Do-It-Yourself PKI Infrastructure
Producing Free, Private X.509 Certificates For use with PKI-based Solutions

Michael.W.Czapski@gmail.com
May, 2009

Table of Contents

Warning

This document discusses the use of cryptographic software and manipulation of cryptographic objects. Using or discussing cryptography software is illegal in some parts of the world. It is you responsibility to ensure that you comply with any import/export and use laws that apply to you.

Preliminaries

When working with PKI-based security solutions one typically requires one or more X.509 Certificates and related private keys. X.509 Certificates are typically purchased from well known Certification Authorities, such Verisign, for a fair amount of money and are valid for 1 or 2 years. It is not perhaps widely known that one can create a perfectly functional X.509 Certificate and use it in PKI-based solutions by oneself, free of charge and valid for an arbitrary amount of time. While tools are available to both generate key pairs and create X.509 Certificates, the how of it is somewhat obscure. This document discusses the use of the OpenSSL software in creation of private PKI objects such as Key Pairs and X.509 Certificates and PKCS#12 Keystores. It discusses the use of Windows-based scripts, developed by the author, that make the process painless and quick.

The discussion below, and scripts provided in the package, applies to Windows. Adjust as necessary for Unix.

Download the OpenSSL distribution by following the links from http://www.openssl.org/ and locating the binary distribution for Windows: http://www.openssl.org/related/binaries.html. Download just the Win32 OpenSSL v0.9.8k Light. All we need form the package are openssl.exe, libeay32.dll and ssleay32.dll.

Create a directory hierarchy, rooted as some arbitrarily chosen directory, to contain cryptographic objects and binaries. Change working directory to this arbitrarily chosen directory, for example `C:\JCAPS6U1Projects\SecMail`. In this directory execute the following commands:

```
mkdir -p pki\doc\mcz pki\bin pki\ca\DemoCA
```

This will produce a hierarchy like that shown below.

```
C:\JCAPS6U1Projects\SecMail>tree pki
Folder PATH listing for volume C_Drive
Volume serial number is D8D0-733B
C:\JCAPS6U1PROJECTS\SECMAIL\PKI
├───bin
├───ca
│   └───DemoCA
└───doc
    └───mcz
```

Install the package to a subdirectory of the working directory, for example C:\JCAPS6U1Projects\SecMail\OpenSSL. Ignore warnings about missing redistributable, if any. Copy openssl.exe, libeay32.dll and ssleay32.dll from C:\JCAPS6U1Projects\SecMail\OpenSSL\bin to C:\JCAPS6U1Projects\SecMail\pki\bin.
Uninstall the package, unless you need to/would like to keep it around.

Verify that the binary executes:

```
cd C:\JCAPS6U1Projects\SecMail\pki>
bin\openssl –help
```

Unzip the following batch scripts, available at http://mediacast.sun.com/users/Michael.Czapski-Sun/media/PKI_Scripts_Windows.zip/details, to the pki\doc\mcz directory, and modify as you might require – the literals in configuration files, for example, like organisaiton names, email addresses and such:

```
└───doc
    └───mcz
            CA_01_create_new_ca.notes.txt.cmd
            CA_02_isse_and_revoke_end_use_cers.notes.cmd
            CA_03_revoke_end_use_cers.notes.cmd
            EU_01_request_a_certificate.notes.txt.cmd
            EU_02_finish_certificate_acquisition.notes.txt.cmd
```

The following assumptions are embedded in the scripts:

1. directory names, generated file names, file paths, passphrases and export passwords are hardcoded and/or derived from the name of the party whose cryptographic objects are being manipulated
2. CA's name is hardcoded in the scripts that start with CA0x– you can change them if you need to. Changes are localised to three lines at the top of each script and are clearly marked.
    a. CA's name is hardcoded as democa
    b. CA's passphrase is hardcoded as democa
    c. CA's 'export' password is hardcoded as democaexport
    d. CA's contact email address is hardcoded as certification@authority.com
3. End user (EU) name and contact email address are provided on the command line for the commands starting with EU0x.
    a. EU's passphrase is hardcoded as the EU's name concatenated with EU's name i.e. EU's name doubled. If the name of the EU is long, say more that about 10 characters, you may get failures from some commands. If this is the case, either shorten the name, modify the script to not double the name for the passphrase or hardcode the passphrase.
    Example: If EU's name is TParty the EU's passphrase is TPartyTParty
    b. EU's export password is the literal 'export' appended to the EU's name
    Example: If EU's name is TParty then the export password is TPartyexport
4. Scripts create all subdirectories they need.
5. Names of the objects/files the scripts create are derived from the name of the subject (CA or EU). The rules are hardcoded. Change at your own risk
6. There are other areas where you might need to change scripts. For example, it is assumed that everyone on the planet lives in Sydney, NSW, Australia ☺ (locality name, stateOrProvince name and country name defaults). If this is not the case for you, you may need to change the defaults – those changes are merely cosmetic and do not affect the scripts or the usability of the cryptographic objects produced.

The directory from which you run all scripts is the directory that contains the 'bin' subdirectory. This is the directory to which you would have unzipped the archive.

Note: The certification requests and certificates created by the scripts ***_are not suitable_*** for use with web servers. Web servers ***_expect_ and _require_*** the 'common name' to be the fully-qualified domain name of the web server or its IP address. You can still create appropriate CSRs and Certificates with the commands embedded in the scripts but you will need to change the scripts to ask for and set the 'common name' correctly or you will need to issue the commands by hand with appropriate modifications.

If you need to do all the PKI stuff yourself then you need to play the part of two parties – the Certification Authority (CA) and the user (EU). You use the scripts in the following order:

1. (CA) create a CA infrastructure (once only to create the CA and bootstrap it for further use)
2. For each party that needs a certificate:
    a. (EU) Create a end-use certificate signing request and copy it to the CA's end-use requests directory
    b. (CA) create and sign end-user certificate (the script will automatically deposit the certificate, in different formats, in the end-user's directory
    c. (EU) Create a KeyStore for Java-based tools and products

Note that you only need to create the CA infrastructure (step 1) once.
Note that you will need to go through the steps 2a, 2b and 2c once for each end-use certificate you need.
Note that each 'party' gets one certificate and that the certificate is good for signing as well as encryption. If you need to have separate certificates for different purposes you can create two 'parties' with slightly different names and rename certificate files afterwards as you see fit. If the certificate options must explicitly disallow signing or encryption you will need to fiddle with configuration fields and issue commands by hand – you will be pretty much on your own there. If you know how to do that kind of stuff you don't need my scripts anyway ☺

(1) Create a CA

Issue the following command:

doc\mcz\CA_01_create_new_ca.notes.txt.cmd

That is it as far as CA creation is concerned – a lot of effort saved ☺

What you should see is reproduces as an appendix A.

Once done, the following objects will be created:

```
C:\JCAPS6U1Projects\SecMail\pki>tree /f ca
Folder PATH listing for volume C_Drive
Volume serial number is D8D0-733B
C:\JCAPS6U1PROJECTS\SECMAIL\PKI\CA
└───DemoCA
    │   catest.conf
    │   catest.pem.crt
    │   catest.pem.private.key
    │   catest.pem2.crt
    │   democa.conf
    │   democa.der.crl
    │   democa.der.crt
    │   democa.pem.crl
    │   democa.pem.crt
    │   democa.pem2.crt
    │   democa.pkcs12.keystore.p12
    │   democa.srl
    │   democa.srl.old
```

```
│   democaIssuesIndex.txt
│   democaIssuesIndex.txt.attr
│   democaIssuesIndex.txt.attr.old
│   democaIssuesIndex.txt.old
│   democaReq.conf
│
├───eu_requests
│       catest.pem.csr
│
├───new_certificates
│       00.pem
│       catest.pem2.crt
│
└───private
        democa.pem.private.key
```

(2a) Create End Use Certificate Signing Request

This task involves creating a key pair (private key and public key pair) and generating a certificate signing request.

On the command line execute the following command, replacing the two placeholders with appropriate values:

doc\mcz\EU_01_request_a_certificate.notes.txt.cmd <end_user_party_name> <contact_email_address>

You will see commands and output echoed to the console window. At the end of the process, which takes a couple of seconds, you will have a subdirectory <end_user_party_name> that contains the following:

- <end_user_party_name>.conf
- <end_user_party_name>.pem.csr
- <end_user_party_name>.pem.private.key

For example, once the command shown below executes:

```
C:\JCAPS6U1Projects\SecMail\pki>doc\mcz\EU_01_request_a_certificate.n
otes.txt.cmd msender msender@some.company.com
```

you will see the following

```
C:\JCAPS6U1Projects\SecMail\pki>tree /f msender
Folder PATH listing for volume C_Drive
Volume serial number is D8D0-733B
C:\JCAPS6U1PROJECTS\SECMAIL\PKI\MSENDER
    msender.conf
    msender.pem.csr
    msender.pem.private.key

No subfolders exist
```

<end_user_party_name>.pem.csr, the Certificate Signing Request, is automatically copied to the CAs `ca\DemoCA\eu_requests` directory where DemoCA is the name of the CA.

Appendix B shows the output generated by this step in the console window.

(2b) Create and Sign End-Use Certificate

Tell CA to create a certificate for a specific party, where <end_user_party_name> is the exact name you provided in the previous step.

doc\mcz\CA_02_isse_and_revoke_end_use_cers.notes.cmd <end_user_party_name>

Note that the CA will write certificates directly to the party's directory.

For example the following command will create certificate objects for party msender:

```
doc\mcz\CA_02_isse_and_revoke_end_use_cers.notes.cmd msender
```

Sample output is reproduced in Appendix C.

(2c) Create a KeyStore

The certificate that the CA issues is good for most things – IE will take it, IIS will take it, Apache will take it, Outlook will take it with some convincing, …, but Java-based tools and products ***_will not_*** take it in many cases. To round off the process one would now create a KeyStore that would be acceptable to Java-based tools and products.

doc\mcz\EU_02_finish_certificate_acquisition.notes.txt.cmd < end_user_party_name>

Sample output is provided in appendix D.

Command:

doc\mcz\EU_02_finish_certificate_acquisition.notes.txt.cmd msender

produces the following objects:

```
C:\JCAPS6U1Projects\SecMail\pki>tree /f msender
Folder PATH listing for volume C_Drive
Volume serial number is D8D0-733B
C:\JCAPS6U1PROJECTS\SECMAIL\PKI\MSENDER
    msender.conf
    msender.der.cer
    msender.der.crt
    msender.pem.crt
    msender.pem.csr
    msender.pem.private.key
    msender.pem2.crt
    msender.pkcs12.keystore.p12
```

No subfolders exist

Note the following:

Object	passphrase	use
msender.conf	Not applicable	Used when generating Certificate Signing Request – not used thereafter
msender.der.cer	Not applicable	End-use X.509 Certificate in DER format (most tools don't care how the certificate is encoded and can deal equally well with PEM and DER forms) – used for encryption and digital signature verification
msender.der.crt	Not applicable	End-use X.509 Certificate in DER format (most tools don't care how the certificate is encoded and can deal equally well with PEM and DER forms) – same as above, just a different file extension – used for encryption and digital signature verification
msender.pem.crt	Not applicable	End-use X.509 Certificate PEM encoded (most tools don't care how the certificate is encoded and can deal equally well with PEM and DER forms) – used for encryption and digital signature verification
msender.pem.csr	Not applicable	Certificate Singing Request, PEM encoded – only used to request a certificate from the Certification Authority
msender.pem.private.key	msendermsender	msender's private key, PEM encoded – used for digital signing and decryption
msender.pem2.crt	Not applicable	End-use X.509 Certificate, PEM encoded, with 'human readable' information (most tools don't care how the certificate is encoded and can deal equally well with PEM and DER forms) – used for encryption and digital signature verification
msender.pkcs12.keystore.p12	msendermsender	PKCS#12 Keystore containing msender's private key and certificate.

Use discussion

Let's discuss what is meant by security in the context of PKI-based solutions.

One can “secure” communication by requiring the use of Digital Signing and Encryption, individually or in combination. By requiring and using different combinations of security tokens one obtains varying degrees of “security”.

Digital Signatures are used to ensure integrity of messages on the wire, that is, facilitate detection of message alteration in transit. They are also used to convey authenticity of messages. The entire message, or selected parts of the message when using XML Digital Signatures over XML, can be digitally signed. The reason one would digitally sign parts of the message rather then the entire message is typically cost. If only certain parts of XML messages must be protected from tampering and must be guaranteed to be authentic then signing selected parts of the message will minimise the high resource consumption typically associated with cryptographic operations.

Conveyance of authenticity relies on the properties and use patterns of key pairs in public key cryptography. Both keys of the pair are generated at the same time. The two keys are related in such a way that one cannot be derived from the other and that plaintext encrypted with one can only be decrypted with the other. One of the keys, the private key, is kept confidential by the party that owns the key pair. The other key, the public key, embedded in a “certificate” which guarantees its authenticity and integrity, is distributed to any party with whom secure communication will be undertaken. Public Key Infrastructure (PKI) is the means of guaranteeing public key authenticity and integrity through issuance and revocation of “certificates’, and possibly distribution of public keys (certificates). There is a great deal more to all this but for the purpose of this discussion it is enough to say that if the owner of the private key encrypts some plaintext and sends it to the recipient, the recipient will be able to decrypt it only with sender’s public key, to which he/she has ready access. Because of the properties of the key pair the recipient knows that only the “other” key of the pair could have been used to encrypt the plaintext that he/she just decrypted. Because the “other” key, the private key, is supposed to be kept secret by its owner the recipient assumes that only the owner of the private key could possibly have encrypted the plaintext. This guarantees message authenticity.

Encryption can be used to ensure message integrity and protect confidentiality of information on the wire. Either the entire message or selected parts of the message, if the message is an XML message and XML Encryption is used, can be encrypted for the same reasons that an entire message or selected parts of a message would be digitally signed. If the encrypted parts of the message are tampered with, decryption will fail and the recipient will conclude that the message was tampered with. The confidentiality of the message is guaranteed in a way similar to authenticity guarantee when using digital signatures. By encrypting the message with a public key of the recipient, the sender of the message ensures that only the recipient, the holder of the private key of the key pair, can possibly decrypt the message. This allows anyone to send a confidential message to the owner of the private key regardless of where there was a prior communication between them.

Please note the following usage pattern to understand which objects are used for what and which objects are used for whom.

The following process is followed to obtain various cryptographic objects:

1. Create a key pair consisting of a “private key” and “public key” (Script EU_01_.... does that)
2. Create a Certificate Signing Request (CSR), which embeds the “publick key”, and send it to the Certification Authority for certification (Script EU_01_... does that)
3. Use the Certificate Signing Request to issue a X.509 Certificate for the requesting party and deliver it to the requesting party (Script CA_02_... does that)
4. Produce various version of the X.509 Certificate for different uses – PEM encoded, DER, PKCS#12 Keystore (Script EU_02_... does this)

Please note that if you are the party requesting and receiving a certificate from the CA then the following rules apply:

1. You never give away your private key to anyone under any circumstances – if you do the private key ceases to be private and becomes useless for the purpose of secure communication
2. You never use your own X.509 Certificate for anything that you do – you provide it to everybody else who needs to enter into secure communication with you – your X.509 Certificate is what everybody else uses
3. If your certificate is issued by a private CA (such as is the case in this discussion), that is not Verisign, RSA Security or similar, you must also provide the CA certificate to the parties to whom you are providing your own certificate – the CA X.509 Certificate of the democa CA for this discussion is located in pki\ca\democa folder.

The following variants of CA certificates are available for distribution:

```
└─────democa
      democa.der.cer
      democa.der.crt
      democa.pem.crt
      democa.pem2.crt
```

It typically does not matter which variant you provide to others.

Appendix A: CA Creation output

```
C:\JCAPS6U1Projects\SecMail\pki>doc\mcz\CA_01_create_new_ca.notes.txt.cmd

C:\JCAPS6U1Projects\SecMail\pki>set CAName=democa

C:\JCAPS6U1Projects\SecMail\pki>set BasePassword=democa

C:\JCAPS6U1Projects\SecMail\pki>set contacteMailAddress=certification@authority.com

C:\JCAPS6U1Projects\SecMail\pki>set CARoot=ca\\democa

C:\JCAPS6U1Projects\SecMail\pki>set CAKey=ca\\democa\\private\\democa.pem.private.key

C:\JCAPS6U1Projects\SecMail\pki>set CAPEMCert=ca\\democa\\democa.pem.crt

C:\JCAPS6U1Projects\SecMail\pki>set CAPEM2Cert=ca\\democa\\democa.pem2.crt

C:\JCAPS6U1Projects\SecMail\pki>set CADERCert=ca\\democa\\democa.der.crt

C:\JCAPS6U1Projects\SecMail\pki>set CADER2Cert=ca\\democa\\democa.der.cer

C:\JCAPS6U1Projects\SecMail\pki>set
CAPKCS12Store=ca\\democa\\democa.pkcs12.keystore.p12

C:\JCAPS6U1Projects\SecMail\pki>set CAReqConfig=ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>set CAConfig=ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>set CAPass=democa

C:\JCAPS6U1Projects\SecMail\pki>set CAExportPass=democaexport

C:\JCAPS6U1Projects\SecMail\pki>set CANewCertsDir=ca\\democa\\new_certificates

C:\JCAPS6U1Projects\SecMail\pki>set CACRLPEMFile=ca\\democa\\democa.pem.crl

C:\JCAPS6U1Projects\SecMail\pki>set CACRLDERFile=ca\\democa\\democa.der.crl

C:\JCAPS6U1Projects\SecMail\pki>set CAEURequests=ca\\democa\\eu_requests

C:\JCAPS6U1Projects\SecMail\pki>set OPENSSL=bin\openssl

C:\JCAPS6U1Projects\SecMail\pki>mkdir ca\\democa\\new_certificates

C:\JCAPS6U1Projects\SecMail\pki>mkdir ca\\democa\private

C:\JCAPS6U1Projects\SecMail\pki>mkdir ca\\democa\\eu_requests

C:\JCAPS6U1Projects\SecMail\pki>echo Y  | copy NUL ca\\democa\democaIssuesIndex.txt
        1 file(s) copied.

C:\JCAPS6U1Projects\SecMail\pki>echo 00  1>ca\\democa\democa.srl

C:\JCAPS6U1Projects\SecMail\pki>echo [ req ]  1>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo default_bits        = 2048
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo default_keyfile     =
ca\\democa\\private\\democa.pem.private.key  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo distinguished_name  = req_distinguished_name
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo attributes          = req_attributes
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #x509_extensions     = v3_ca             # self
signed cert extensions  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo encrypt_rsa_key     = yes
1>>ca\\democa\\democaReq.conf
```

```
C:\JCAPS6U1Projects\SecMail\pki>echo default_md          = sha1
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo [ req_distinguished_name ]
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo countryName         = Country Name
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo countryName_min     = 2
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo countryName_max     = 2
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo countryName_default = AU
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo countryName_value   =
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo stateOrProvinceName           = State or Province
Name  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo stateOrProvinceName_default   = NSW
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo stateOrProvinceName_value     =
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo localityName        = Locality Name
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo localityName_default= Sydney
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo localityName_value  =
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo organizationName             = Organization Name
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo organizationName_default      = democa
Certification Authority  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo organizationName_value        =
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo organizationalUnitName        = Organizational
Unit Name  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo organizationalUnitName_default= democa Security
Division  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo organizationalUnitName_value  =
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo commonName          = Common Name
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo commonName_max      = 64
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo commonName_default  = democa
1>>ca\\democa\\democaReq.conf
```

```
C:\JCAPS6U1Projects\SecMail\pki>echo commonName_value    =
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo emailAddress        = Email Address
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo emailAddress_max    = 40
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo emailAddress_value  = certification@authority.com
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo [ req_attributes ]  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo challengePassword             = A Challenge
Password  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo challengePassword_min         = 8
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo challengePassword_max         = 20
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo challengePassword_value       =
1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl req -new -config
ca\\democa\\democaReq.conf -set_serial 5 -keyform PEM -passout pass:democa -x509 -days
3000 -keyout ca\\democa\\private
\\democa.pem.private.key -outform PEM -out ca\\democa\\democa.pem.crt
Loading 'screen' into random state - done
Generating a 2048 bit RSA private key
.....................+++
............+++
writing new private key to 'ca\\democa\\private\\democa.pem.private.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.
-----
Country Name [AU]:
State or Province Name [NSW]:
Locality Name [Sydney]:
Organization Name [democa Certification Authority]:
Organizational Unit Name [democa Security Division]:
Common Name [democa]:
Email Address []:certification@authority.com

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl pkcs12 -export -passin pass:democa -
password pass:democaexport -in ca\\democa\\democa.pem.crt -inkey
ca\\democa\\private\\democa.pem.pri
vate.key -out ca\\democa\\democa.pkcs12.keystore.p12 -name democa -des3 -nomaciter -
noiter
Loading 'screen' into random state - done

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl x509 -in ca\\democa\\democa.pem.crt -
inform PEM -text -noout
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 5 (0x5)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=AU, ST=NSW, L=Sydney, O=democa Certification Authority, OU=democa
Security Division, CN=democa/emailAddress=certification@authority.com
        Validity
            Not Before: May  3 23:16:05 2009 GMT
            Not After : Jul 20 23:16:05 2017 GMT
```

```
        Subject: C=AU, ST=NSW, L=Sydney, O=democa Certification Authority, OU=democa
Security Division, CN=democa/emailAddress=certification@authority.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
                Modulus (2048 bit):
                    00:da:03:1a:0f:6c:6e:c6:31:ab:08:79:1c:b5:86:
                    94:fc:e0:8e:00:89:6f:d4:76:c2:65:c3:ef:51:17:
                    68:bb:e8:ea:89:bd:67:6d:16:ac:5e:b7:65:2f:a6:
                    4e:8d:ff:67:56:50:a2:11:ea:07:ae:51:b6:51:1a:
                    1a:64:d7:2e:82:78:2c:cd:a5:7a:42:2d:c8:db:6d:
                    c4:3f:82:8d:b0:1f:c4:55:37:e2:70:6b:2b:92:6d:
                    71:e8:c1:94:5b:89:41:c3:fc:8c:e7:0d:33:32:6c:
                    54:fe:a5:10:62:42:ac:99:99:16:18:d1:23:0a:69:
                    e9:d6:ab:3e:ba:d3:42:a8:11:66:2e:03:5d:64:a7:
                    f6:cc:1d:da:18:d1:77:cb:10:9b:1c:01:4b:aa:77:
                    c7:cc:2a:3d:29:4d:19:d7:27:9b:b8:e3:9d:26:5a:
                    20:2f:de:13:53:de:76:f2:47:97:e6:af:41:5f:49:
                    db:dd:c3:ee:9f:24:7b:81:6b:07:c2:1f:19:ec:5f:
                    b3:e2:cb:59:fc:d0:bd:8e:8c:6d:f9:08:b5:ad:cd:
                    9c:d0:ae:cb:0f:4c:e2:a4:5c:7f:05:cb:38:24:f3:
                    4f:78:72:0d:e8:43:3e:68:e4:33:af:fd:43:3f:7b:
                    df:5b:69:5a:f2:ec:a5:28:b2:e1:44:5b:45:c1:b5:
                    ba:6d
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha1WithRSAEncryption
        d5:6d:bf:f5:33:a6:f9:6b:56:0c:b2:30:ed:3d:e7:06:cb:52:
        26:1b:a8:46:ac:b3:93:91:0d:b0:cb:90:85:7c:a2:a1:8e:a2:
        9a:61:89:f6:1d:cd:2b:64:77:6e:af:2c:8f:f9:59:ff:92:b3:
        13:ae:7a:7f:dd:31:19:13:2e:26:c0:77:e9:43:6b:04:d6:1e:
        94:34:74:67:bb:a1:07:0f:d1:f1:c3:d7:1c:cc:20:b7:72:0f:
        e4:80:2b:08:19:d5:ae:67:6b:c3:05:87:b9:3e:e3:76:fd:d3:
        af:9a:f2:0d:66:8d:30:3e:b9:94:7b:c7:75:6d:5b:70:ad:99:
        5b:40:1a:29:2c:cf:e1:91:a3:d9:47:2d:c8:82:8a:29:83:ce:
        4f:b1:7f:78:b5:d5:5b:d4:d9:99:63:a6:33:ce:6a:3f:98:ce:
        fa:0e:09:43:14:87:7d:f0:c2:67:0e:6d:b7:98:0e:a5:68:6b:
        95:62:f9:41:a1:e1:e6:6e:97:31:ad:d5:d1:db:7e:19:91:f3:
        6b:5e:68:40:d2:97:ae:35:6d:79:66:d7:38:c0:81:c5:47:ac:
        85:6e:a2:49:19:b0:fb:70:dd:1a:8e:66:1c:78:98:e0:3f:f7:
        d2:ea:39:fd:07:4e:14:b7:f8:88:f0:b1:f3:e6:a2:a1:c3:8a:
        76:6f:30:29

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl pkcs12 -info -in
ca\\democa\\democa.pkcs12.keystore.p12 -passin pass:democaexport -passout
pass:democaexport
MAC Iteration 1
MAC verified OK
PKCS7 Encrypted data: pbeWithSHA1And40BitRC2-CBC, Iteration 1
Certificate bag
Bag Attributes
    friendlyName: democa
    localKeyID: C5 EB CC 64 44 4D E4 5C C5 ED 20 05 DA D9 9C 9B 9E 03 F2 1B
subject=/C=AU/ST=NSW/L=Sydney/O=democa Certification Authority/OU=democa Security
Division/CN=democa/emailAddress=certification@authority.com
issuer=/C=AU/ST=NSW/L=Sydney/O=democa Certification Authority/OU=democa Security
Division/CN=democa/emailAddress=certification@authority.com
-----BEGIN CERTIFICATE-----
MIID4DCCAsgCAQUwDQYJKoZIhvcNAQEFBQAwgbUxCzAJBgNVBAYTAkFVMQwwCgYD
VQQIEwNOU1cxDzANBgNVBAcTBlN5ZG5leTEnMCUGA1UEChMeZGVtb2NhIENlcnRp
ZmljYXRpb24gQXV0aG9yaXR5MSEwHwYDVQQLExhkZW1vY2EgU2VjdXJpdHkgRGl2
aXNpb24xDzANBgNVBAMTBmRlbW9jYTEqMCgGCSqGSIb3DQEJARYbY2VydGlmaWNh
dGlvbkBhdXRob3JpdHkuY29tMB4XDTA5MDUwMzIzMTYwNVoXDTE3MDcyMDIzMTYw
NVowgbUxCzAJBgNVBAYTAkFVMQwwCgYDVQQIEwNOU1cxDzANBgNVBAcTBlN5ZG5l
eTEnMCUGA1UEChMeZGVtb2NhIENlcnRpZmljYXRpb24gQXV0aG9yaXR5MSEwHwYD
VQQLExhkZW1vY2EgU2VjdXJpdHkgRGl2aXNpb24xDzANBgNVBAMTBmRlbW9jYTEq
MCgGCSqGSIb3DQEJARYbY2VydGlmaWNhdGlvbkBhdXRob3JpdHkuY29tMIIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2gMaD2xuxjGrCHkctYaU/OCOAIlv
1HbCZcPvURdou+jqib1nbRasXrdlL6ZOjf9nVlCiEeoHrlG2URoaZNcugngszaV6
Qi3I223EP4KNsB/EVTficGsrkm1x6MGUW4lBw/yM5w0zMmxU/qUQYkKsmZkWGNEj
Cmnp1qs+utNCqBFmLgNdZKf2zB3aGNF3yxCbHAFLqnfHzCo9KU0Z1yebuOOdJlog
L94TU9528keX5q9BX0nb3cPunyR7gWsHwh8Z7F+z4stZ/NC9joxt+Qi1rc2c0K7L
D0zipFx/Bcs4JPNPeHIN6EM+aOQzr/1DP3vfW2la8uylKLLhRFtFwbW6bQIDAQAB
MA0GCSqGSIb3DQEBBQUAA4IBAQDVbb/1M6b5a1YMsjDtPecGy1ImG6hGrLOTkQ2w
y5CFfKKhjqKaYYn2Hc0rZHduryyP+Vn/krMTrnp/3TEZEy4mwHfpQ2sE1h6UNHRn
u6EHD9Hxw9cczCC3cg/kgCsIGdWuZ2vDBYe5PuN2/dOvmvINZo0wPrmUe8d1bVtw
rZlbQBopLM/hkaPZRy3Igoopg85PsX94tdVb1NmZY6Yzzmo/mM76DglDFId98MJn
Dm23mA6laGuVYvlBoeHmbpcxrdXR234ZkfNrXmhA0peuNW15Ztc4wIHFR6yFbqJJ
```

```
GbD7cN0ajmYceJjgP/fS6jn9B04Ut/iI8LHz5qKhw4p2bzAp
-----END CERTIFICATE-----
PKCS7 Data
Shrouded Keybag: pbeWithSHA1And3-KeyTripleDES-CBC, Iteration 1
Bag Attributes
    friendlyName: democa
    localKeyID: C5 EB CC 64 44 4D E4 5C C5 ED 20 05 DA D9 9C 9B 9E 03 F2 1B
Key Attributes: <No Attributes>
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,284FFCC3039E16AB

WSWKtsinYDhJRPl1Fd43LI9mhvo0aYJcDFV6OqY1MgKqR7XmST0RNbt0uyOaYhkN
hCz2E5nkxzERNKikk39eirSqoKUUqKhjb+ixjos8RnU2G8iTk9IIt9MLfLD0bQfS
Qlr2So7kl1v2qJcNbQ3ycwyvaNPB5bfpZJEVuKE11HMoOm0wxl2GWGqo2QoP6r/F
kTwWRFsABkarC5+lGRwFwCoBSpk8/g5nwMeUMy/XGfR6FZ0RyyNyH/8XkbXhq6gs
zP7WYRDs+S6XA+QB/BTNIUKIR/xGIhD9w1dkpBrs3HJvrRUOAbIN38j6IRAQo/Fa
6jpc0oFak6DAfD9rvJnwlq1eU0HqoylVLRj6CgX4LNMCo23NyZmIsQCGANRxYh7f
ZMZSXI+LEonNosMDZX3NFcfg/nHaMTXVR1bF4wkMqEJWsLbHkfDl4le9Zg4gnMRp
tnz5/a30x+8Sqidh74wBD+6ujiWAVbTUTVPs7lnTYG4HiFG/kDfSJQ0/1wCVB8v4
RMKMWEJPX/QaETCT8EZkH1Zq/JSAZU21/gKMRaC7YaFqQpDSpQonIt3WSJRwEQcM
/u+C3MBxSYYPHz087fIcBP848s1O/pVSl3016e1sdL8TXYXYfF8Eh8t5YbZtNIUi
+M1bynmde1/r0SFIEiwy906vimFAEtuv5C61VIK/4IApiTLmSCl5ZF6eNZOdHrPo
rITC7dqQyrhTupRTIEDHwQByIJbZzM+BAakiCfLVfq4N0CFcUicg356N40ihneL8
F98KFOPXU+DoJ3n9O1TXiCC2PCb/0bCloTozOvatTTzwGwi7P4gj2EkU/x6lvfYy
yPx2xTSMWookfNFSg8VwELBPG4JDfIQVjhJKe/pnVhAIKLaxpzmlCVPZJqp+h+nb
TDXJLFtVSpPqKVRbZyrzGb6HM8VEHvqS5VbKsHGrJ1ficNbvjsgvksPsXTDIB0Mi
5eieLQUdxDuSrMjFMCl4LAk3B9wFQWSZ1ZcaZJtIBy0EOpB0ldcMvIQ40q1sNbax
m78UvZ2Q/7BxhR4h7rtIFUqg8Yzs/MYehu4YyrUzoF1JOpvqCiRcRDSDM0caycU3
WM9IyAmjAj62HpBZZBxBmuq5d7liYfrwVrEMdXaKL6Q5aH8vKFqvTyp7zHIObgps
iLdcVQObk9gEt4ZRjjs6Y+BOARcNZ42LzBG7/J4P2iYuEjJItCAlYgidBMioP+kH
NeSM65N9F05JIeIxpAuAszzLtyY1oXtF+oqCTWEupBorNKwZdsJ9EDXG1picg14G
KDHWHu8nyEYgHZqit5tm81z2AYYKQ2pg/pLV9vqWNN28xFSg3TvjukG9184ao+Ld
v33GxH+szpO73BhFWXpDeiEiwqsv25/8rbkQVFx1SnxNz2bZMhTjvI3QvmNXqL/t
2Bs8bGi8W9GLvCCX0pYKS9cPgwlRHk1/o1WcJ7uyvPyUvgrT+V5zFHOyz/z/gs6C
WiY2L40xfZOn8hMDiyPPfph1aAxdAcfAYfyvygtnRCcAu/AH4pF2cqAmp/vHqMEy
NhYJHxiE969CnBOx0vk6QQhY2EQEB4lANsnhAfg+i5K6j3OddaT+kw==
-----END RSA PRIVATE KEY-----

C:\JCAPS6U1Projects\SecMail\pki>C:\jdk1.5.0_18\bin\keytool -storetype pkcs12 -keystore
ca\\democa\\democa.pkcs12.keystore.p12 -list -v -storepass democaexport

Keystore type: pkcs12
Keystore provider: SunJSSE

Your keystore contains 1 entry

Alias name: democa
Creation date: 4/05/2009
Entry type: keyEntry
Certificate chain length: 1
Certificate[1]:
Owner: EMAILADDRESS=certification@authority.com, CN=democa, OU=democa Security
Division, O=democa Certification Authority, L=Sydney, ST=NSW, C=AU
Issuer: EMAILADDRESS=certification@authority.com, CN=democa, OU=democa Security
Division, O=democa Certification Authority, L=Sydney, ST=NSW, C=AU
Serial number: 5
Valid from: Mon May 04 09:16:05 EST 2009 until: Fri Jul 21 09:16:05 EST 2017
Certificate fingerprints:
         MD5:  35:C0:B9:C9:1E:F5:34:19:8E:06:D4:B9:34:C9:D0:DE
         SHA1: C5:EB:CC:64:44:4D:E4:5C:C5:ED:20:05:DA:D9:9C:9B:9E:03:F2:1B


*******************************************
*******************************************



C:\JCAPS6U1Projects\SecMail\pki>bin\openssl x509 -in ca\\democa\\democa.pem.crt -
inform PEM -text -out ca\\democa\\democa.pem2.crt

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl x509 -in ca\\democa\\democa.pem.crt -
inform PEM -out ca\\democa\\democa.der.crt -outform DER

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl x509 -in ca\\democa\\democa.pem.crt -
inform PEM -out ca\\democa\\democa.der.cer -outform DER
```

```
C:\JCAPS6U1Projects\SecMail\pki>bin\openssl x509 -in ca\\democa\\democa.der.crt -
inform DER -text -noout
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 5 (0x5)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=AU, ST=NSW, L=Sydney, O=democa Certification Authority, OU=democa
Security Division, CN=democa/emailAddress=certification@authority.com
        Validity
            Not Before: May  3 23:16:05 2009 GMT
            Not After : Jul 20 23:16:05 2017 GMT
        Subject: C=AU, ST=NSW, L=Sydney, O=democa Certification Authority, OU=democa
Security Division, CN=democa/emailAddress=certification@authority.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
                Modulus (2048 bit):
                    00:da:03:1a:0f:6c:6e:c6:31:ab:08:79:1c:b5:86:
                    94:fc:e0:8e:00:89:6f:d4:76:c2:65:c3:ef:51:17:
                    68:bb:e8:ea:89:bd:67:6d:16:ac:5e:b7:65:2f:a6:
                    4e:8d:ff:67:56:50:a2:11:ea:07:ae:51:b6:51:1a:
                    1a:64:d7:2e:82:78:2c:cd:a5:7a:42:2d:c8:db:6d:
                    c4:3f:82:8d:b0:1f:c4:55:37:e2:70:6b:2b:92:6d:
                    71:e8:c1:94:5b:89:41:c3:fc:8c:e7:0d:33:32:6c:
                    54:fe:a5:10:62:42:ac:99:99:16:18:d1:23:0a:69:
                    e9:d6:ab:3e:ba:d3:42:a8:11:66:2e:03:5d:64:a7:
                    f6:cc:1d:da:18:d1:77:cb:10:9b:1c:01:4b:aa:77:
                    c7:cc:2a:3d:29:4d:19:d7:27:9b:b8:e3:9d:26:5a:
                    20:2f:de:13:53:de:76:f2:47:97:e6:af:41:5f:49:
                    db:dd:c3:ee:9f:24:7b:81:6b:07:c2:1f:19:ec:5f:
                    b3:e2:cb:59:fc:d0:bd:8e:8c:6d:f9:08:b5:ad:cd:
                    9c:d0:ae:cb:0f:4c:e2:a4:5c:7f:05:cb:38:24:f3:
                    4f:78:72:0d:e8:43:3e:68:e4:33:af:fd:43:3f:7b:
                    df:5b:69:5a:f2:ec:a5:28:b2:e1:44:5b:45:c1:b5:
                    ba:6d
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha1WithRSAEncryption
        d5:6d:bf:f5:33:a6:f9:6b:56:0c:b2:30:ed:3d:e7:06:cb:52:
        26:1b:a8:46:ac:b3:93:91:0d:b0:cb:90:85:7c:a2:a1:8e:a2:
        9a:61:89:f6:1d:cd:2b:64:77:6e:af:2c:8f:f9:59:ff:92:b3:
        13:ae:7a:7f:dd:31:19:13:2e:26:c0:77:e9:43:6b:04:d6:1e:
        94:34:74:67:bb:a1:07:0f:d1:f1:c3:d7:1c:cc:20:b7:72:0f:
        e4:80:2b:08:19:d5:ae:67:6b:c3:05:87:b9:3e:e3:76:fd:d3:
        af:9a:f2:0d:66:8d:30:3e:b9:94:7b:c7:75:6d:5b:70:ad:99:
        5b:40:1a:29:2c:cf:e1:91:a3:d9:47:2d:c8:82:8a:29:83:ce:
        4f:b1:7f:78:b5:d5:5b:d4:d9:99:63:a6:33:ce:6a:3f:98:ce:
        fa:0e:09:43:14:87:7d:f0:c2:67:0e:6d:b7:98:0e:a5:68:6b:
        95:62:f9:41:a1:e1:e6:6e:97:31:ad:d5:d1:db:7e:19:91:f3:
        6b:5e:68:40:d2:97:ae:35:6d:79:66:d7:38:c0:81:c5:47:ac:
        85:6e:a2:49:19:b0:fb:70:dd:1a:8e:66:1c:78:98:e0:3f:f7:
        d2:ea:39:fd:07:4e:14:b7:f8:88:f0:b1:f3:e6:a2:a1:c3:8a:
        76:6f:30:29

C:\JCAPS6U1Projects\SecMail\pki>echo [ ca ]  1>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo default_ca      = CA_default            # The
default ca section  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo [ CA_default ]  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo dir            = ca\\democa                # top
dir  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo database       =
ca\\democa\\democaIssuesIndex.txt    # index file.  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo new_certs_dir  = ca\\democa\\new_certificates
# new certs dir  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo certificate    = ca\\democa\\democa.pem.crt
# The CA cert  1>>ca\\democa\\democa.conf
```

```
C:\JCAPS6U1Projects\SecMail\pki>echo serial         = ca\\democa\\democa.srl         #
serial no file  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo private_key    =
ca\\democa\\private\\democa.pem.private.key # CA private key
1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo RANDFILE       = ca\\democa\\private\\.rand    #
random number file  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo default_days   = 3000                   # how
long to certify for  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo default_crl_days= 30                    # how long
before next CRL  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo default_md     = sha1                  # md to
use  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo policy         = policy_any            # default
policy  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo ##email_in_dn    = no                    # Don't
add the email into cert DN  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo nameopt        = ca_default            # Subject
name display option  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo certopt        = ca_default            #
Certificate display option  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo ##copy_extensions = none                # Don't
copy extensions from request  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo copy_extensions = copy                  # copy
extensions from request  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo x509_extensions = v3_ca
1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo crl_extensions = crl_ext    # comment this out to
maje V2 CRL (which netscape cannot handle, apparently  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo preserve = yes  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo [ policy_any ]  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo countryName            = supplied
1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo stateOrProvinceName    = supplied
1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo organizationName       = supplied
1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo organizationalUnitName = supplied
1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo commonName             = supplied
1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo emailAddress           = supplied
1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo [ v3_ca ]  1>>ca\\democa\\democa.conf
```

```
C:\JCAPS6U1Projects\SecMail\pki>echo keyUsage=nonRepudiation, digitalSignature,
keyEncipherment, dataEncipherment, keyAgreement  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo subjectAltName = email:copy
1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo basicConstraints = critical,CA:false
1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo extendedKeyUsage =
clientAuth,serverAuth,emailProtection  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo [ crl_ext ]  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo
##authorityKeyIdentifier=keyid:always,issuer:always  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo authorityKeyIdentifier=issuer:always
1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>echo #  1>>ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>set EUName=catest

C:\JCAPS6U1Projects\SecMail\pki>set EUBasePassword=catest

C:\JCAPS6U1Projects\SecMail\pki>set EUContacteMailAddress=testcert@authority.com

C:\JCAPS6U1Projects\SecMail\pki>set EUCommonName=catest

C:\JCAPS6U1Projects\SecMail\pki>set EURoot=ca\\democa

C:\JCAPS6U1Projects\SecMail\pki>set EUKey=ca\\democa\catest.pem.private.key

C:\JCAPS6U1Projects\SecMail\pki>set EUPEMCSR=ca\\democa\\eu_requests\catest.pem.csr

C:\JCAPS6U1Projects\SecMail\pki>set EUPEMCert=ca\\democa\catest.pem.crt

C:\JCAPS6U1Projects\SecMail\pki>set EUPEM2Cert=ca\\democa\catest.pem2.crt

C:\JCAPS6U1Projects\SecMail\pki>set EUDERCert=ca\\democa\catest.der.crt

C:\JCAPS6U1Projects\SecMail\pki>set
EUPKCS12Store=ca\\democa\catest.pkcs12.keystore.p12

C:\JCAPS6U1Projects\SecMail\pki>set EUConfig=ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>set EUPass=catest

C:\JCAPS6U1Projects\SecMail\pki>set EUExportPass=catestexport

C:\JCAPS6U1Projects\SecMail\pki>echo [ req ]  1>ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>echo distinguished_name=req_distinguished_name
1>>ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>echo req_extensions = v3_req
1>>ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>echo prompt=no  1>>ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>echo [ req_distinguished_name ]
1>>ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>echo C=AU  1>>ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>echo ST=NSW  1>>ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>echo L=Sydney  1>>ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>echo O=catest via SeeBeyond Sydney
1>>ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>echo OU=catest  1>>ca\\democa\catest.conf
```

```
C:\JCAPS6U1Projects\SecMail\pki>echo CN=catest  1>>ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>echo emailAddress=testcert@authority.com
1>>ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>echo [ v3_req ]  1>>ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>echo basicConstraints = CA:FALSE
1>>ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>echo keyUsage = nonRepudiation, digitalSignature,
keyEncipherment, dataEncipherment, keyAgreement  1>>ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>echo
extendedKeyUsage=serverAuth,clientAuth,emailProtection  1>>ca\\democa\catest.conf

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl req -new -config ca\\democa\catest.conf -
days 3000 -outform PEM -out ca\\democa\\eu_requests\catest.pem.csr -keyform PEM -
passout pass:c
atest -keyout ca\\democa\catest.pem.private.key -sha1
Loading 'screen' into random state - done
Generating a 512 bit RSA private key
...++++++++++++
.............++++++++++++
writing new private key to 'ca\\democa\catest.pem.private.key'
-----

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl req -config ca\\democa\catest.conf -in
ca\\democa\\eu_requests\catest.pem.csr -inform PEM -text -noout
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=AU, ST=NSW, L=Sydney, O=catest via SeeBeyond Sydney, OU=catest,
CN=catest/emailAddress=testcert@authority.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:bd:a8:63:0b:c4:54:f4:1f:08:1f:71:bf:99:ee:
                    a8:f8:46:55:e6:58:32:4f:71:63:13:ca:18:28:e0:
                    65:11:66:65:7c:cb:aa:8f:dc:2e:09:3a:0d:78:f8:
                    44:32:53:e6:d2:ed:44:25:b0:98:50:ff:2b:b2:33:
                    10:20:b2:02:4d
                Exponent: 65537 (0x10001)
        Attributes:
        Requested Extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment, Data
Encipherment, Key Agreement
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication, E-mail
Protection
    Signature Algorithm: sha1WithRSAEncryption
        78:5f:a8:7d:c8:c4:9c:13:d7:e9:bd:a9:b3:3a:3a:02:f4:9f:
        f4:ba:6b:ec:2a:de:01:32:7f:a8:58:ec:ea:fc:64:28:a3:37:
        33:2e:ca:14:ae:c4:45:ac:a8:dc:42:f6:8f:16:b3:28:e5:33:
        a1:f7:fe:0a:96:b7:a3:a8:6c:34

C:\JCAPS6U1Projects\SecMail\pki>echo Y  1>yy.txt

C:\JCAPS6U1Projects\SecMail\pki>echo Y  1>>yy.txt

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl ca -config ca\\democa\\democa.conf -
keyfile ca\\democa\\private\\democa.pem.private.key -keyform PEM -cert
ca\\democa\\democa.pem.crt -p
assin pass:democa -verbose -in ca\\democa\\eu_requests\catest.pem.csr -out
ca\\democa\\new_certificates\catest.pem2.crt -outdir ca\\democa\\new_certificates\\
0<yy.txt
Using configuration from ca\\democa\\democa.conf
Loading 'screen' into random state - done
0 entries loaded from the database
generating index
message digest is sha1
policy is policy_any
next serial number is 00
```

```
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=AU, ST=NSW, L=Sydney, O=catest via SeeBeyond Sydney, OU=catest,
CN=catest/emailAddress=testcert@authority.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:bd:a8:63:0b:c4:54:f4:1f:08:1f:71:bf:99:ee:
                    a8:f8:46:55:e6:58:32:4f:71:63:13:ca:18:28:e0:
                    65:11:66:65:7c:cb:aa:8f:dc:2e:09:3a:0d:78:f8:
                    44:32:53:e6:d2:ed:44:25:b0:98:50:ff:2b:b2:33:
                    10:20:b2:02:4d
                Exponent: 65537 (0x10001)
        Attributes:
        Requested Extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment, Data
Encipherment, Key Agreement
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication, E-mail
Protection
    Signature Algorithm: sha1WithRSAEncryption
        78:5f:a8:7d:c8:c4:9c:13:d7:e9:bd:a9:b3:3a:3a:02:f4:9f:
        f4:ba:6b:ec:2a:de:01:32:7f:a8:58:ec:ea:fc:64:28:a3:37:
        33:2e:ca:14:ae:c4:45:ac:a8:dc:42:f6:8f:16:b3:28:e5:33:
        a1:f7:fe:0a:96:b7:a3:a8:6c:34
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'AU'
stateOrProvinceName   :PRINTABLE:'NSW'
localityName          :PRINTABLE:'Sydney'
organizationName      :PRINTABLE:'catest via SeeBeyond Sydney'
organizationalUnitName:PRINTABLE:'catest'
commonName            :PRINTABLE:'catest'
emailAddress          :IA5STRING:'testcert@authority.com'
The subject name appears to be ok, checking data base for clashes
Everything appears to be ok, creating and signing the certificate
Successfully added extensions from config
Certificate is to be certified until Jul 20 23:16:08 2017 GMT (3000 days)
Sign the certificate? [y/n]:

1 out of 1 certificate requests certified, commit? [y/n]Write out database with 1 new
entries
writing new certificates
writing ca\\democa\\new_certificates\\/00.pem
Data Base Updated

C:\JCAPS6U1Projects\SecMail\pki>echo Y   | del yy.txt

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl x509 -in
ca\\democa\\new_certificates\catest.pem2.crt -out ca\\democa\catest.pem.crt -outform
pem

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl x509 -in
ca\\democa\\new_certificates\catest.pem2.crt -out ca\\democa\catest.pem2.crt -outform
pem -text

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl ca -config ca\\democa\\democa.conf -passin
pass:democa -keyfile ca\\democa\\private\\democa.pem.private.key -cert
ca\\democa\\democa.pem
.crt -verbose -revoke ca\\democa\\new_certificates\catest.pem2.crt
Using configuration from ca\\democa\\democa.conf
Loading 'screen' into random state - done
V       170720231608Z           00      unknown /C=AU/ST=NSW/O=catest via SeeBeyond
Sydney/OU=catest/CN=catest/emailAddress=testcert@authority.com
1 entries loaded from the database
generating index
Revoking Certificate 00.
Data Base Updated
```

```
C:\JCAPS6U1Projects\SecMail\pki>bin\openssl ca -config ca\\democa\\democa.conf -passin
pass:democa -keyfile ca\\democa\\private\\democa.pem.private.key -verbose  -crldays 30
-gencr
l -out ca\\democa\\democa.pem.crl
Using configuration from ca\\democa\\democa.conf
Loading 'screen' into random state - done
R       170720231608Z   090503231609Z   00      unknown /C=AU/ST=NSW/O=catest via
SeeBeyond Sydney/OU=catest/CN=catest/emailAddress=testcert@authority.com
1 entries loaded from the database
generating index
making CRL
signing CRL

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl crl -in ca\\democa\\democa.pem.crl -inform
PEM -out ca\\democa\\democa.der.crl -outform DER

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl crl -in ca\\democa\\democa.pem.crl -inform
PEM -text -noout
Certificate Revocation List (CRL):
        Version 2 (0x1)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: /C=AU/ST=NSW/L=Sydney/O=democa Certification Authority/OU=democa
Security Division/CN=democa/emailAddress=certification@authority.com
        Last Update: May  3 23:16:09 2009 GMT
        Next Update: Jun  2 23:16:09 2009 GMT
        CRL extensions:
            X509v3 Authority Key Identifier:
                DirName:/C=AU/ST=NSW/L=Sydney/O=democa Certification
Authority/OU=democa Security
Division/CN=democa/emailAddress=certification@authority.com
                serial:05

Revoked Certificates:
    Serial Number: 00
        Revocation Date: May  3 23:16:09 2009 GMT
    Signature Algorithm: sha1WithRSAEncryption
        3b:cb:ca:09:25:f1:33:6b:b0:6c:74:25:1d:38:33:0b:81:73:
        b8:56:66:62:02:a2:32:c9:5a:0c:5f:79:b4:26:cb:00:6e:cb:
        87:1d:2c:d2:03:98:eb:97:04:4c:36:28:72:0d:70:47:21:f4:
        4a:1e:0e:b2:d0:e2:9a:19:6d:83:46:63:9e:9d:71:3e:bb:15:
        bf:a8:f1:b8:3a:48:b7:d6:55:65:e4:ac:59:d0:d0:be:e0:a8:
        49:9b:68:8f:da:b1:f5:02:09:d6:7a:ef:00:56:93:a4:42:17:
        40:97:4e:6b:8b:3f:b1:03:1d:f5:41:ca:8b:78:e3:d0:b8:2d:
        b5:47:4e:b6:35:fe:ca:9c:1b:74:20:1d:66:67:26:1d:0f:82:
        89:6e:3a:55:b8:f9:d0:54:75:87:10:c1:84:e2:c5:5b:c1:c1:
        a8:e6:d4:8a:7d:27:1e:e1:c3:95:82:b3:a4:d9:1e:5b:b7:fd:
        8e:a1:bd:46:dc:99:82:4b:02:42:d2:e3:f5:69:c2:c4:c1:ce:
        0b:3c:07:46:c0:db:fd:91:38:75:bf:22:d9:0b:cd:18:a2:e3:
        04:f7:f4:6e:74:c5:fd:03:bb:f1:39:8f:d5:77:7e:7b:cb:69:
        8d:27:e8:1d:d4:21:37:ec:f3:4a:51:7e:03:c9:79:a5:06:2f:
        b0:73:be:67

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl crl -in ca\\democa\\democa.der.crl -inform
DER -text -noout
Certificate Revocation List (CRL):
        Version 2 (0x1)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: /C=AU/ST=NSW/L=Sydney/O=democa Certification Authority/OU=democa
Security Division/CN=democa/emailAddress=certification@authority.com
        Last Update: May  3 23:16:09 2009 GMT
        Next Update: Jun  2 23:16:09 2009 GMT
        CRL extensions:
            X509v3 Authority Key Identifier:
                DirName:/C=AU/ST=NSW/L=Sydney/O=democa Certification
Authority/OU=democa Security
Division/CN=democa/emailAddress=certification@authority.com
                serial:05

Revoked Certificates:
    Serial Number: 00
        Revocation Date: May  3 23:16:09 2009 GMT
    Signature Algorithm: sha1WithRSAEncryption
        3b:cb:ca:09:25:f1:33:6b:b0:6c:74:25:1d:38:33:0b:81:73:
        b8:56:66:62:02:a2:32:c9:5a:0c:5f:79:b4:26:cb:00:6e:cb:
        87:1d:2c:d2:03:98:eb:97:04:4c:36:28:72:0d:70:47:21:f4:
        4a:1e:0e:b2:d0:e2:9a:19:6d:83:46:63:9e:9d:71:3e:bb:15:
        bf:a8:f1:b8:3a:48:b7:d6:55:65:e4:ac:59:d0:d0:be:e0:a8:
```

```
        49:9b:68:8f:da:b1:f5:02:09:d6:7a:ef:00:56:93:a4:42:17:
        40:97:4e:6b:8b:3f:b1:03:1d:f5:41:ca:8b:78:e3:d0:b8:2d:
        b5:47:4e:b6:35:fe:ca:9c:1b:74:20:1d:66:67:26:1d:0f:82:
        89:6e:3a:55:b8:f9:d0:54:75:87:10:c1:84:e2:c5:5b:c1:c1:
        a8:e6:d4:8a:7d:27:1e:e1:c3:95:82:b3:a4:d9:1e:5b:b7:fd:
        8e:a1:bd:46:dc:99:82:4b:02:42:d2:e3:f5:69:c2:c4:c1:ce:
        0b:3c:07:46:c0:db:fd:91:38:75:bf:22:d9:0b:cd:18:a2:e3:
        04:f7:f4:6e:74:c5:fd:03:bb:f1:39:8f:d5:77:7e:7b:cb:69:
        8d:27:e8:1d:d4:21:37:ec:f3:4a:51:7e:03:c9:79:a5:06:2f:
        b0:73:be:67
C:\JCAPS6U1Projects\SecMail\pki>
```

Appendix B, Step 2a

```
C:\JCAPS6U1Projects\SecMail\pki>doc\mcz\EU_01_request_a_certificate.notes.txt.cmd
msender msender@some.company.com

C:\JCAPS6U1Projects\SecMail\pki>set OPENSSL=bin\openssl

C:\JCAPS6U1Projects\SecMail\pki>set EUName=msender

C:\JCAPS6U1Projects\SecMail\pki>set EUBasePassword=msendermsender

C:\JCAPS6U1Projects\SecMail\pki>set EUContacteMailAddress=msender@some.company.com

C:\JCAPS6U1Projects\SecMail\pki>set CAName=democa

C:\JCAPS6U1Projects\SecMail\pki>set EUCommonName=msender

C:\JCAPS6U1Projects\SecMail\pki>set EURoot=msender

C:\JCAPS6U1Projects\SecMail\pki>set EUKey=msender\msender.pem.private.key

C:\JCAPS6U1Projects\SecMail\pki>set EUPEMCSR=msender\msender.pem.csr

C:\JCAPS6U1Projects\SecMail\pki>set EUPEMCert=msender\msender.pem.crt

C:\JCAPS6U1Projects\SecMail\pki>set EUPEM2Cert=msender\msender.pem2.crt

C:\JCAPS6U1Projects\SecMail\pki>set EUDERCert=msender\msender.der.crt

C:\JCAPS6U1Projects\SecMail\pki>set EUPKCS12Store=msender\msender.pkcs12.keystore.p12

C:\JCAPS6U1Projects\SecMail\pki>set EUConfig=msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>set EUPass=msendermsender

C:\JCAPS6U1Projects\SecMail\pki>set EUExportPass=msendermsender

C:\JCAPS6U1Projects\SecMail\pki>set CARoot=ca\\democa

C:\JCAPS6U1Projects\SecMail\pki>set CAEURequests=ca\\democa\\eu_requests

C:\JCAPS6U1Projects\SecMail\pki>mkdir msender

C:\JCAPS6U1Projects\SecMail\pki>echo [ req ]  1>msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>echo distinguished_name=req_distinguished_name
1>>msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>echo req_extensions = v3_req   1>>msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>echo prompt=no  1>>msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>echo [ req_distinguished_name ]
1>>msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>echo C=AU  1>>msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>echo ST=NSW  1>>msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>echo L=Sydney  1>>msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>echo O=msender  1>>msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>echo OU=msender  1>>msender\msender.conf
```

```
C:\JCAPS6U1Projects\SecMail\pki>echo CN=msender  1>>msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>echo emailAddress=msender@some.company.com
1>>msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>echo [ v3_req ]  1>>msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>echo basicConstraints = CA:FALSE
1>>msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>echo keyUsage = nonRepudiation, digitalSignature,
keyEncipherment, dataEncipherment, keyAgreement  1>>msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>echo
extendedKeyUsage=emailProtection,clientAuth,serverAuth  1>>msender\msender.conf

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl req -new -config msender\msender.conf -
days 1000 -outform PEM -out msender\msender.pem.csr -keyform PEM -passout
pass:msendermsender -ke
yout msender\msender.pem.private.key -sha1
Loading 'screen' into random state - done
Generating a 512 bit RSA private key
.........................++++++++++++
..++++++++++++
writing new private key to 'msender\msender.pem.private.key'
-----

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl req -config msender\msender.conf -in
msender\msender.pem.csr -inform PEM -text -noout
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=AU, ST=NSW, L=Sydney, O=msender, OU=msender,
CN=msender/emailAddress=msender@some.company.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:a7:e2:11:9e:0d:3c:29:28:a4:fa:ea:f6:0b:58:
                    87:35:86:f9:36:a3:4d:29:6b:d1:58:51:a4:a4:77:
                    77:9d:a4:3c:0d:be:ee:bf:9e:3d:b8:36:4c:2f:2f:
                    4c:dd:90:bd:35:20:62:5b:02:f7:3b:4f:8e:6d:f9:
                    40:5d:01:44:19
                Exponent: 65537 (0x10001)
        Attributes:
        Requested Extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment, Data
Encipherment, Key Agreement
            X509v3 Extended Key Usage:
                E-mail Protection, TLS Web Client Authentication, TLS Web Server
Authentication
    Signature Algorithm: sha1WithRSAEncryption
        3a:fc:7b:21:6c:ab:51:5d:2b:ee:8a:1b:55:c7:cc:ea:10:aa:
        48:55:f3:04:45:3f:ae:65:be:27:85:34:a3:e7:4f:04:6e:b4:
        7e:bd:7e:de:80:84:98:88:97:8c:84:8d:a2:c4:4c:6a:8f:bd:
        77:c6:3c:a7:cf:55:4b:a3:2e:91

C:\JCAPS6U1Projects\SecMail\pki>copy msender\msender.pem.csr ca\\democa\\eu_requests
        1 file(s) copied.

C:\JCAPS6U1Projects\SecMail\pki>
```

Appendix C, Step 2b

```
C:\JCAPS6U1Projects\SecMail\pki>doc\mcz\CA_02_isse_and_revoke_end_use_cers.notes.cmd
msender

C:\JCAPS6U1Projects\SecMail\pki>set CAName=democa

C:\JCAPS6U1Projects\SecMail\pki>set BasePassword=democa

C:\JCAPS6U1Projects\SecMail\pki>set contacteMailAddress=certification@authority.com
```

```
C:\JCAPS6U1Projects\SecMail\pki>set CARoot=ca\\democa

C:\JCAPS6U1Projects\SecMail\pki>set CAKey=ca\\democa\\private\\democa.pem.private.key

C:\JCAPS6U1Projects\SecMail\pki>set CAPEMCert=ca\\democa\\democa.pem.crt

C:\JCAPS6U1Projects\SecMail\pki>set CAPEM2Cert=ca\\democa\\democa.pem2.crt

C:\JCAPS6U1Projects\SecMail\pki>set CADERCert=ca\\democa\\democa.der.crt

C:\JCAPS6U1Projects\SecMail\pki>set
CAPKCS12Store=ca\\democa\\democa.pkcs12.keystore.p12

C:\JCAPS6U1Projects\SecMail\pki>set CAReqConfig=ca\\democa\\democaReq.conf

C:\JCAPS6U1Projects\SecMail\pki>set CAConfig=ca\\democa\\democa.conf

C:\JCAPS6U1Projects\SecMail\pki>set CAPass=democa

C:\JCAPS6U1Projects\SecMail\pki>set CAExportPass=democaexport

C:\JCAPS6U1Projects\SecMail\pki>set CANewCertsDir=ca\\democa\\new_certificates

C:\JCAPS6U1Projects\SecMail\pki>set CACRLPEMFile=ca\\democa\\democa.pem.crl

C:\JCAPS6U1Projects\SecMail\pki>set CACRLDERFile=ca\\democa\\democa.der.crl

C:\JCAPS6U1Projects\SecMail\pki>set CAEURequests=ca\\democa\\eu_requests

C:\JCAPS6U1Projects\SecMail\pki>set OPENSSL=bin\openssl

C:\JCAPS6U1Projects\SecMail\pki>set EUName=msender

C:\JCAPS6U1Projects\SecMail\pki>set EUBasePassword=msendermsender

C:\JCAPS6U1Projects\SecMail\pki>set EUCommonName=msender

C:\JCAPS6U1Projects\SecMail\pki>set EURoot=msender

C:\JCAPS6U1Projects\SecMail\pki>set EUPEMCSR=ca\\democa\\eu_requests\msender.pem.csr

C:\JCAPS6U1Projects\SecMail\pki>set EUPEMCert=msender\msender.pem.crt

C:\JCAPS6U1Projects\SecMail\pki>set EUPEM2Cert=msender\msender.pem2.crt

C:\JCAPS6U1Projects\SecMail\pki>set EUDERCert=msender\msender.der.crt

C:\JCAPS6U1Projects\SecMail\pki>echo Y  1>yy.txt

C:\JCAPS6U1Projects\SecMail\pki>echo Y  1>>yy.txt

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl ca -config ca\\democa\\democa.conf -
keyfile ca\\democa\\private\\democa.pem.private.key -keyform PEM -cert
ca\\democa\\democa.pem.crt -p
assin pass:democa -verbose -in ca\\democa\\eu_requests\msender.pem.csr -out
ca\\democa\\new_certificates\msender.pem2.crt -outdir ca\\democa\\new_certificates\\
0<yy.txt
Using configuration from ca\\democa\\democa.conf
Loading 'screen' into random state - done
R       170720231608Z   090503231609Z   00      unknown /C=AU/ST=NSW/O=catest via
SeeBeyond Sydney/OU=catest/CN=catest/emailAddress=testcert@authority.com
1 entries loaded from the database
generating index
message digest is sha1
policy is policy_any
next serial number is 01
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=AU, ST=NSW, L=Sydney, O=msender, OU=msender,
CN=msender/emailAddress=msender@some.company.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:a7:e2:11:9e:0d:3c:29:28:a4:fa:ea:f6:0b:58:
```

```
                    87:35:86:f9:36:a3:4d:29:6b:d1:58:51:a4:a4:77:
                    77:9d:a4:3c:0d:be:ee:bf:9e:3d:b8:36:4c:2f:2f:
                    4c:dd:90:bd:35:20:62:5b:02:f7:3b:4f:8e:6d:f9:
                    40:5d:01:44:19
                Exponent: 65537 (0x10001)
        Attributes:
        Requested Extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment, Data
Encipherment, Key Agreement
            X509v3 Extended Key Usage:
                E-mail Protection, TLS Web Client Authentication, TLS Web Server
Authentication
    Signature Algorithm: sha1WithRSAEncryption
        3a:fc:7b:21:6c:ab:51:5d:2b:ee:8a:1b:55:c7:cc:ea:10:aa:
        48:55:f3:04:45:3f:ae:65:be:27:85:34:a3:e7:4f:04:6e:b4:
        7e:bd:7e:de:80:84:98:88:97:8c:84:8d:a2:c4:4c:6a:8f:bd:
        77:c6:3c:a7:cf:55:4b:a3:2e:91
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'AU'
stateOrProvinceName   :PRINTABLE:'NSW'
localityName          :PRINTABLE:'Sydney'
organizationName      :PRINTABLE:'msender'
organizationalUnitName:PRINTABLE:'msender'
commonName            :PRINTABLE:'msender'
emailAddress          :IA5STRING:'msender@some.company.com'
The subject name appears to be ok, checking data base for clashes
Everything appears to be ok, creating and signing the certificate
Successfully added extensions from config
Certificate is to be certified until Jul 20 23:20:04 2017 GMT (3000 days)
Sign the certificate? [y/n]:

1 out of 1 certificate requests certified, commit? [y/n]Write out database with 1 new
entries
writing new certificates
writing ca\\democa\\new_certificates\\/01.pem
Data Base Updated

C:\JCAPS6U1Projects\SecMail\pki>echo Y   | del yy.txt

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl x509 -in
ca\\democa\\new_certificates\msender.pem2.crt -out msender\msender.pem.crt -outform
PEM

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl x509 -in
ca\\democa\\new_certificates\msender.pem2.crt -out msender\msender.pem2.crt -outform
PEM -text

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl x509 -in
ca\\democa\\new_certificates\msender.pem2.crt -out msender\msender.der.crt -outform
DER
C:\JCAPS6U1Projects\SecMail\pki>
```

Appendix D, Step 2c

```
C:\JCAPS6U1Projects\SecMail\pki>doc\mcz\EU_02_finish_certificate_acquisition.notes.txt
.cmd msender

C:\JCAPS6U1Projects\SecMail\pki>set OPENSSL=bin\openssl

C:\JCAPS6U1Projects\SecMail\pki>set EUName=msender

C:\JCAPS6U1Projects\SecMail\pki>set EUBasePassword=msendermsender

C:\JCAPS6U1Projects\SecMail\pki>set EURoot=msender

C:\JCAPS6U1Projects\SecMail\pki>set EUKey=msender\msender.pem.private.key

C:\JCAPS6U1Projects\SecMail\pki>set EUPEMCert=msender\msender.pem.crt

C:\JCAPS6U1Projects\SecMail\pki>set EUPEM2Cert=msender\msender.pem2.crt

C:\JCAPS6U1Projects\SecMail\pki>set EUDERCert=msender\msender.der.crt
```

```
C:\JCAPS6U1Projects\SecMail\pki>set EUDER2Cert=msender\msender.der.cer

C:\JCAPS6U1Projects\SecMail\pki>set EUPKCS12Store=msender\msender.pkcs12.keystore.p12

C:\JCAPS6U1Projects\SecMail\pki>set EUJKSStore=msender\msender.jks.keystore

C:\JCAPS6U1Projects\SecMail\pki>set EUPass=msendermsender

C:\JCAPS6U1Projects\SecMail\pki>set EUExportPass=msendermsender

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl x509 -in msender\msender.pem.crt -inform
PEM -text -out msender\msender.pem2.crt

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl x509 -in msender\msender.pem.crt -inform
PEM -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=AU, ST=NSW, L=Sydney, O=democa Certification Authority, OU=democa
Security Division, CN=democa/emailAddress=certification@authority.com
        Validity
            Not Before: May  3 23:20:04 2009 GMT
            Not After : Jul 20 23:20:04 2017 GMT
        Subject: C=AU, ST=NSW, O=msender, OU=msender,
CN=msender/emailAddress=msender@some.company.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:a7:e2:11:9e:0d:3c:29:28:a4:fa:ea:f6:0b:58:
                    87:35:86:f9:36:a3:4d:29:6b:d1:58:51:a4:a4:77:
                    77:9d:a4:3c:0d:be:ee:bf:9e:3d:b8:36:4c:2f:2f:
                    4c:dd:90:bd:35:20:62:5b:02:f7:3b:4f:8e:6d:f9:
                    40:5d:01:44:19
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment, Data
Encipherment, Key Agreement
            X509v3 Subject Alternative Name:
                email:msender@some.company.com
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Extended Key Usage:
                TLS Web Client Authentication, TLS Web Server Authentication, E-mail
Protection
    Signature Algorithm: sha1WithRSAEncryption
        64:b7:ba:68:41:4b:ad:6a:bd:c0:6a:21:7c:ed:a4:54:ee:0c:
        46:ee:03:f3:28:de:b9:48:ed:4e:bf:47:4e:d8:40:e9:4d:ad:
        11:f3:df:66:91:d3:b2:9b:df:7e:db:88:cc:5f:ca:2b:77:34:
        e7:fe:2b:bf:b0:7f:6d:d3:26:cf:e1:1a:e2:81:87:24:3e:f2:
        31:f7:8f:c8:bb:8d:39:35:72:2c:49:38:31:2e:ea:85:bc:9d:
        ab:a3:53:79:45:97:b8:2a:04:ca:5d:bb:91:d5:45:07:03:e1:
        e9:14:0b:61:b3:80:d0:71:64:52:0e:d1:21:28:d2:7e:9b:42:
        de:b1:3a:a8:56:97:d7:29:60:71:d2:8d:15:09:e3:13:39:8e:
        d0:41:7c:87:0f:61:a1:78:a3:86:b7:0b:35:4f:6c:20:4b:40:
        2a:19:73:30:2b:35:19:d0:70:8d:45:a3:1a:8a:24:70:95:d9:
        38:b8:92:a1:52:2a:c6:81:5b:ee:06:43:b3:39:bb:06:77:f3:
        c9:0a:bd:1b:e2:0e:ae:c6:c1:08:c1:78:ed:66:2c:4b:09:0a:
        ec:ee:1f:e4:69:7f:bb:60:ba:29:97:9b:db:59:75:02:28:e2:
        d6:a9:ed:48:3e:b6:c4:b2:52:69:59:a1:69:f8:21:3e:d5:c5:
        a8:7b:34:15

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl x509 -in msender\msender.pem.crt -inform
PEM -out msender\msender.der.crt -outform DER

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl x509 -in msender\msender.pem.crt -inform
PEM -out msender\msender.der.cer -outform DER

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl x509 -in msender\msender.der.crt -inform
DER -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
```

```
        Serial Number: 1 (0x1)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=AU, ST=NSW, L=Sydney, O=democa Certification Authority, OU=democa
Security Division, CN=democa/emailAddress=certification@authority.com
        Validity
            Not Before: May  3 23:20:04 2009 GMT
            Not After : Jul 20 23:20:04 2017 GMT
        Subject: C=AU, ST=NSW, O=msender, OU=msender,
CN=msender/emailAddress=msender@some.company.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:a7:e2:11:9e:0d:3c:29:28:a4:fa:ea:f6:0b:58:
                    87:35:86:f9:36:a3:4d:29:6b:d1:58:51:a4:a4:77:
                    77:9d:a4:3c:0d:be:ee:bf:9e:3d:b8:36:4c:2f:2f:
                    4c:dd:90:bd:35:20:62:5b:02:f7:3b:4f:8e:6d:f9:
                    40:5d:01:44:19
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment, Data
Encipherment, Key Agreement
            X509v3 Subject Alternative Name:
                email:msender@some.company.com
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Extended Key Usage:
                TLS Web Client Authentication, TLS Web Server Authentication, E-mail
Protection
    Signature Algorithm: sha1WithRSAEncryption
        64:b7:ba:68:41:4b:ad:6a:bd:c0:6a:21:7c:ed:a4:54:ee:0c:
        46:ee:03:f3:28:de:b9:48:ed:4e:bf:47:4e:d8:40:e9:4d:ad:
        11:f3:df:66:91:d3:b2:9b:df:7e:db:88:cc:5f:ca:2b:77:34:
        e7:fe:2b:bf:b0:7f:6d:d3:26:cf:e1:1a:e2:81:87:24:3e:f2:
        31:f7:8f:c8:bb:8d:39:35:72:2c:49:38:31:2e:ea:85:bc:9d:
        ab:a3:53:79:45:97:b8:2a:04:ca:5d:bb:91:d5:45:07:03:e1:
        e9:14:0b:61:b3:80:d0:71:64:52:0e:d1:21:28:d2:7e:9b:42:
        de:b1:3a:a8:56:97:d7:29:60:71:d2:8d:15:09:e3:13:39:8e:
        d0:41:7c:87:0f:61:a1:78:a3:86:b7:0b:35:4f:6c:20:4b:40:
        2a:19:73:30:2b:35:19:d0:70:8d:45:a3:1a:8a:24:70:95:d9:
        38:b8:92:a1:52:2a:c6:81:5b:ee:06:43:b3:39:bb:06:77:f3:
        c9:0a:bd:1b:e2:0e:ae:c6:c1:08:c1:78:ed:66:2c:4b:09:0a:
        ec:ee:1f:e4:69:7f:bb:60:ba:29:97:9b:db:59:75:02:28:e2:
        d6:a9:ed:48:3e:b6:c4:b2:52:69:59:a1:69:f8:21:3e:d5:c5:
        a8:7b:34:15

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl pkcs12 -export -passin pass:msendermsender
-password pass:msendermsender -in msender\msender.pem.crt -inkey
msender\msender.pem.private.
key -out msender\msender.pkcs12.keystore.p12 -name msender -des3 -nomaciter -noiter
Loading 'screen' into random state - done

C:\JCAPS6U1Projects\SecMail\pki>bin\openssl pkcs12 -info -in
msender\msender.pkcs12.keystore.p12 -passin pass:msendermsender -passout
pass:msendermsender
MAC Iteration 1
MAC verified OK
PKCS7 Encrypted data: pbeWithSHA1And40BitRC2-CBC, Iteration 1
Certificate bag
Bag Attributes
    friendlyName: msender
    localKeyID: 89 F8 27 93 EA 74 49 BE 61 76 6C B2 D5 F0 58 B8 9F F7 99 AE
subject=/C=AU/ST=NSW/O=msender/OU=msender/CN=msender/emailAddress=msender@some.company
.com
issuer=/C=AU/ST=NSW/L=Sydney/O=democa Certification Authority/OU=democa Security
Division/CN=democa/emailAddress=certification@authority.com
-----BEGIN CERTIFICATE-----
MIIDTjCCAjagAwIBAgIBATANBgkqhkiG9w0BAQUFADCBtTELMAkGA1UEBhMCQVUx
DDAKBgNVBAgTA05TVzEPMA0GA1UEBxMGU3lkbmV5MScwJQYDVQQKEx5kZW1vY2Eg
Q2VydGlmaWNhdGlvbiBBdXRob3JpdHkxITAfBgNVBAsTGGRlbW9jYSBTZWN1cml0
eSBEaXZpc2lvbjEPMA0GA1UEAxMGZGVtb2NhMSowKAYJKoZIhvcNAQkBFhtjZXJ0
aWZpY2F0aW9uQGF1dGhvcml0eS5jb20wHhcNMDkwNTAzMjMyMDA0WhcNMTcwNzIw
MjMyMDA0WjB6MQswCQYDVQQGEwJBVTEMMAoGA1UECBMDTlNXMRAwDgYDVQQKEwdt
c2VuZGVyMRAwDgYDVQQLEwdtc2VuZGVyMRAwDgYDVQQDEwdtc2VuZGVyMScwJQYJ
KoZIhvcNAQkBFhhtc2VuZGVyQHNvbWUuY29tcGFueS5jb20wXDANBgkqhkiG9w0B
AQEFAANLADBIAkEAp+IRng08KSik+ur2C1iHNYb5NqNNKWvRWFGkpHd3naQ8Db7u
```

```
v549uDZMLy9M3ZC9NSBiWwL3O0+ObflAXQFEGQIDAQABo2swaTALBgNVHQ8EBAMC
A/gwIwYDVR0RBBwwGoEYbXNlbmRlckBzb21lLmNvbXBhbnkuY29tMAwGA1UdEwEB
/wQCMAAwJwYDVR0lBCAwHgYIKwYBBQUHAwIGCCsGAQUFBwMBBggrBgEFBQcDBDAN
BgkqhkiG9w0BAQUFAAOCAQEAZLe6aEFLrWq9wGohfO2kVO4MRu4D8yjeuUjtTr9H
TthA6U2tEfPfZpHTspvfftuIzF/KK3c05/4rv7B/bdMmz+Ea4oGHJD7yMfePyLuN
OTVyLEk4MS7qhbydq6NTeUWXuCoEyl27kdVFBwPh6RQLYbOA0HFkUg7RISjSfptC
3rE6qFaX1ylgcdKNFQnjEzmO0EF8hw9hoXijhrcLNU9sIEtAKhlzMCs1GdBwjUWj
GookcJXZOLiSoVIqxoFb7gZDszm7BnfzyQq9G+IOrsbBCMF47WYsSwkK7O4f5Gl/
u2C6KZeb21l1Aiji1qntSD62xLJSaVmhafghPtXFqHs0FQ=
-----END CERTIFICATE-----
PKCS7 Data
Shrouded Keybag: pbeWithSHA1And3-KeyTripleDES-CBC, Iteration 1
Bag Attributes
    friendlyName: msender
    localKeyID: 89 F8 27 93 EA 74 49 BE 61 76 6C B2 D5 F0 58 B8 9F F7 99 AE
Key Attributes: <No Attributes>
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,843450B3A36D08C7

Jt2j+bN/aWHoh5BDkrA2OCPqYrGz/m4zyYXiQjYIwFCUqkMGDfvB/Gx/JMg/1zqK
R4ys8hSdIpm1kOgy57LwN5FkNcCKF+y0WGD4SBFfPSI9UQHtfIGrSWNT5vd37jiP
9NfS0pV53AMKJM+p4lyS/ybQyTIZKiSo0E6FoA+Mzn75c+UDNRow/bItE7oq5I4Z
kyEvC3U+vfUNgpcC6Bq1vvyphWFg+JlzulN2vz77awX5WcrFCpGShI9UQ7kfX1lQ
G3bLq6w59WmQxoiG42ZRmwT6kL1Xe5uu4bo8IAvTAHmcUgQSkJPxsxw0AdNPuYLR
iNtojhKV5BOgDLedxZ1gG7GBmVjUbB8Jg01q5wddBc+YtjeDEM5FEQxFHFjdfJde
l3z6v+sDgZmKYNt+b3ubI1jGYaPoZqcjKxh6dKdqLfA=
-----END RSA PRIVATE KEY-----

C:\JCAPS6U1Projects\SecMail\pki>C:\jdk1.5.0_18\bin\keytool -storetype pkcs12 -keystore
msender\msender.pkcs12.keystore.p12 -list -v -storepass msendermsender

Keystore type: pkcs12
Keystore provider: SunJSSE

Your keystore contains 1 entry

Alias name: msender
Creation date: 4/05/2009
Entry type: keyEntry
Certificate chain length: 1
Certificate[1]:
Owner: EMAILADDRESS=msender@some.company.com, CN=msender, OU=msender, O=msender,
ST=NSW, C=AU
Issuer: EMAILADDRESS=certification@authority.com, CN=democa, OU=democa Security
Division, O=democa Certification Authority, L=Sydney, ST=NSW, C=AU
Serial number: 1
Valid from: Mon May 04 09:20:04 EST 2009 until: Fri Jul 21 09:20:04 EST 2017
Certificate fingerprints:
         MD5:  AF:26:20:22:22:03:8F:FB:85:11:09:B1:70:CE:6F:68
         SHA1: 89:F8:27:93:EA:74:49:BE:61:76:6C:B2:D5:F0:58:B8:9F:F7:99:AE


*******************************************
*******************************************


C:\JCAPS6U1Projects\SecMail\pki>
```

Acknowledgments

This document discusses creation of PKI and cryptographic objects using the OpenSSL software. The author gratefully acknowledges the authors of the OpenSSL software.