端验证策略，可以取值为 none、optional、require。none 表示不需要客户端证书，optional 表示客户端可以提供有效证书，require 表示客户端必须提供有效证书。

对于单向 SSL，可以配置为 none 或 optional，对双向 SSL 必须配置为 require。如“SSLVerify Client require”表示要验证客户端证书。

在对客户端证书进行验证的情况下，还需要配置验证级别。在多级 CA 证书情况下，必须进行配置，否则会导致 SSL 握手失败。配置指令用 SSLVerifyDepth。简单说，SSLVerifyDepth 指定了验证证书链的长度。例如，“SSLVerifyDepth 10”表示配置最大验证证书链长度为 10。

22.2.3　访问 Web Server

完成配置后，重新启动 Apache 服务器进程，打开浏览器访问 Web 服务器，显示如图 22-28 所示的内容。

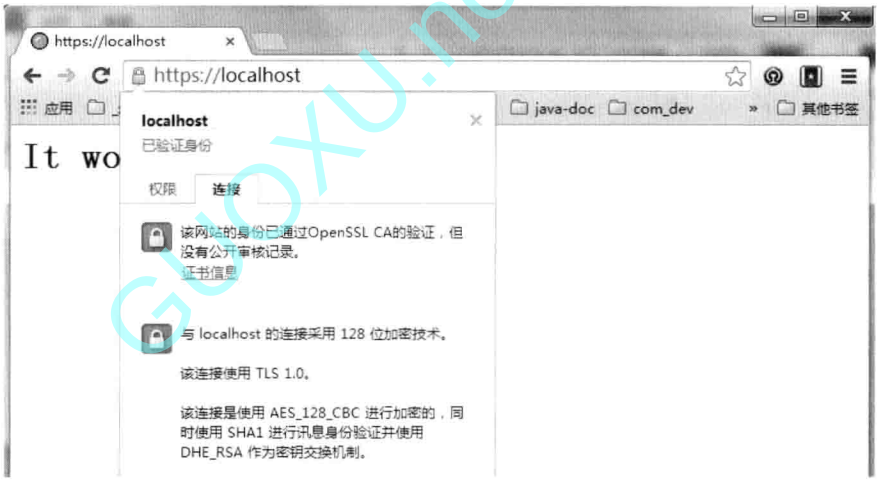

图 22-28　使用 SSL 访问 Apache 网站

从图 22-28 可以看到，客户端与服务器端完成了 SSL 握手，使用了 128 位对称加密算法，且使用了 TLS1.0 协议，CA 证书为 OpenSSL CA。

22.3　Tomcat 服务器证书配置

22.3.1　下载并安装服务器证书

从 http://tomcat.apache.org/下载 7.0 版本的 Tomcat 并进行安装。假定 Tomcat 安装在 D:\tomcat7 目录下。配置 Tomcat 的安全服务包括两部分，一是产生服务器证书，二是配置 SSL 连接器。

1. 产生服务器证书

（1）第一步，生成服务器私钥，产生密钥库文件

使用 Java 提供的 keytool 工具，在安装目录的 conf 子目录下执行命令：

keytool -genkey -alias tcserver -keypass test123 -keyalg RSA -keysize 1024 -validity 365 -keystore tcssl.keystore -storepass pass123 -dname " CN=127.0.0.1， C=CN "

参数含义请参考“12.3 JCA/JCE”的“使用证书”章节。在当前目录下产生了 tcssl.keystore 密钥库文件，包含服务器的私钥。

（2）第二步，生成证书请求文件

执行命令：

keytool -certreq -alias tcserver -sigalg SHA1withRSA -file servercert.req -keystore tcssl.keystore-storepass pass123 -keypass test123

在当前目录下产生了 servercert.req 文件，使用 OpenSSL CA 对此证书请求进行签发。

（3）第三步，签发服务器证书

执行命令：

D:\var\OpenSSL-Win32\bin\openssl.exe ca -in .\servercert.req -out servercert.crt -days 360

产生的证书如图 22-29 所示。

图 22-29 服务器证书信息

（4）第四步，先导入 CA 证书，然后导入签发完成的服务器证书。

执行如下命令，可导入 CA 证书：

keytool -import -alias caroot -trustcacerts -keystore tcssl.keystore -storepass pass123 -file D:\var\OpenSSL-Win32\bin\demoCA\cacert.pem

结果如图 22-30 所示，表示导入证书成功，其中当提示是否信任此证书时，选择 y。

使用如下命令，可导入服务器证书：

keytool -import -alias tcserver -keystore tcssl.keystore -storepass pass123 -keypass test123 -file servercert.crt