Certificate creation and use with Cisco VCS

Deployment Guide

D14548.05

February 2012

Contents

Document revision history

The following table summarizes the changes that have been applied to this document.

Revision	Date	Description
1	November 2009	Initial release.
2	October 2010	New document styles applied. New appendices added for decoding certificates and guidance on generating certificates for use with Microsoft Office Communications Server (OCS).
3	September 2011	Updated for Microsoft Lync 2010 (Lync).
4	December 2011	Minor updates for clarification.
5	February 2012	Major clarifications and updates, including OpenSSL-specific section.

Introduction

Objectives and intended audience

This deployment guide provides instructions on how to create X.509 cryptographic certificates for use with the Cisco TelePresence Video Communication Server (Cisco VCS), and how to load them into Cisco VCS.

PKI Introduction

Public Key Infrastructure (PKI) provides the mechanisms through which communications can be secured (encrypted and integrity protected) and identities can be verified. Underlying PKI is:

- **A public/private key pair**

  A public key is used to encrypt data sent to a server, but only the private key (kept secret by the server) can be used to decrypt it.

- **Signatures of data**

  Data can be “signed” by a server, by using a combination of a cryptographic hash of the data and the server’s private key. A client can verify the signature by using the server’s public key and verifying the same hash. This ensures the data has been sent from the expected server, and has not been tampered with.

- **Certificates**

  A certificate is a wrapper around a public key, and provides information about the owner of the key. This metadata is provided in X.509 format, and typically includes the server name and contact details for the owner.

- **A certificate chain**

  A certificate can be signed by a Certificate Authority (CA) using its own private key. In turn, therefore, a certificate can be verified as being signed by a CA by checking the signature against the CA’s certificate (public key). Web browsers and other clients have a list of CA certificates that they trust, and can thus verify the certificates of individual servers.

Transport Layer Security (TLS) is the standard mechanism for securing a TCP connection between hosts on a TCP/IP network. For example, secure HTTP (HTTPS) uses TLS to encrypt and verify traffic. The following is an overview of certificate use with TLS:

1. An initial TCP connection is made, and the client sends its capabilities (including cipher suites) and a random number.
2. The sever responds with its choice of those capabilities, another random number, and its certificate.
3. The client verifies that the server certificate was issued (signed) by a CA that it trusts.
4. The client sends a “pre-master secret”, encrypted with the server’s public key.
5. This pre-master secret, combined with the exchanged random numbers (to prevent replay attacks), is used to generate a “master secret”, with which the remaining communications of this TLS session are encrypted between the client and server.

The following sections provide a brief overview of how these PKI components can be used with the Cisco VCS.

Overview of certificate use on the Cisco VCS

Cisco VCS needs certificates for:

- Secure HTTP with TLS (HTTPS) connectivity

- TLS connectivity for SIP signaling, for endpoints and neighbor zones
- LDAP authentication

The Cisco VCS can use its list of trusted Certificate Authority (CA) certificates to validate other devices connecting to it.

The Cisco VCS can use the Server Certificate and the Private key to provide a signed certificate to provide evidence that the Cisco VCS is the device it says it is. This can be used with neighboring devices such as Microsoft Lync or Cisco Unified Communications Manager, as well as administrators using the web interface.

A certificate is the identifier of Cisco VCS, and it contains names by which it is known and to which traffic is routed. If a VCS is known by multiple names for these purposes, such as if it is part of a cluster, this must be represented in the X.509 subject data, according to the guidance of RFC5922. Therefore, the certificate should contain both the FQDN of the VCS itself, and of the cluster. If a certificate is shared across cluster peers, it must list all possible peer FQDNs. The following lists show what should be included in the X.509 subject, depending on the deployment model chosen.

If the VCS is not clustered:

- Subject Common Name = FQDN of VCS
- Subject Alternate Names = Blank

If the VCS is clustered, with individual certificates per VCS:

- Subject Common Name = FQDN of VCS
- Subject Alternate Names = FQDN of VCS, FQDN of cluster

If the VCS is clustered, with a single certificate per cluster:

- Subject Common Name = FQDN of cluster
- Subject Alternate Names = FQDN of cluster, FQDN of VCS peer 1, …. FQDN of VCS peer n

Certificate generation overview

X.509 certificates may be supplied from a third party, or may be generated by, for instance, a certificate generator such as OpenSSL or a tool available in applications like Microsoft Office Communications Server (OCS). Third-party certificates supplied by recognized certificate authorities are recommended, although VCS deployments in controlled or test environments can use internally generated certificates.

This document presents three methods of generating the root certificate, client/server certificate for the VCS, and private key:

- “Certificate Generation Process using Microsoft Certification Authority and OpenSSL“ on page 7 documents the process using OpenSSL to generate the private key and certificate request, which can be signed by a CA, and documents how to do the signing with the Microsoft Certification Authority on Microsoft Lync.
- “Certificate Generation Process using OpenSSL” on page 11 documents the OpenSSL-only process, which could be used with a third party or internally managed CA.
- “Certificate generation process using Microsoft OCS” on page 15 documents the process of using Microsoft Office Communications Server (OCS).

If third party certificates are used, for mutual TLS authentication the **Server** certificate must be capable of being used as a **Client** certificate as well, thus allowing the VCS to authenticate as a client device to a neighboring server.

Certificate generation is usually a 3-stage process:

- Stage 1 – generation of a private key
- Stage 2 – creation of a certificate request.
- Stage 3 – authorization and creation of the certificate.

Loading certificates onto Cisco VCS

The Cisco VCS uses standard X.509 certificates. The certificate information must be supplied to the VCS in PEM format. Typically 3 elements are loaded:

- A list of certificates of trusted certificate authorities.
- The server certificate (which is generated by the certificate authority, identifying the ID of the certificate holder, and should be able to act as both a client and server certificate).
- The private key (used to sign data sent to the client, and decrypt data sent from the client, encrypted with the public key in the server certificate). This must only be kept on the VCS and backed up in a safe place – security of the TLS communications relies upon this being kept secret.

This process is documented in “Load certificates and private key onto Cisco VCS” on page 25.

Certificate generation process using Microsoft Certification Authority and OpenSSL

This procedure uses OpenSSL to create a certificate request, and uses the Microsoft Certification Authority application to generate the signed server certificate.

Create a certificate request using OpenSSL

From a command prompt:

1. For Windows: change to the directory where OpenSSL is installed (typically a ‘bin’ directory)
   For Mac OS X: stay in the root of the user’s directory.
2. For Windows: copy openssl.cfg to openssl_vcs.cfg
   For Mac OS X: copy /system/library/openssl/openssl.cnf to the root of the user’s directory as openssl_vcs.cfg
3. If the certificate is for a cluster of VCSs:
   a. Use a text editor to edit the openssl_vcs.cfg file that was created by the above copy command, and ensure that the line:
      ```
      “req_extensions = v3_req # The extensions to add to a certificate request”
      ```
      has no # at the beginning of the line – delete the # if it is there.
   b. Scroll down to the “`[ v3_req ]`” section and below this section title add:
      ```
      subjectAltName="DNS:<FQDN of VCS cluster>,DNS:<FQDN of VCS peer 1>,DNS:<FQDN
      of VCS peer 2>,DNS:<FQDN of VCS peer n>"
      ```
      as the bottom line of this section (before) “`[ v3_ca ]`”, filling in the details for the VCS deployment as appropriate (cluster FQDN and FQDNs of all peers).
   c. Save the file.

   No changes need to be made to the openssl_vcs.cfg file if the certificate is for a single VCS.
4. Generate a private key by running the following command:
   ```
   openssl genrsa -out privatekey.pem 2048
   ```
   The `privatekey.pem` file will be used to create the certificate request and will also be required for loading into the VCS. The file is created in the directory that the `openssl` command is run from.
5. Generate a certificate request (suitable for use with Microsoft Certification Authority) by running the following command:
   ```
   openssl req –new –key privatekey.pem –config openssl_vcs.cfg –out certcsr.der
   -outform DER
   ```
6. Enter the data requested, including:
   - Country
   - State or province
   - Locality name
   - Organization name
   - Organizational unit
   - Common name – this is the VCS cluster FQDN if the certificate is for a cluster of VCSs or it is the FQDN of the VCS if the certificate is for a single VCS
   - Email address – optional, can leave blank
   - A challenge password – optional, can leave blank
   - An optional company name – optional, can leave blank

   After entering the requested data, the certificate request file certcsr.der is now available.

To validate that DNS entries have been entered correctly into the request, the certcsr.der file can be decoded using the command:

```
openssl req –text –noout –in certcsr.der –inform DER
```

The certificate request has now been generated.

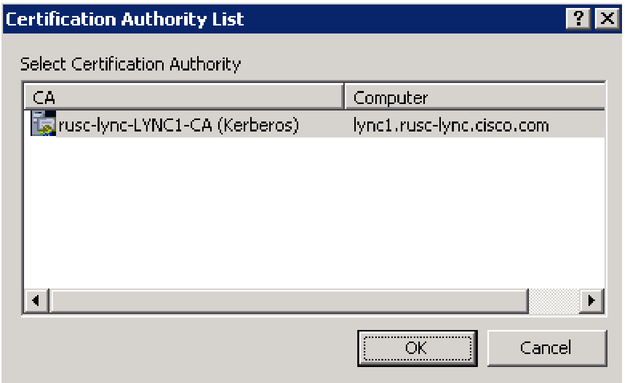
Authorize certificate request and generate a PEM certificate file using Microsoft Certification Authority

The Microsoft Certification Authority application may be installed on the Lync server, or another server in the network.

1. Copy `certcsr.der` to the server where the Microsoft Certification Authority application is installed (e.g. copy to the desktop).
2. Submit the certificate request from a command prompt by typing:

   ```
   certreq –submit –attrib "CertificateTemplate:WebServer"
   C:\Users\<user>\Desktop\certcsr.der
   ```

   This triggers the Certification Authority window to open:

   Certification Authority List

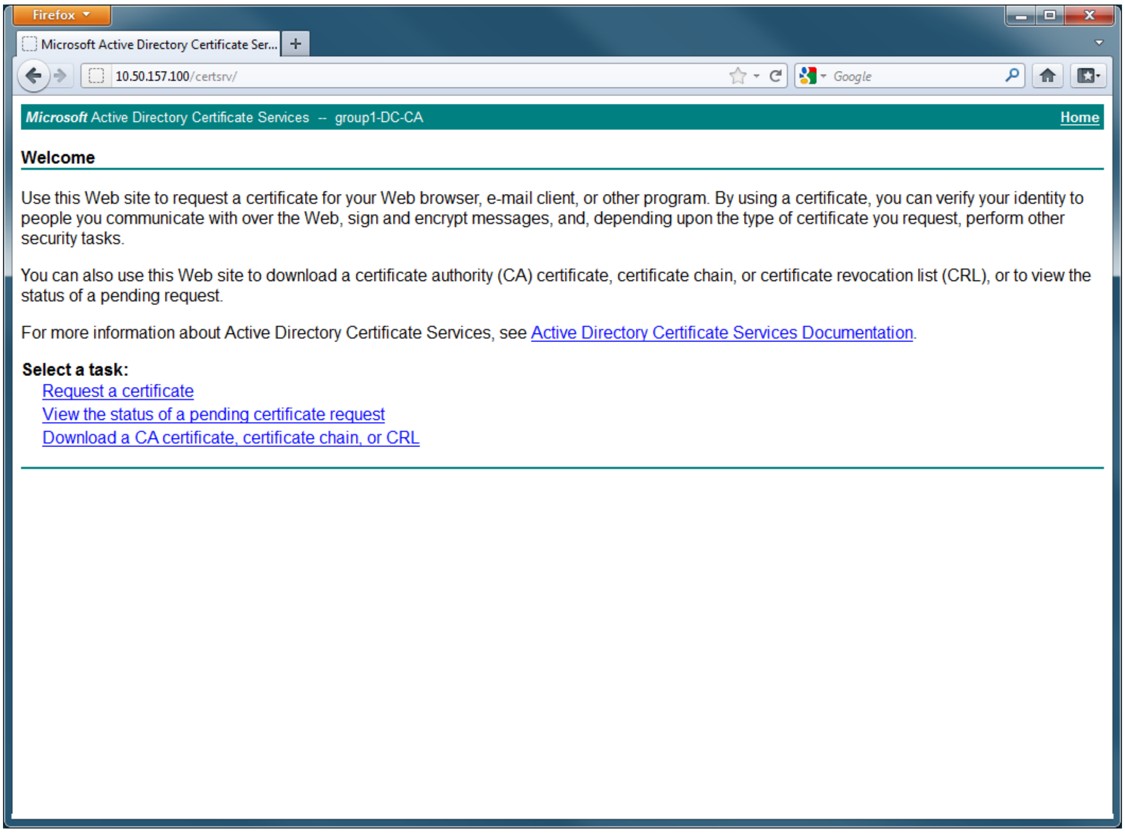
   Select Certification Authority

CA	Computer
rusc-lync-LYNC1-CA (Kerberos)	lync1.rusc-lync.cisco.com

   Note that this must be run as the administrator user.
3. Select the Authority to use (typically only one offered) and click OK.
4. When requested, save the certificate, (browse to the required folder if the default Libraries>Documents folder is not to be used) calling it `server.cer` for example.
5. Rename **`server.cer`** to **`server.pem`** for use with the Cisco VCS.

Get the Microsoft CA certificate

1. Web browse to <IP or URL of the Microsoft Certificate Server>/certsrv and log in



Microsoft Active Directory Certificate Services -- group1-DC-CA Home

Welcome

Use this Web site to request a certificate for your Web browser, e-mail client, or other program. By using a certificate, you can verify your identity to people you communicate with over the Web, sign and encrypt messages, and, depending upon the type of certificate you request, perform other security tasks.

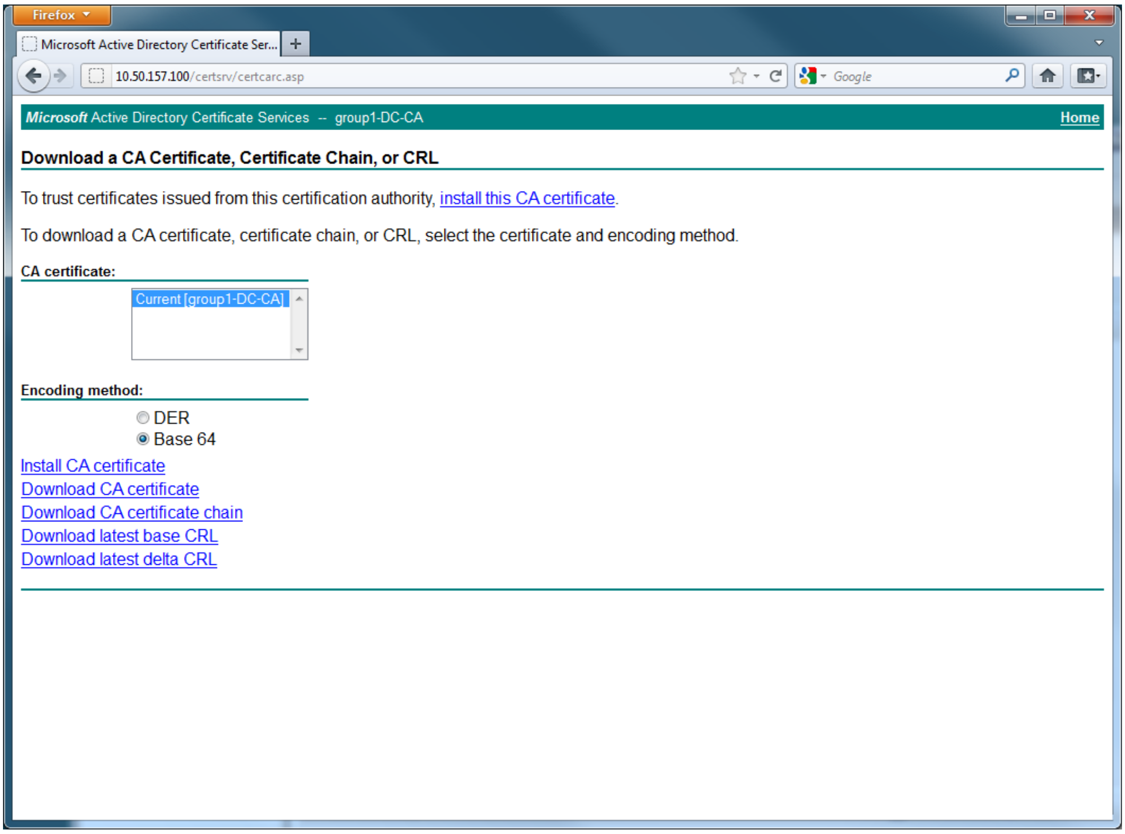
You can also use this Web site to download a certificate authority (CA) certificate, certificate chain, or certificate revocation list (CRL), or to view the status of a pending request.

For more information about Active Directory Certificate Services, see Active Directory Certificate Services Documentation.

Select a task:

Request a certificate
View the status of a pending certificate request
Download a CA certificate, certificate chain, or CRL

2. Select Download a CA certificate, certificate chain or CRL



Firefox
Microsoft Active Directory Certificate Ser...
10.50.157.100/certsrv/certcarc.asp
Google

Microsoft Active Directory Certificate Services -- group1-DC-CA Home

Download a CA Certificate, Certificate Chain, or CRL

To trust certificates issued from this certification authority, install this CA certificate.

To download a CA certificate, certificate chain, or CRL, select the certificate and encoding method.

CA certificate:

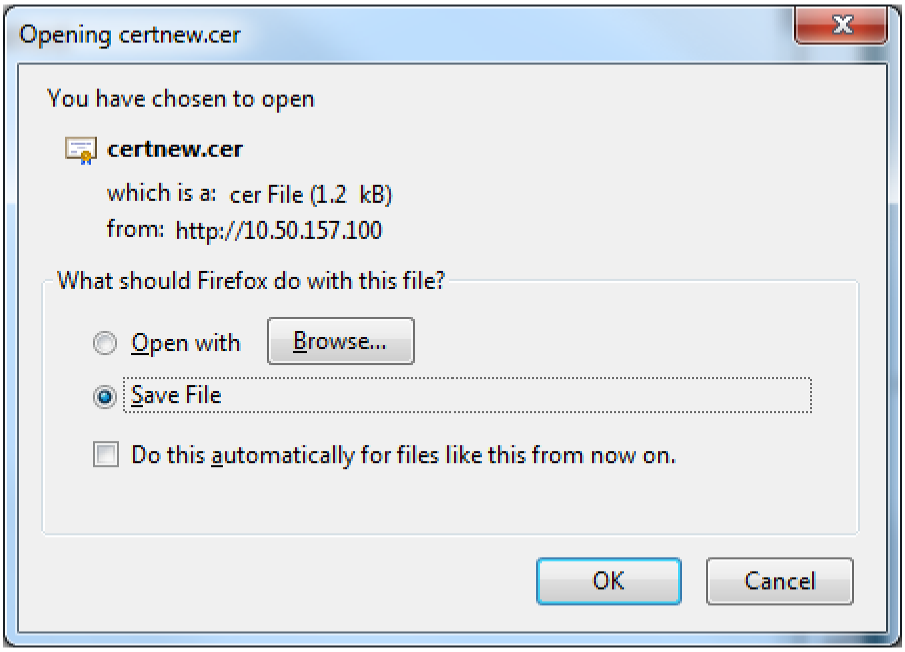
Current [group1-DC-CA]

Encoding method:

○ DER
● Base 64

Install CA certificate
Download CA certificate
Download CA certificate chain
Download latest base CRL
Download latest delta CRL

3. Select Base 64
4. Select Download CA certificate



Opening certnew.cer

You have chosen to open

certnew.cer

which is a: cer File (1.2 kB)

from: http://10.50.157.100

What should Firefox do with this file?

○ Open with

● Save File

□ Do this automatically for files like this from now on.

5. Choose **Save file** and click **OK**.
6. Rename `certnew.cer` to `certnew.pem`.

Files **`privatekey.pem`**, **`server.pem`** and **`certnew.pem`** are now available

Go to the “Load certificates and private key onto Cisco VCS” section in this document and upload **`certnew.pem, privatekey.pem`** and **`server.pem`** to VCS.

Certificate generation process using OpenSSL

This section describes first the process for generating a private key and certificate request for the Cisco VCS using OpenSSL. This is a generic process that only relies on the free OpenSSL package and not on any other software. It is appropriate when certificates are required for interfacing with neighboring devices where the Microsoft Lync and OCS tools are not available, for test purposes, and for providing output to interact with Certificate Authorities.

The output for the certificate request generation process can be given to a Certificate Authority that may be internal or external to the organization, and which can be used to produce the X.509 certificates that can be used to authenticate the Cisco VCS to neighboring devices.

This section also briefly describes how OpenSSL could be used to manage a private Certificate Authority, but does not intend to be comprehensive. Various components of these processes can be used when interfacing with third party CAs.

OpenSSL and Mac OS X or Linux

OpenSSL is already installed on Mac OS X, and is usually installed on Linux.

OpenSSL and Windows

If you do not have OpenSSL already installed, this is available as a free download from http://www.openssl.org/related/binaries.html.

- Choose the relevant 32 bit or 64 bit OpenSSL – the ‘Light’ version is all that is needed.

If you receive a warning while installing OpenSSL that C++ files cannot be found, load the “Visual C++ Redistributables” also available on this site and then re-load the OpenSSL software.

Create a certificate request using OpenSSL

This process will create a private key and certificate request for the server that can then be validated by a CA. This could be a CA that has been created and managed locally, or a third-party CA.

From a command prompt:

1. For Windows: change to the directory where OpenSSL is installed (typically a ‘bin’ directory)
   For Mac OS X or Linux: stay in the root of the user’s directory.
2. For Windows: copy openssl.cfg to openssl_vcs.cfg
   For Mac OS X: copy /System/Library/OpenSSL/openssl.cnf to the user’s current directory as openssl_vcs.cfg
   For Linux: copy /etc/ssl/openssl.cnf to the user’s current directory as openssl_vcs.cfg
3. If the certificate is for a cluster of VCSs:
   a. Use a text editor to edit the openssl_vcs.cfg file that was created by the above copy command, and ensure that the line:
      ```
      “req_extensions = v3_req # The extensions to add to a certificate request”
      ```
      has no `#` at the beginning of the line – delete the `#` if it is there.
   b. Scroll down to the “`[ v3_req ]`” section and below this section title add:
      ```
      subjectAltName="DNS:<FQDN of VCS cluster>,DNS:<FQDN of VCS peer 1>,DNS:<FQDN
      of VCS peer 2>,DNS:<FQDN of VCS peer n>"
      ```
      as the bottom line of this section (before) “`[ v3_ca ]`”, filling in the details for the VCS deployment as appropriate.
   c. Save the file.

   No changes need to be made to the openssl_vcs.cfg file if the certificate is for a single VCS.
4. Generate a private key by running the following command:
   ```
   openssl genrsa -out privatekey.pem 2048
   ```

The `privatekey.pem` file will be used to create the certificate request and will also be required for loading into the VCS. The file is created in the directory that the `openssl` command is run from.

5. Generate a certificate request by running the following command:

```
openssl req –new –key privatekey.pem –config openssl_vcs.cfg –out certcsr.pem
-outform PEM
```

6. Enter the data requested, including:
   - Country
   - State or province
   - Locality name
   - Organization name
   - Organizational unit
   - Common name – this is the VCS cluster FQDN if the certificate is for a cluster of VCSs or it is the FQDN of the VCS if the certificate is for a single VCS
   - Email address – optional, can leave blank
   - A challenge password – optional, can leave blank
   - An optional company name – optional, can leave blank

After entering the requested data, the operation completes and the certificate request file **`certcsr.pem`** is now available.

This certificate request file can be passed to an internal or third-party Certificate Authority for generating the X.509 certificate. OpenSSL can be used to operate a private CA, and this process is documented in the following section.

Operating as a Certificate Authority using OpenSSL

A major deployment will likely make use of a third-party certificate authority, or already have one internal to an organization's IT department; it is possible to use OpenSSL to manage certificates in a private certificate authority. This process is outlined below.

Configuring OpenSSL to act as a CA

OpenSSL is powerful software, and when operating as a CA, requires a number of directories and databases to be configured for tracking issued certificates.

The list of directories and files can be found in the openssl configuration file under the section **`[ CA_default ].`** By default, the three files/directories required to be created are:

- A **`demoCA`** directory in the current directory.
- An empty file called **`index.txt`** in the demoCA directory.
- A file called **`serial`** in the demoCA directory, storing the current serial number of an issued certificate. This should contain a 4-digit hexadecimal number, such as "1000".

Create a Certificate Authority using OpenSSL

This process will create a private key and certificate of a Certificate Authority (CA), which can then be used to validate other certificates. Note that this will not be trusted by devices outside of those on which it is explicitly installed.

From a command prompt:

1. For Windows: change to the directory where OpenSSL is installed (typically a 'bin' directory)
   For Mac OS X: stay in the root of the user's directory.
2. Generate a private key for the CA by running the following command:

```
openssl genrsa –des3 -out ca.key 2048
```

This will prompt for a password with which to encrypt the private key: choose a strong password and record it in a safe place. The `ca.key` file will be used to create the CA certificate and to sign other certificates.

3. Generate the CA certificate by running the following command.
   For Windows:
   ```
   openssl req –new –x509 –key ca.key –config openssl.cfg –extensions v3_ca –out
   ca.crt
   ```
   For OS X:
   ```
   openssl req –new –x509 –key ca.key –extensions v3_ca –out ca.crt
   ```
4. Enter passphrase for the key, and then enter the data requested, including:
   - Country
   - State or province
   - Locality name
   - Organization name
   - Organizational unit
   - Common name – this is typically the name of a contact person for this CA
   - Email address – optional, can leave blank

After entering the requested data, the operation completes and the certificate authority certificate **`ca.crt`** is now available.

Create a signed certificate using OpenSSL

This process will sign the server certificate with the generated CA key, using the previously generated certificate request.

From a command prompt:

1. Generate a signed server certificate by running the following command.
   ```
   openssl ca –outdir . –cert ca.crt –keyfile ca.key –infiles certcsr.pem –out
   server.pem
   ```
2. You will be prompted to enter the password for the CA's private key.

The signed certificate for the server is now available as **`server.pem`**.

Creating self-signed certificates using OpenSSL

A self-signed certificate does not use a CA, and therefore there is no chain of trust behind it. However, a self-signed certificate can be a quick way of deploying a certificate for encryption and, if it can be exchanged securely to the client, can be used for individual identity verification. Self-signed certificates are not appropriate for production deployments but may be useful in a test deployment, and thus instructions on their generation are provided below.

A CA certificate is essentially a self-signed certificate, since it is the root of the chain of trust (there is no other authority to verify it). For a self-signed certificate, however, no passkey is required when creating the key, since the server requires access to the key for communications.

From a command prompt:

1. For Windows: change to the directory where OpenSSL is installed (typically a 'bin' directory)
   For Mac OS X: stay in the root of the user's directory.
2. Generate a private key by running the following command:
   ```
   openssl genrsa -out privatekey.pem 2048
   ```
   The `server.pem` encrypt communications and will need to be copied onto the VCS. The file is created in the directory that the `openssl` command is run from.

3. Generate the CA certificate by running the following command.
   For Windows:

   ```
   openssl req –new –x509 –key privatekey.pem –config openssl.cfg –out server.pem
   ```

   For OS X:

   ```
   openssl req –new –x509 –key privatekey.pem –out server.pem
   ```

4. Enter passphrase for the key, and then enter the data requested, including:
   - Country
   - State or province
   - Locality name
   - Organization name
   - Organizational unit
   - Common name – this is typically the name of a contact person for this CA
   - Email address – optional, can leave blank

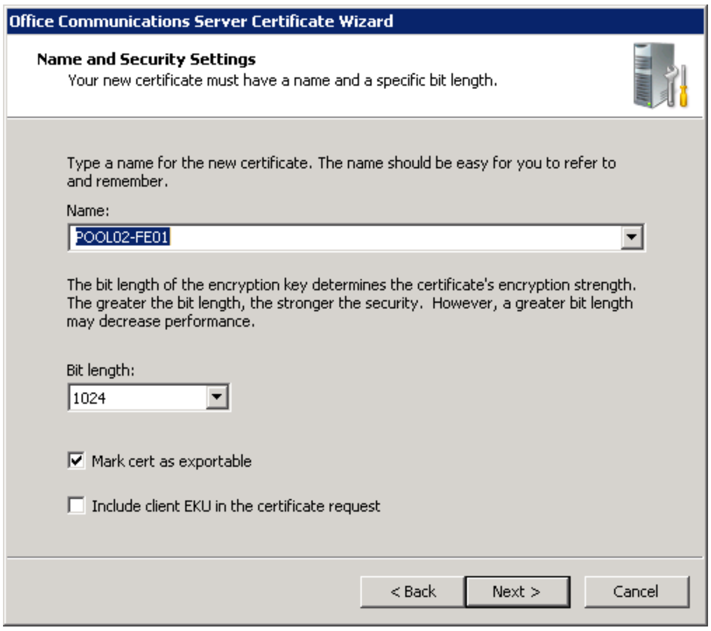
After entering the requested data, the operation completes and the server certificate **`server.pem`** is now available. The private key is stored in **`privatekey.pem`**.

Certificate generation process using Microsoft OCS

Generate a certificate request

To obtain a private key, root (CA) certificate and the server / client certificate using OCS:

1. On OCS select **Start > Administrative Tools > Office Communications Server 2007**.
2. Expand **Enterprise pools**, if available, otherwise expand **Standard Edition Servers**.
3. Expand **OCS Pool**.
4. Select the specific OCS.
5. Click **Certificates** and follow the wizard, as described in the following steps.
6. On the **Name and Security Settings** page:
    a. For the Name, select the relevant pool (usually default value).
    b. Select a **Bit length** of 2048.
    c. Select **Mark cert as exportable**.
    d. Do not include EKU in the certificate request.



Office Communications Server Certificate Wizard

Name and Security Settings

Your new certificate must have a name and a specific bit length.

Type a name for the new certificate. The name should be easy for you to refer to and remember.

Name: POOL02-FE01

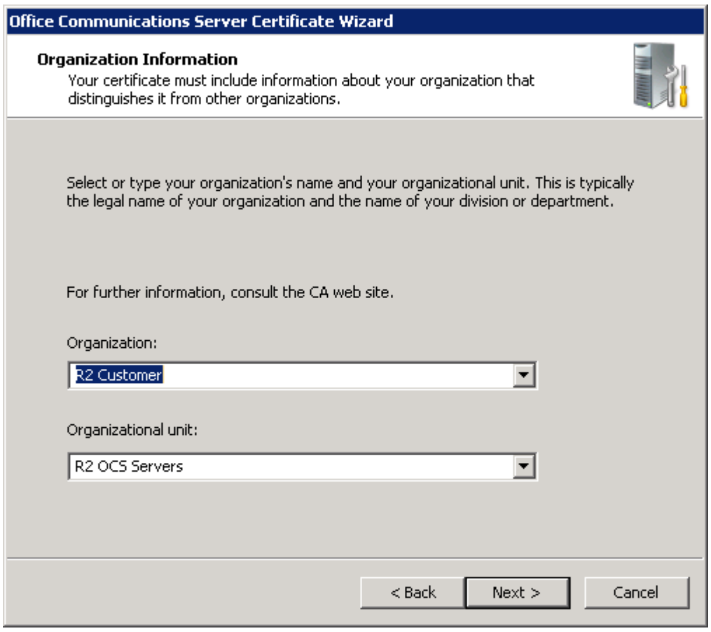
The bit length of the encryption key determines the certificate's encryption strength. The greater the bit length, the stronger the security. However, a greater bit length may decrease performance.

Bit length: 1024

☑ Mark cert as exportable

□ Include client EKU in the certificate request

7. Leave Organization Information entries as default.



Office Communications Server Certificate Wizard

Organization Information

Your certificate must include information about your organization that distinguishes it from other organizations.

Select or type your organization's name and your organizational unit. This is typically the legal name of your organization and the name of your division or department.

For further information, consult the CA web site.

Organization: R2 Customer

Organizational unit: R2 OCS Servers

8. Specify Subject Name details.

   If the certificate is for a VCS in a cluster, either:

   - Create a certificate that can be loaded onto every peer of the cluster.
   - Create individual certificates for each VCS peer.

   If creating a certificate that can be loaded onto every peer of the cluster:

   - Subject name = VCS cluster's FQDN, e.g.vcscluster1.test-customer.com
   - Subject Alternate Name = a comma separated list of the VCS cluster's FQDN and the VCS peers' FQDNs (DNS Local hostname concatenated with DNS Domain)
     e.g. vcscluster1.test-customer.com,vcspeer1.test-customer.com,vcspeer2.test-customer.com

   If creating individual certificates for each VCS peer:

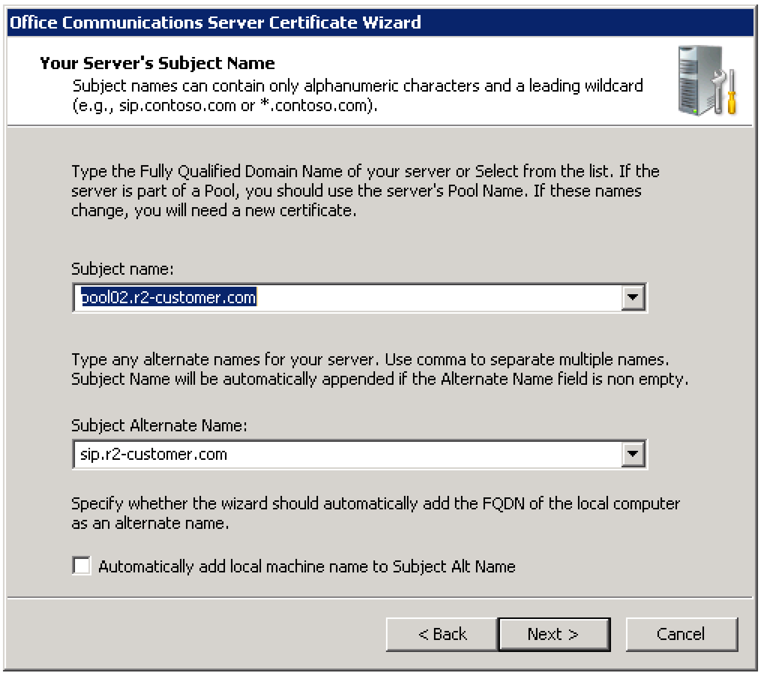
   - Subject name = VCS peer's FQDN (DNS Local hostname concatenated with DNS Domain)
     e.g. vcspeer1.test-customer.com
   - Subject Alternate Name = a comma separated list of the VCS cluster's FQDN and the VCS peer's FQDN, e.g. vcscluster1.test-customer.com,vcscluster1.test-customer.com

   If the certificate is for a standalone VCS:

   - Subject name = VCS's FQDN (DNS Local hostname concatenated with DNS Domain)
     e.g. vcs.test-customer.com
   - Subject Alternate Name = blank

   Do not select **Automatically add local machine name to Subject Alt Name** (this is for the Cisco VCS, not the OCS PC).

   - When prompted that "Subject name does not match pool name … continue?". Click **Yes**.



Office Communications Server Certificate Wizard

Your Server's Subject Name

Subject names can contain only alphanumeric characters and a leading wildcard (e.g., sip.contoso.com or *.contoso.com).

Type the Fully Qualified Domain Name of your server or Select from the list. If the server is part of a Pool, you should use the server's Pool Name. If these names change, you will need a new certificate.

Subject name: pool02.r2-customer.com

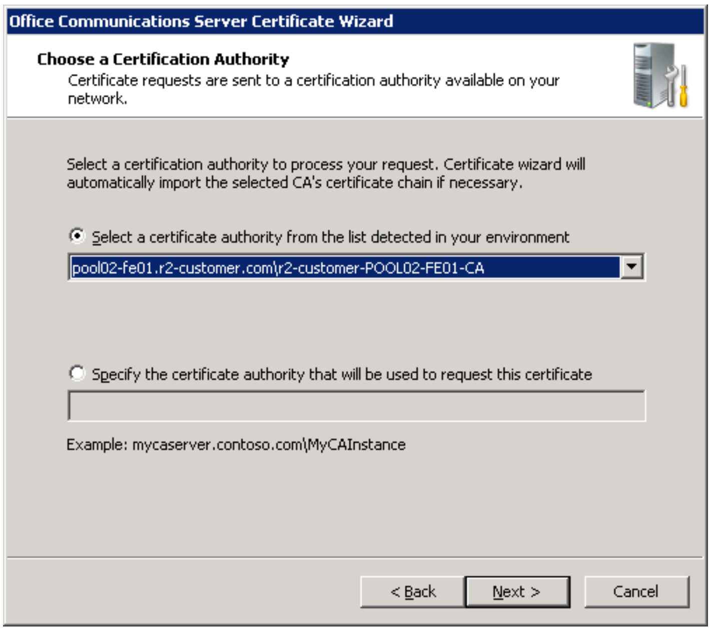
Type any alternate names for your server. Use comma to separate multiple names. Subject Name will be automatically appended if the Alternate Name field is non empty.

Subject Alternate Name: sip.r2-customer.com

Specify whether the wizard should automatically add the FQDN of the local computer as an alternate name.

☐ Automatically add local machine name to Subject Alt Name

9. Add geographical information when prompted.



Office Communications Server Certificate Wizard

Choose a Certification Authority

Certificate requests are sent to a certification authority available on your network.

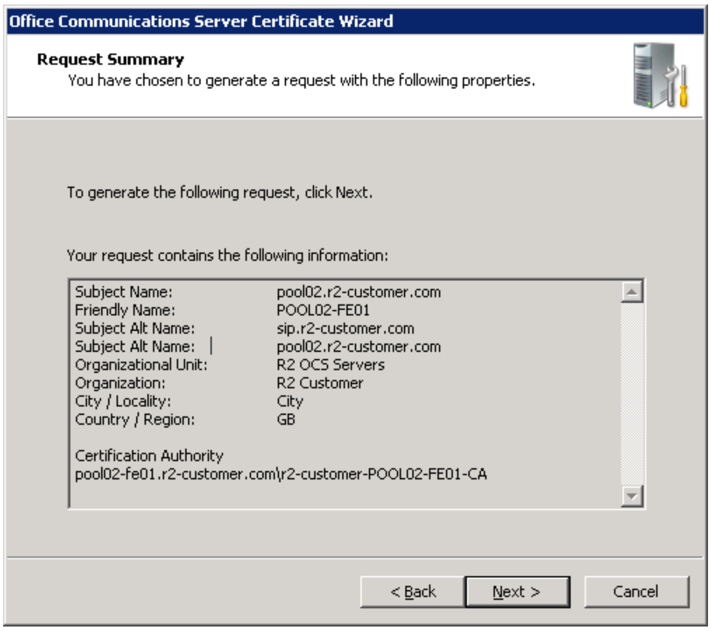
Select a certification authority to process your request. Certificate wizard will automatically import the selected CA's certificate chain if necessary.

● Select a certificate authority from the list detected in your environment

pool02-fe01.r2-customer.com\r2-customer-POOL02-FE01-CA

○ Specify the certificate authority that will be used to request this certificate

Example: mycaserver.contoso.com\MyCAInstance



You have chosen to generate a request with the following properties.

To generate the following request, click Next.

Your request contains the following information:

```
Subject Name:              pool02.r2-customer.com
Friendly Name:             POOL02-FE01
Subject Alt Name:          sip.r2-customer.com
Subject Alt Name:          pool02.r2-customer.com
Organizational Unit:       R2 OCS Servers
Organization:              R2 Customer
City / Locality:           City
Country / Region:          GB

Certification Authority
pool02-fe01.r2-customer.com\r2-customer-POOL02-FE01-CA
```

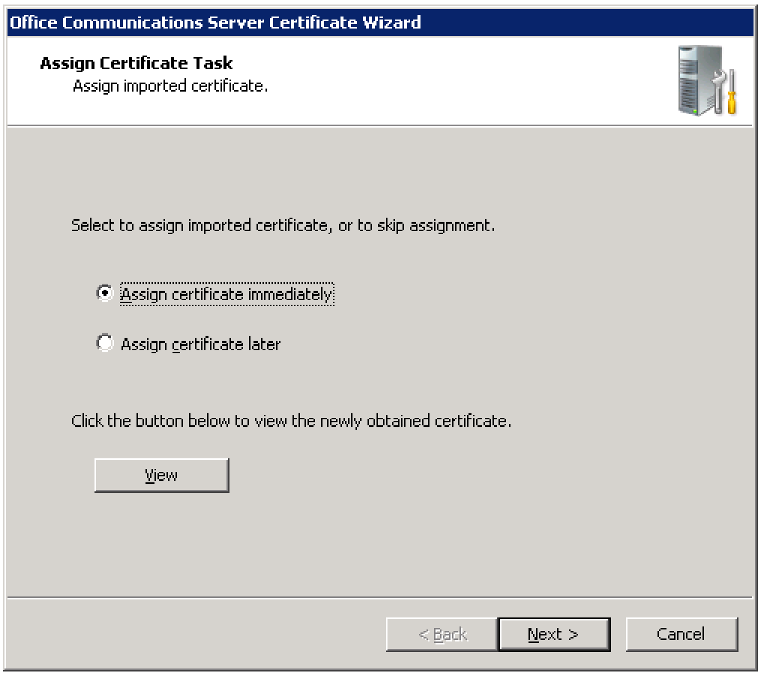
10. Stop at ‘Assign certificate immediately’.

The certificate request has now been generated. The next stage is to authorize the request.

Authorize certificate request and generate a PEM certificate file using OCS

The certificate can be created on OCS:

1. Select **Start > All Programs > Administrative Tools > Certification Authority**.
2. Choose the relevant Local Certification Authority and select **Pending Requests**.
3. Right-click on the certificate request generated above, select **All tasks**.
4. Follow the wizard until you get to the **Assign imported certificate** step.
5. Click **View**.



Assign imported certificate.

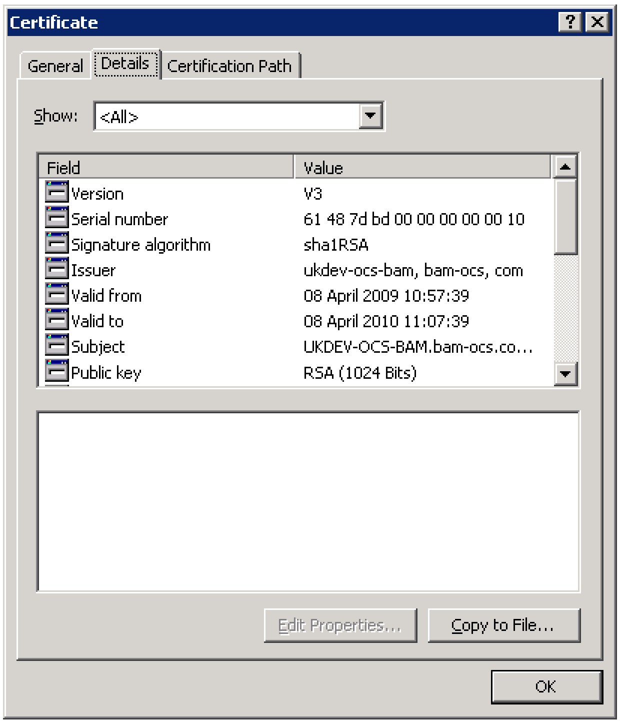
Select to assign imported certificate, or to skip assignment.

● Assign certificate immediately

○ Assign certificate later

Click the button below to view the newly obtained certificate.

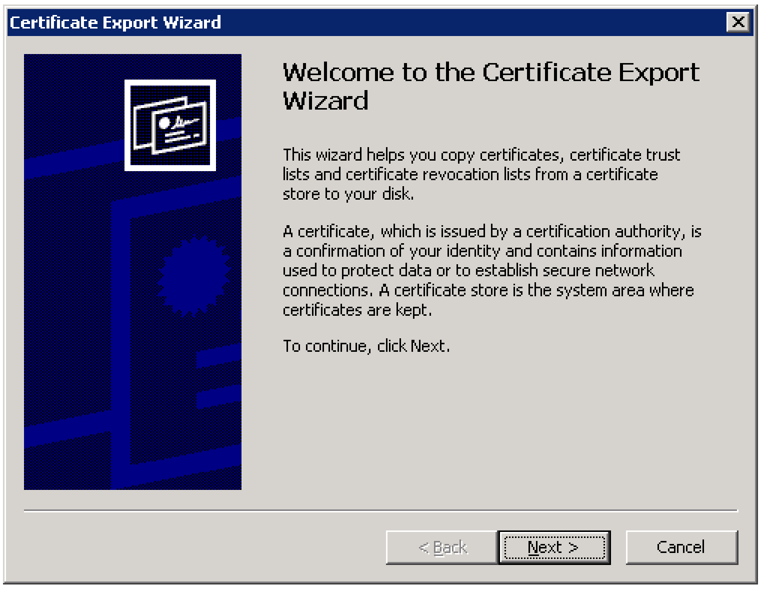
6. Select the **Details** tab, then click **Copy to File**.



Field	Value
Version	V3
Serial number	61 48 7d bd 00 00 00 00 00 10
Signature algorithm	sha1RSA
Issuer	ukdev-ocs-bam, bam-ocs, com
Valid from	08 April 2009 10:57:39
Valid to	08 April 2010 11:07:39
Subject	UKDEV-OCS-BAM.bam-ocs.co...
Public key	RSA (1024 Bits)

The Certificate Export Wizard opens.

7. Click **Next**.



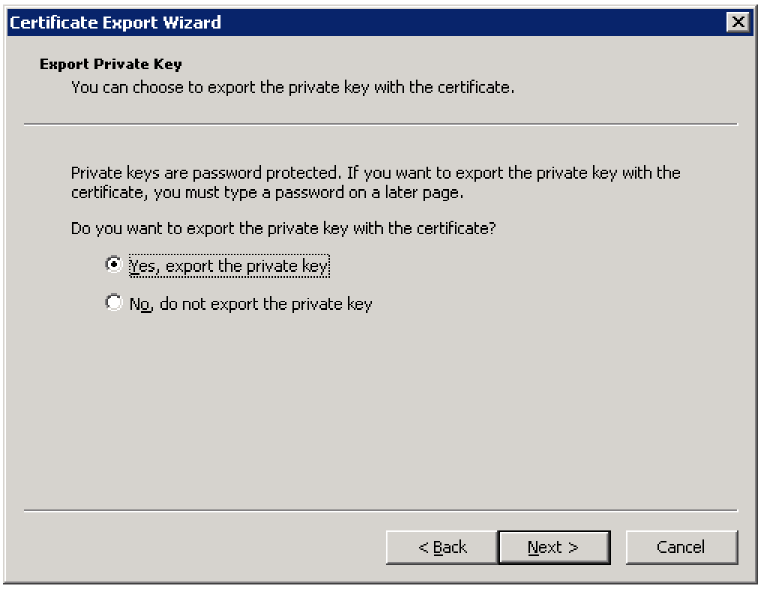
Welcome to the Certificate Export Wizard

This wizard helps you copy certificates, certificate trust lists and certificate revocation lists from a certificate store to your disk.

A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

To continue, click Next.

8. Ensure that **Yes, export the private key** is selected and click **Next**.



Certificate Export Wizard

Export Private Key

You can choose to export the private key with the certificate.

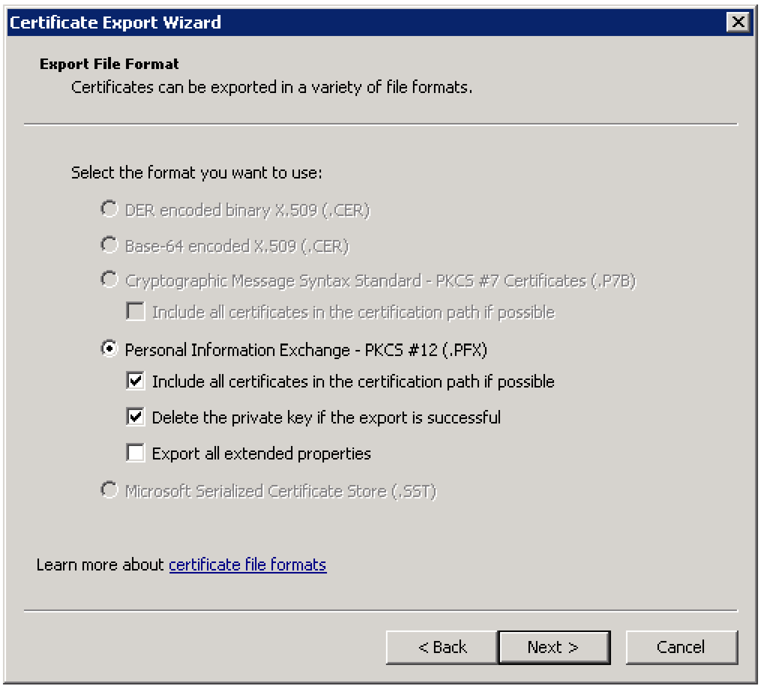
Private keys are password protected. If you want to export the private key with the certificate, you must type a password on a later page.

Do you want to export the private key with the certificate?

● Yes, export the private key

○ No, do not export the private key

9. Ensure that the selections are as shown below (Note that the order of the tick boxes may differ in different versions of Windows), then click **Next**.



Certificate Export Wizard

Export File Format

Certificates can be exported in a variety of file formats.

Select the format you want to use:

○ DER encoded binary X.509 (.CER)

○ Base-64 encoded X.509 (.CER)

○ Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)

□ Include all certificates in the certification path if possible

● Personal Information Exchange - PKCS #12 (.PFX)

☑ Include all certificates in the certification path if possible

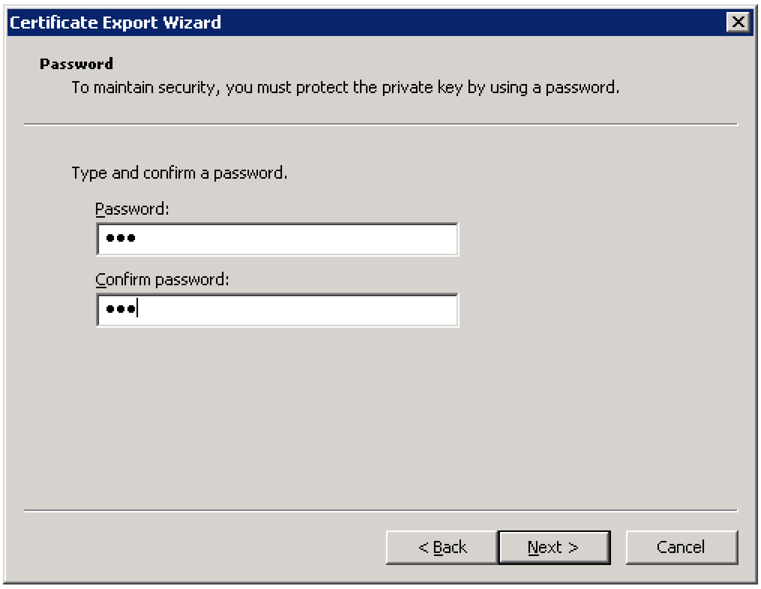
☑ Delete the private key if the export is successful

□ Export all extended properties

○ Microsoft Serialized Certificate Store (.SST)

Learn more about certificate file formats

10. Enter a **Password** that will be used to encrypt the private key (remember this as it will be needed to decrypt the private key certificate later) and click **Next**.



Certificate Export Wizard

Password

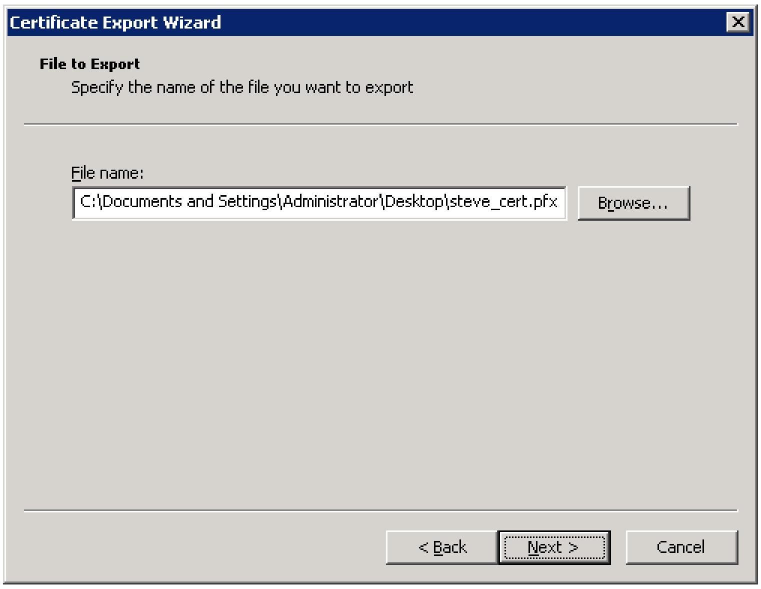
To maintain security, you must protect the private key by using a password.

Type and confirm a password.

Password: •••

Confirm password: •••

11. Enter the path and **File name** to save the certificates file (the file should have the extension .pfx) and click **Next**.



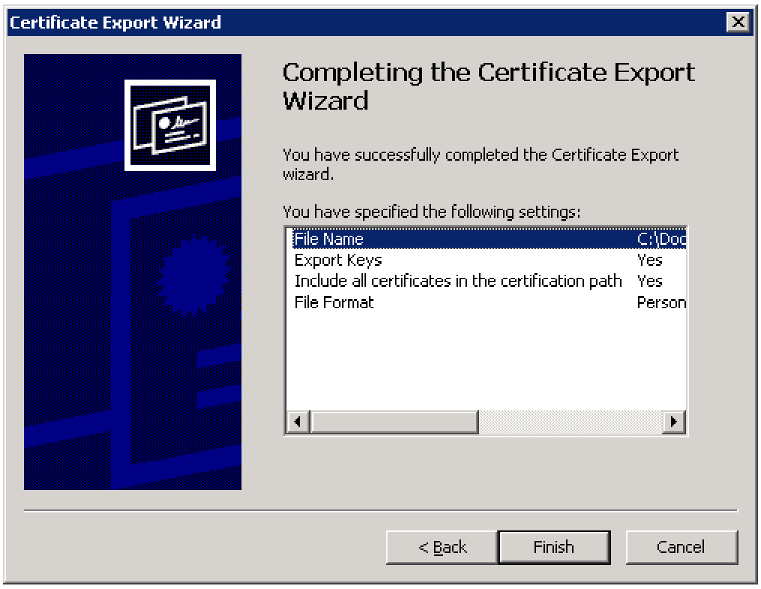
Certificate Export Wizard

File to Export

Specify the name of the file you want to export

File name: C:\Documents and Settings\Administrator\Desktop\steve_cert.pfx

12. Click **Finish**.



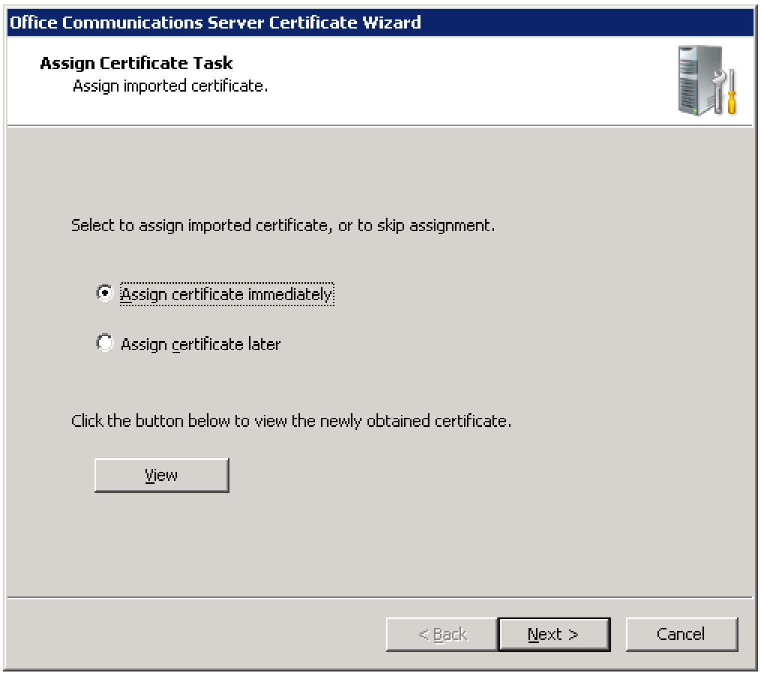
Completing the Certificate Export Wizard

You have successfully completed the Certificate Export wizard.

You have specified the following settings:

File Name	C:\Doc
Export Keys	Yes
Include all certificates in the certification path	Yes
File Format	Person

13. Certificate export was successful – click **OK**.
    You are returned to the Certificate Details page.
14. Click **OK** to close.
    You are returned to the Assign Certificate wizard:



Assign Certificate Task

Assign imported certificate.

Select to assign imported certificate, or to skip assignment.

● Assign certificate immediately

○ Assign certificate later

Click the button below to view the newly obtained certificate.

15. Click **Cancel**.

Note: Do not assign the certificate.

The certificate has now been generated and exported as a .pfx file, for example cert_inf.pfx (the .pfx contains the private key, the server certificate and root (CA) certificate in pkcs12 format).

On a Linux (Windows or other) system running openssl:

1. Copy the .pfx file to the system where openssl can be run.
2. Execute the command line:
   ```
   openssl pkcs12 -in cert_inf.pfx -out cert_inf.pem
   ```
3. Type in the password that was used during the `cert_inf.pfx` generation.
4. Enter and verify a PEM pass phrase (to be used in the next stage – removing the password from the private key).

5. Go to “Process the ‘cert_inf.pem’ file into server certificate, CA certificate and private key”.

Process the ‘cert_inf.pem’ file into server certificate, CA certificate and private key

The ‘.pem’ file contains 3 sections:

1. Private key section:

```
Bag Attributes
…
Key Attributes
   X509v3 Key Usage: 10
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,CAD24BC3A702A939
l5JTMK+Irjx2ptUWXbqEgaGwOE6UiXJE+x4Ga7sc9MXB2fvzHNehiyFHS+FoqFZK
…
7vGI/4Ezt5Stajm/p+ENGD+jMstT3a3Q85SnfvwBGVtdleKFUZ0E4Q==
-----END RSA PRIVATE KEY-----
```

2. Server certificate:

```
Bag Attributes
   localKeyID: 01 00 00 00
   friendlyName: VCS-certificate
subject=/C=NO/L=Oslo/O=net2/OU=ast/CN=vcs.net2.int
issuer=/DC=int/DC=net2/CN=net2-CA
-----BEGIN CERTIFICATE-----
MIIE3DCCA8SgAwIBAgIKI/Co1QAAAAAADzANBgkqhkiG9w0BAQUFADA9MRMwEQYK
…
6aRy7KXfeBLN/pqBEVSjjlCljmXa5hc5AGmzyTfrSRXviHE3qpmT0Lnld31f6qGK
-----END CERTIFICATE-----
```

- Note: the issuer specifies the issuer that approves this Server certificate.
- Note: the server certificate typically has a ‘friendlyName’ in the header.

3. CA certificate:

```
Bag Attributes: <Empty Attributes>
subject=/DC=int/DC=net2/CN=net2-CA
issuer=/DC=int/DC=net2/CN=net2-CA
-----BEGIN CERTIFICATE-----
MIIDVTCCAj2gAwIBAgIQQwRnazGMG5JGOacRYvwNlTANBgkqhkiG9w0BAQUFADA9
…
tQy4Hfj50yemURZ7mFCXGapcegsOC5/WDhOcIrlkcDJ2lcvBgUj4rbI=
-----END CERTIFICATE-----
```

The root CA certificate is a self signed certificate; the subject and issuer of the certificate are the same. Where there is an external certificate authority, the root CA certificate can be used from the external certificate authority.

Certificates can have hierarchical trust.

- The server certificate is trusted by (created using) a secondary certificate.
- The secondary certificate is trusted by (created using) the root CA certificate.
- The root CA certificate is self signed.

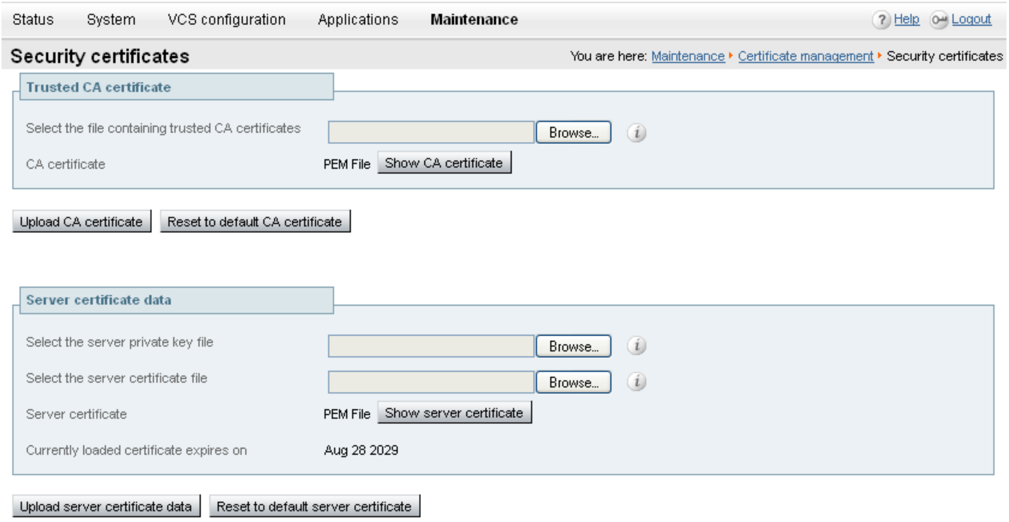
In this case, 4 entries will be found in the created PEM file.

The CA file loaded onto the Cisco VCS should consist of the full CA chain between the root CA and the server certificate (concatenated in a single file) – because to keep the hierarchy and be able to authenticate the server certificate, both of these certificates are needed. (See Appendix 3 - Example certificate with a root CA and secondary CA.)

1. Using a text editor, edit the 3 sections of the PEM file into separate files:
   - The private key section: - to a file, for example, `priv_e.pem`
   - Server certificate section: - to a file, for example, `server.pem`
   - CA certificate section: - to a file, for example, `root_ca.pem`
2. Decrypt the private key using the command line:
   ```
   openssl rsa -in priv_e.pem -out priv.pem
   ```
3. Enter the PEM pass phrase.
4. Copy the three files to the PC which is going to be used to configure the VCS Control.

Load certificates and private key onto VCS

1. On the VCS Control, go to the **Security** page (**Maintenance > Certificate management > Security certificates**) or (**Maintenance > Security**) depending on VCS version.



Security certificates

You are here: Maintenance ▸ Certificate management ▸ Security certificates

Trusted CA certificate

Select the file containing trusted CA certificates

CA certificate: PEM File

Server certificate data

Select the server private key file

Select the server certificate file

Server certificate: PEM File

Currently loaded certificate expires on: Aug 28 2029

2. For the **Trusted CA certificate**, click **Browse** (**Choose file** if using Chrome) and select the Root (CA) certificate file, for example `root_ca.pem`, or `certnew.pem` and click **Upload CA certificate**.
3. For the **Server certificate data** browse to the two files: **private key file**, for example `priv.pem`, or `privatekey.pem` and then browse to the (decrypted) **server certificate file**, for example `server.pem` and click **Upload server certificate data**.

The certificates are now loaded onto the Cisco VCS.

Appendix 1 – Converting a DER certificate file to PEM format

A private key, root (CA) certificate and the server / client certificate can be generated using 3rd party tools (or purchased from a certificate authority), and may be generated as PEM (required format, extension .pem) or DER (extension .cer) format files.

Certificates must be in PEM format for use on the Cisco VCS. Conversion from DER to PEM format can be done in one of two ways, either using OpenSSL or Windows, as documented in the following sections.

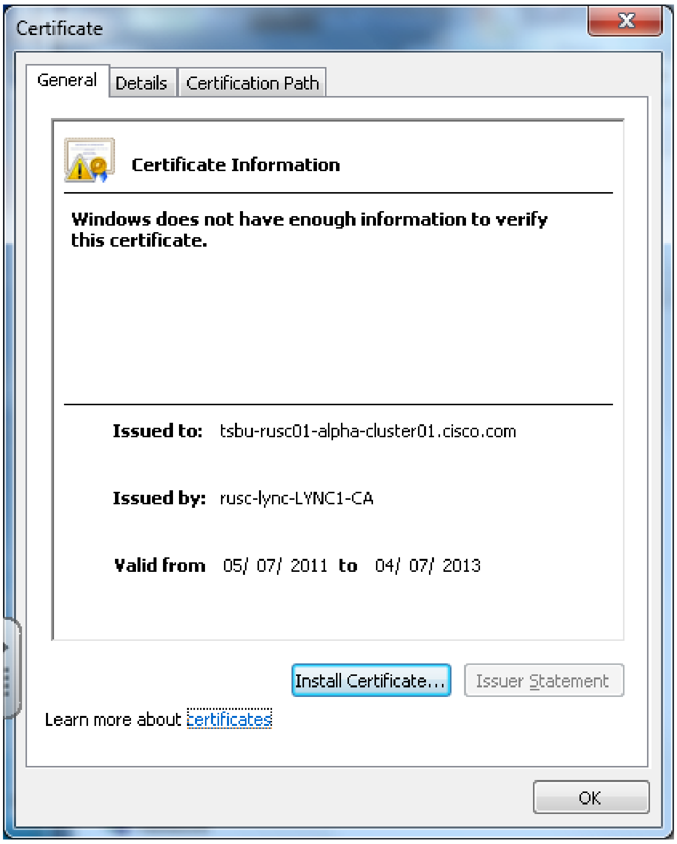
Converting a DER certificate file to a PEM file using OpenSSL

To convert from DER to PEM format, on a system running openssl, execute the command line:

```
openssl x509 -in <filename>.cer -inform DER -out <filename>.pem -outform PEM
```

Converting a DER certificate file to a PEM file using Microsoft Windows

1. Double click on the DER file to convert (this will likely have a '.cer' extension).



Certificate Information

Windows does not have enough information to verify this certificate.

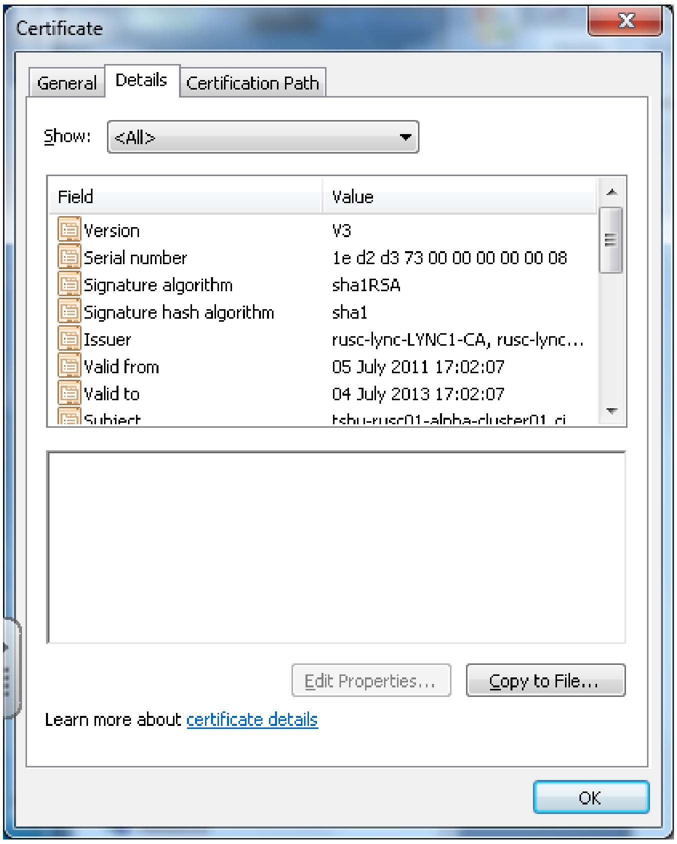
Issued to: tsbu-rusc01-alpha-cluster01.cisco.com

Issued by: rusc-lync-LYNC1-CA

Valid from 05/ 07/ 2011 to 04/ 07/ 2013

Learn more about certificates

2. Select the **Details** tab.



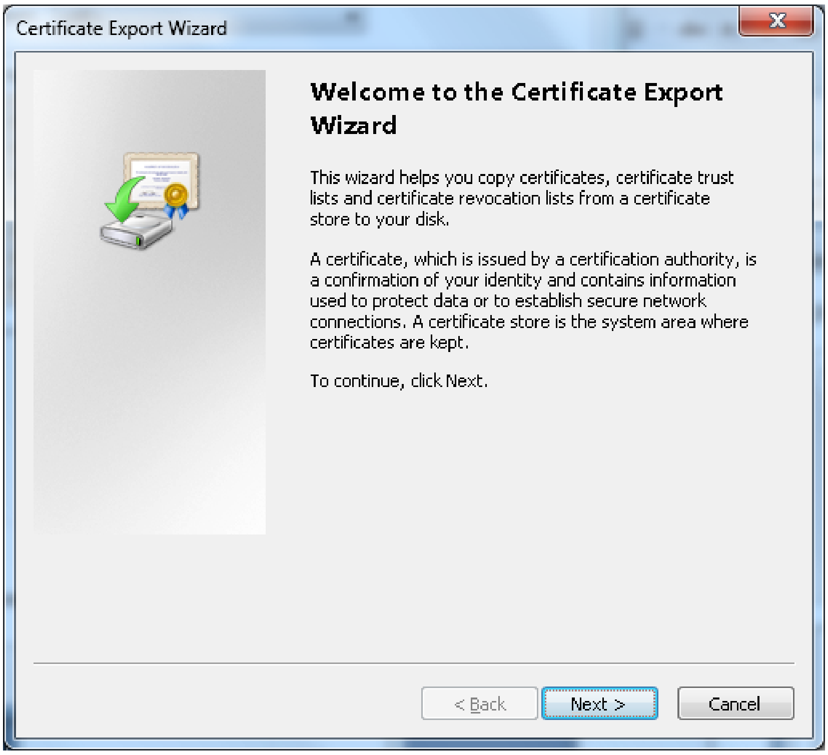
Show: <All>

Field	Value
Version	V3
Serial number	1e d2 d3 73 00 00 00 00 00 08
Signature algorithm	sha1RSA
Signature hash algorithm	sha1
Issuer	rusc-lync-LYNC1-CA, rusc-lync...
Valid from	05 July 2011 17:02:07
Valid to	04 July 2013 17:02:07
Subject	tshu-rusc01-alpha-cluster01.ci

Learn more about certificate details

3. Click **Copy to File…**

4. On the Welcome page, click **Next**.



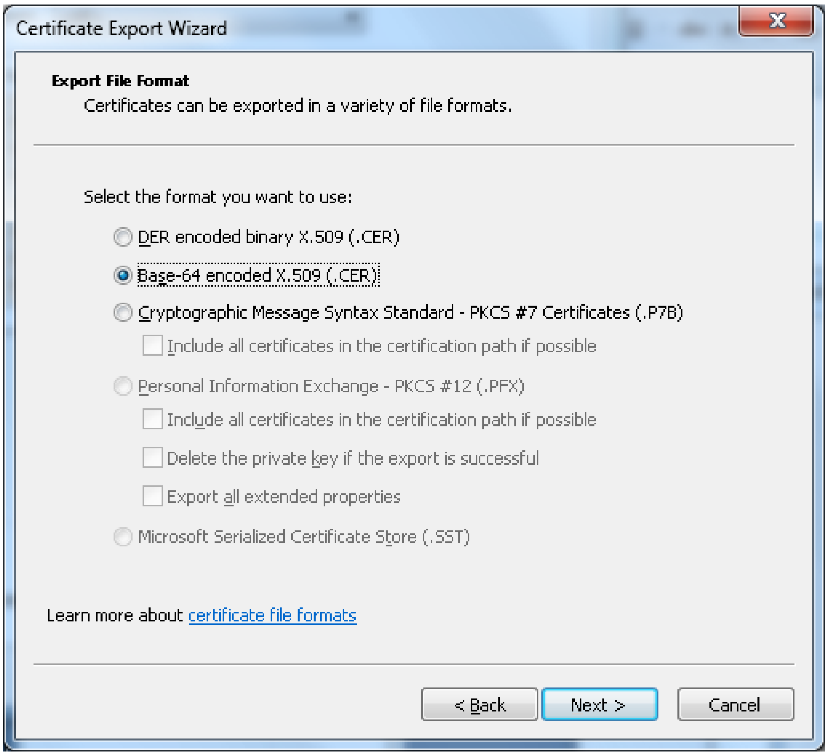
Welcome to the Certificate Export Wizard

This wizard helps you copy certificates, certificate trust lists and certificate revocation lists from a certificate store to your disk.

A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

To continue, click Next.

5. Select Base-64 encoded X.509 (.CER) and click **Next**.



Certificate Export Wizard

Export File Format

Certificates can be exported in a variety of file formats.

Select the format you want to use:

○ DER encoded binary X.509 (.CER)

● Base-64 encoded X.509 (.CER)

○ Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)

□ Include all certificates in the certification path if possible

○ Personal Information Exchange - PKCS #12 (.PFX)

□ Include all certificates in the certification path if possible

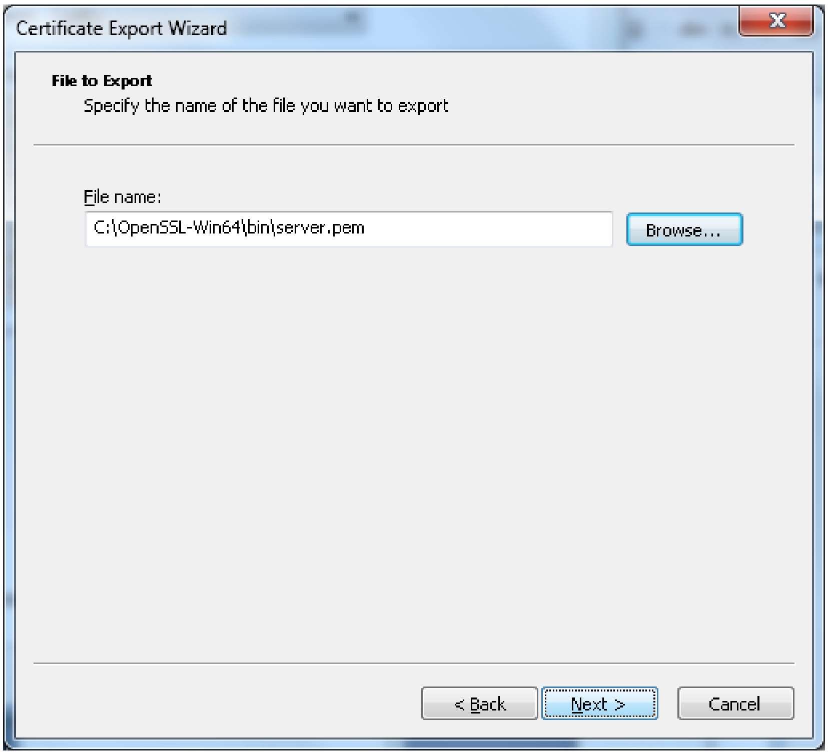
□ Delete the private key if the export is successful

□ Export all extended properties

○ Microsoft Serialized Certificate Store (.SST)

Learn more about certificate file formats

6. Click **Browse** and select required destination for file (e.g. `server.pem`) and then click **Next**.



Certificate Export Wizard

File to Export

Specify the name of the file you want to export

File name: C:\OpenSSL-Win64\bin\server.pem

7. Click **Finish**.
8. Change the filename from **`server.pem.cer`** to **`server.pem`**
9. This will be used in the "Load certificates and private key onto Cisco VCS" section of this document

Appendix 2 - Example certificate

This certificate contains:

- Private key
- Server certificate
- Root CA (self signed)

```
Bag Attributes
   Microsoft Local Key set: <No Values>
   localKeyID: 01 00 00 00
   friendlyName: le-d2c14df5-fb1f-459e-a269-dee27900c015
   Microsoft CSP Name: Microsoft RSA SChannel Cryptographic Provider
Key Attributes
   X509v3 Key Usage: 10
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,CAD24BC3A702A939

l5JTMK+Irjx2ptUWXbqEgaGwOE6UiXJE+x4Ga7sc9MXB2fvzHNehiyFHS+FoqFZK
6XrvdcB4vUrUC/PEhiBfdezVYqiYfEZRdtiFTKL9xJZ6TNo5qOcXo9kXE2gut/uc
…
8sIBOWKErI1LQBJFhaZl117KiC1j4vy+9sFyUtm+CVw72CU4HBYIv2wJM169m0FK
7vGI/4Ezt5Stajm/p+ENGD+jMstT3a3Q85SnfvwBGVtdleKFUZ0E4Q==
-----END RSA PRIVATE KEY-----
Bag Attributes
   localKeyID: 01 00 00 00
   friendlyName: VCS-certificate
subject=/C=NO/L=Oslo/O=net2/OU=ast/CN=vcs.net2.int
issuer=/DC=int/DC=net2/CN=net2-CA
-----BEGIN CERTIFICATE-----
MIIE3DCCA8SgAwIBAgIKI/Co1QAAAAAADzANBgkqhkiG9w0BAQUFADA9MRMwEQYK
CZImiZPyLGQBGRYDaW50MRQwEgYKCZImiZPyLGQBGRYEbmV0MjEQMA4GA1UEAxMH
…
wtkRwhL8xcuMHQybEDRcPZlhxDNGS9oORSCVnnpqZkbbh6fmRJNqYGjk6hM9dgMW
6aRy7KXfeBLN/pqBEVSjjlCljmXa5hc5AGmzyTfrSRXviHE3qpmT0Lnld31f6qGK
-----END CERTIFICATE-----
Bag Attributes: <Empty Attributes>
subject=/DC=int/DC=net2/CN=net2-CA
issuer=/DC=int/DC=net2/CN=net2-CA
-----BEGIN CERTIFICATE-----
MIIDVTCCAj2gAwIBAgIQQwRnazGMG5JGOacRYvwNlTANBgkqhkiG9w0BAQUFADA9
MRMwEQYKCZImiZPyLGQBGRYDaW50MRQwEgYKCZImiZPyLGQBGRYEbmV0MjEQMA4…
EGGKPlsJjToyU2E9s2yZHU1XJ7QEGDqUETmeqL3I5Bk9hXE2SotJQ1OQaYwA07MY
tQy4Hfj50yemURZ7mFCXGapcegsOC5/WDhOcIrlkcDJ2lcvBgUj4rbI=
-----END CERTIFICATE-----
```

Appendix 3 - Example certificate with a root CA and secondary CA

This certificate contains:

- Private key
- Server certificate
- Root CA (self signed)
- Secondary root certificate (signed by Root CA)

```
Bag Attributes
    1.3.6.1.4.1.311.17.2: <No Values>
    localKeyID: 01 00 00 00
    friendlyName: le-7430b4bb-13af-45c5-832d-067edcee82f3
    Microsoft CSP Name: Microsoft RSA SChannel Cryptographic Provider
Key Attributes
    X509v3 Key Usage: 10
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,6C4F771351A95DA5

eqFef6VcBEUeD3LxeyHgFyW2nOqXnbTyEldT5FswIxZAadFRfFFqE2xsozVSaaeX
/SQ3FF6lcZeW6uKTYm4ImyFKc/RNFGZv0dXEnVsVia0HWbkDeDJ8ZbyFRqKEEJaJ
…
bHIpbVYITKHmASDs0RyybY9U8y4WE6N+6F2PJWKEIWo4oeKO0vyCfQAS+q3bb8LL
Sj5XaxFzd15XDmNrkwlUINumM5LxHhAUvmo7oiiqdRTyB+57p0+rjg==
-----END RSA PRIVATE KEY-----
Bag Attributes
    localKeyID: 01 00 00 00
    1.3.6.1.4.1.311.17.3.20: B0 D2 22 E9 68 FC 0F A3 CE 17 54 95 DB 9C 65 F9 67 FF C4 69
    1.3.6.1.4.1.311.17.3.71: 4A 00 54 00 53 00 58 00 57 00 30 00 32 00 32 00 2E 00 61 00 75 00 73 00
74 00 72 00 69 00 61 00 2E 00 6C 00 6F 00 63 00 61 00 6C 00 00 00
    friendlyName: vcsctelepresence.myco.local
subject=/C=AT/ST=Vienna/L=Vienna/O=Myco TA AG/OU=ICT/CN=vcsctelepresence.myco.local
issuer=/serialNumber=35126/CN=Myco TA local 01/emailAddress=pki@email.com/O=Myco TA
AG/C=AT
-----BEGIN CERTIFICATE-----
MIIGAjCCBOqgAwIBAgIKJFzWrwAAAAC6hjANBgkqhkiG9w0BAQUFADCBgjEOMAwG
A1UEBRMFMzUxMjYxJDAiBgNVBAMTG1RlbGVrb20gQXVzdHJpYSBUQSBsb2NhbCAw
…
UAzGHbbIvX1T6kVLRTJ7heXz0BhX5ar3pWP6D7lUhGkQB+hfluUdzuePMoQaNIAG
H/LV1efZ
-----END CERTIFICATE-----
Bag Attributes
    1.3.6.1.4.1.311.17.3.20: 24 7D 5D 14 7E 6D 8D B0 88 24 9E DD 77 CF 77 C7 20 06 06 94
    1.3.6.1.4.1.311.17.3.75: 36 00 42 00 43 00 43 00 45 00 43 00 38 00 43 00 39 00 31 00 30 00 44 00
41 00 46 00 35 00 31 00 31 00 31 00 34 00 43 00 44 00 43 00 38 00 36 00 44 00 37 00 31 00 41 00 37
00 34 00 31 00 36 00 5F 00 00 00
subject=/serialNumber=32618/CN=eSignature TA Basic/OU=eSignature TA Basic/O=Myco TA
AG/C=AT
```

```
issuer=/serialNumber=32618/CN=eSignature TA Basic/OU=eSignature TA Basic/O=Myco TA
AG/C=AT
-----BEGIN CERTIFICATE-----
MIIFtTCCBJ2gAwIBAgICf2owDQYJKoZIhvcNAQEFBQAweTEOMAwGA1UEBRMFMzI2
MTgxHDAaBgNVBAMTE2VTaWduYXR1cmUgVEEgQmFzaWMxHDAaBgNVBAsTE2VTaWdu
…
b1RT/9LLr+7ly7kC2EmCYTEXTtgakin1Rx+cLA7YtT62WVFSGSISP06INQ7HXEuc
dxxKC8ccBZSlsV6eRUE5ZVSn3jIv1ALzRA==
-----END CERTIFICATE-----
Bag Attributes
    1.3.6.1.4.1.311.17.3.20: 3B AE F1 8D F2 9D 61 53 40 78 F9 81 00 F7 A7 B4 CB 27 53 1B
    1.3.6.1.4.1.311.17.3.75: 36 00 42 00 43 00 43 00 45 00 43 00 38 00 43 00 39 00 31 00 30 00 44 00
41 00 46 00 35 00 31 00 31 00 31 00 34 00 43 00 44 00 43 00 38 00 36 00 44 00 37 00 31 00 41 00 37
00 34 00 31 00 36 00 5F 00 00 00
subject=/serialNumber=35126/CN=Myco TA local 01/emailAddress=pki@telekom.at/O=Myco TA
AG/C=AT
issuer=/serialNumber=32618/CN=eSignature TA Basic/OU=eSignature TA Basic/O=Myco TA
AG/C=AT
-----BEGIN CERTIFICATE-----
MIIGSDCCBTCgAwIBAgIDAIk2MA0GCSqGSIb3DQEBBQUAMHkxDjAMBgNVBAUTBTMy
NjE4MRwwGgYDVQQDExNlU2lnbmF0dXJlIFRBIEJhc2ljMRwwGgYDVQQLExNlU2ln
…
xYSc88bZVf5JooXMImP5kPFVgLsPW5BuWgC6VIdTnopSJrYYCD65oxy/rwL+00G9
XK9EL6BuFqa7+4MZlyEVxJspqCZOfThr86IEHg==
-----END CERTIFICATE-----
```

Appendix 4 – Decoding certificates

Decoding certificates using OpenSSL

A PEM file (e.g. "**`cert.pem`**") can be decoded by issuing the following command:

```
openssl x509 -text -in cert.pem
```

A DER file (e.g. "**`cert.cer`**") can be decoded by issuing the following command:

```
openssl x509 -text –inform DER -in cert.cer
```

Decoding certificates using Firefox

The certificate in use for a website being visited can be viewed in Firefox by clicking on the security information button on the address bar, and then selecting "More Information".

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB’s public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS” WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.