COMPUTING SUBJECT:	Root and Server Certificates
TYPE:	Assignment
IDENTIFICATION	CertificateX509 No. 2
COPYRIGHT:	*Michael Claudius*
LEVEL:	Medium
TIME CONSUMPTION:	1-2 hours
EXTENT:	50 lines
OBJECTIVE:	mekecert , pvk2pfx comands
PRECONDITIONS:	Computer Networking Ch. 8.5
COMMANDS:	

IDENTIFICATION: CertificateX509 No.2 /MC

Mission

You are to make a secure connection communication by setting up a server and a client using the secure socket layer (SSL) by sharing the certificate provided by the server. This we shall do in three steps/assignments:

1. CertificateX509, Install Windows SDK and investigate the tools *makecert* and *pvk2pfx*
2. **CreateCertificateX509, Create self-signed X509 Root and Server SSL certificates**
3. SecureSocketsC, Use the certificates and SSLStream for secure socket communication

You have already done the first assignment and this assignment is the Assignment No.2

Purpose

For developing and testing one can create self-signed certificates (e.g. SSL certificates for Root, server and clients) instead of just buying them from Verisign or other providers. This is the purpose of this assignment.

.

Useful links

http://stackoverflow.com/questions/9982865/sslstream-example-how-do-i-get-certificates-that-work

http://stackoverflow.com/questions/14214396/how-to-create-a-certificate-to-use-with-sslstream-authenticateasserver-without-i

http://www.codeproject.com/Articles/25677/Simple-WCF-X-Certificate

http://www.jayway.com/2014/09/03/creating-self-signed-certificates-with-makecert-exe-for-development/

The Mission

To create the certificates, you can either follow the instructions given in the link:

http://www.jayway.com/2014/09/03/creating-self-signed-certificates-with-makecert-exe-for-development/

where they are running a .cmd batch file created in Notepad **OR** just type the commands in the Command Prompt (cmd).
In the following I explain the last mentioned method and for details on what goes on you can also look at the link given above.

1. Root certificate: Creation

First create your own new folder for your certificates e.g. C:\Certificates
Start a dos prompt as administrator: Click: start -> cmd
Position in the folder for certificates by typing the commads like: cd .. and cd certificates
Type (by copy and paste):

```
makecert -r -pe -n "CN=FakeRootCA" -cy authority -sv RootCA.pvk  RootCA.cer
```

On the way you will be prompted for some passwords (use simple ones like *secret*)

Type: dir
And you will see you have created two files: a .cer file (a X.509 certificate with public key) and a .pvk file (with the private key).

Second step is to create a an .pfx file (personal information exchange) holding both the public and private key from respectively the .cer and .pvk file.
Type: (by copy and paste):

```
pvk2pfx -pvk RootCA.pvk  -spc RootCA.cer -pfx RootCA.pfx -po mysecret
```

On the way you will be prompted for the passwords for subject key (*secret*) and private key (*secret*).
Don't forget your keys. If you forget you must create new certificates!

Type: dir
And you will see that now you have three files in the certificate directory.

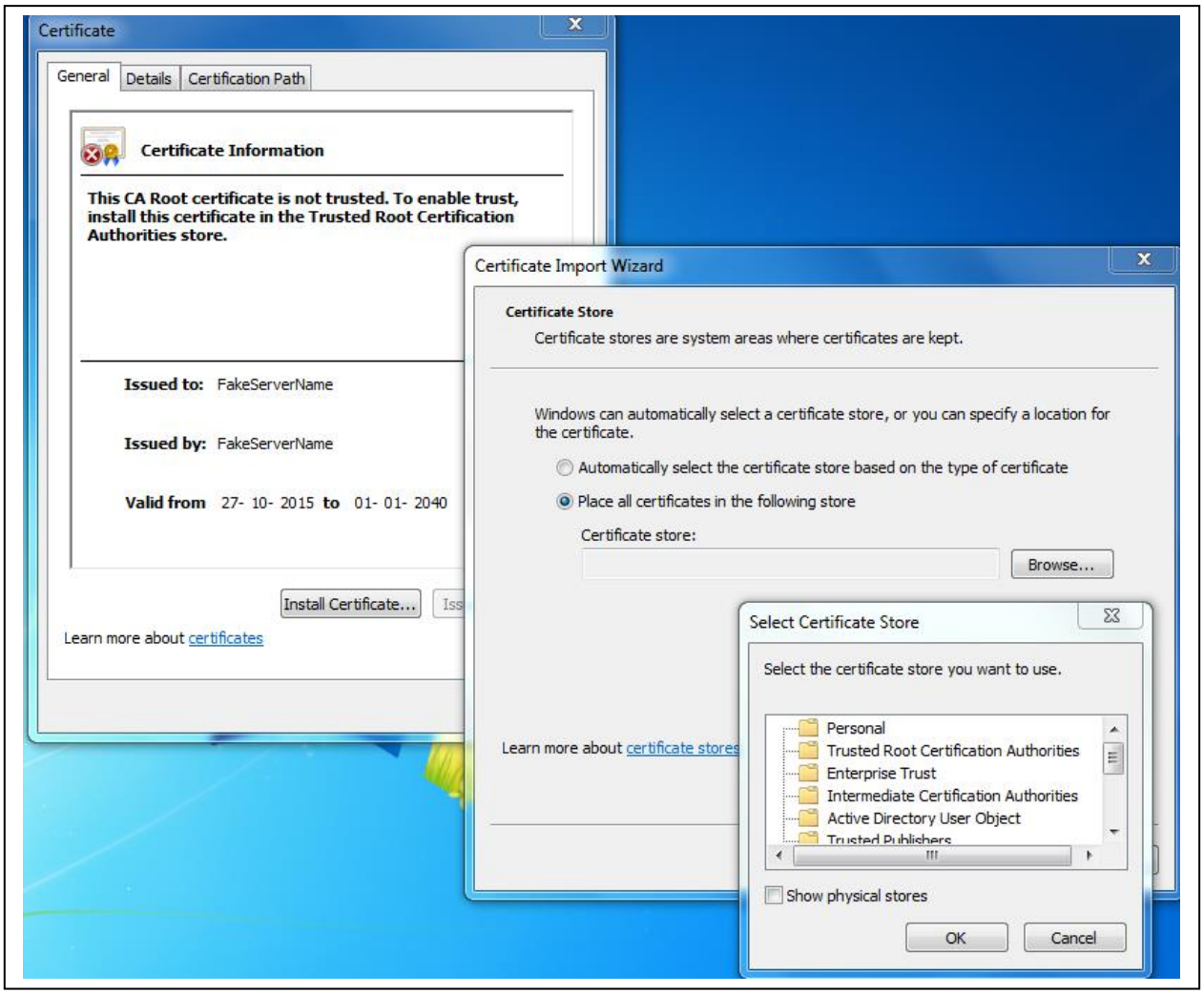
The generated certificate will hold the chosen key pair, the chosen cryptography method (RSA & SHA512) and other standard information.

Comment
There are other possibilities like DSA and MD5, but they are not used here. More general information and information on DSA can be found on:

http://java.sun.com/j2se/1.5.0/docs/tooldocs/windows/keytool.html)
http://en.wikipedia.org/wiki/Digital_Signature_Algorithm (DSA)

2. Root certificate: making it “trusted”

Third step is to install the certificate RootCA.cer in the Trusted Root Certification -> Certificates
Open the RootCA.cer file by double-clicking on it.



Click: Install Certificate
Use: local computer/machine
Choose: Browse
Select: Trusted Root Certification Authorities
Follow the steps (next, ok, finish) and you have now installed the certificate.
Try to open the RootCA.cer file again by double-clicking and notice the difference in information.

3. Server certificate: Creation

Next we create a certificate to handle SSL on the server and this certificate is signed by the RootCA authority.

```
makecert -ic RootCA.cer -iv RootCA.pvk -n "CN= FakeServerName " -pe -sky exchange -sv ServerSSL.pvk ServerSSL.cer
```

Again you will be asked for keys and also the issuer's key, which is the one you choose when creating RootCA.

Type: dir
And you will see you have created two files: a .cer file (a X.509 certificate with public key) and .pvk file (with the private key).

Finally we create a pfx file (personal information exchange) holding both the public and private key from respectively the .cer and .pvk file.
Type: (by copy and paste):

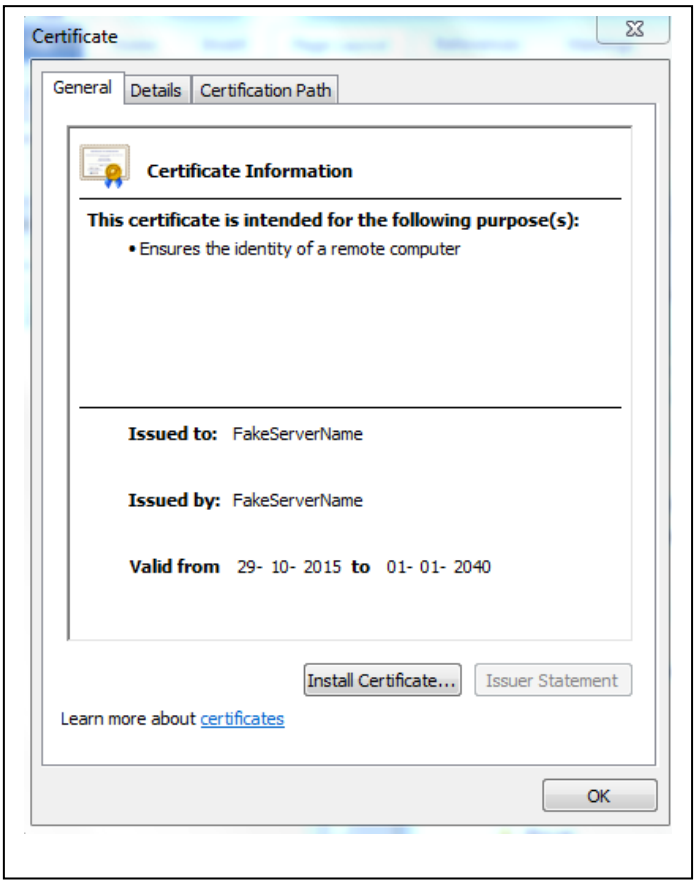
```
pvk2pfx -pvk ServerSSL.pvk  -spc ServerSSL.cer -pfx ServerSSL.pfx -po mysecret
```

On the way you will be prompted for the passwords for subject key and private key (*secret*).
Don't forget your passwords.

Now you have three more files in the certificate directory.

4. Server certificate: making it “trusted”

First open ServerSSL.cer by double-clicking, notice that it has already been automatically installed in Personal -> Certificates. If not install it there.



Secondly, we shall install and import the certificate ServerSSL.pfx into the folder:

Personal -> Certificates

Open the ServerSSL.pfx file by double-clicking. The procedure is very similar to the previous one for RootCA certificate.

Remember that the private key for .pfx file is the password stated by the -po option (*mysecret* in this tutorial).

Maybe you can see the difference by opening the ServerSSL.cer file again by double-clicking.

Now we are ready to use the certificates in C# programs in the next assignment SecureSocketC.