RSA Key Manager and SME

This chapter describes the procedures to be followed to set up the RSA Key Manager (RKM) to work with SME.

This chapter includes the following topics:

Note RSA Key Manager is not supported for SME Disk. It is only applicable for SME Tape.

- Prerequisites for RKM, on page 1
- Configuring RKM, on page 1
- Feature History for RKM, on page 5

Prerequisites for RKM

In order to implement a complete working security solution between Cisco KMC and RKM, you need to install and set up the RKM application.

The following applications are required:

- Windows WK2, XP, or W2K3 host
- DCNM-SAN Server, Release, 3.2(3)
- OpenSSL
- JAVA JDK or JRE

Configuring RKM

The process of setting up the RKM to work with SME, involves the following tasks:

After completing these tasks, you will be able to select RSA as the key manager for SME, and create a cluster.

Installing the RKM Application

To install the RKM application, follow the instructions provided in the RSA Install Guide.

Generating CA Certificates

The files that are created during this process are stored in the /bin directory of the OpenSSL program.

To generate CA certificates, follow these steps:

Before you begin

- Generating CA certificates requires access to an OpenSSL system. You can obtain a Windows version at http://gnuwin32.sourceforge.net/packages/openssl.htm.

Step 1 Double-click openssl.exe in the directory.

Step 2 Create the key using the OpenSSL application. Enter the following command:

Example:

```
OpenSSL> genrsa -out rt.key 1024
Loading 'screen' into random state - done
Generating RSA private key, 1024 bit long modulus
.++++++
.......++++++
e is 65537 (0x10001)
```

Step 3 Set how long the certificate will be valid. Keep track of this date.

Note Use a different common name for the client and server certificates.

Example:

```
OpenSSL> req -new -key rt.key -x509 -days 365 -out rt.cert
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 (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:home
Email Address []:
```

Step 4 Create the proper pkcs12 certificate. The export password is the password needed by the SME RSA installation.

Example:

```
OpenSSL> pkcs12 -export -in rt.cert -inkey rt.key -out rt.p12
Loading 'screen' into random state - done
Enter Export Password:
Verifying - Enter Export Password:
```

Step 5 Generate a new key for the client.

Example:

```
OpenSSL> genrsa -out client.key 1024
Loading 'screen' into random state - done
Generating RSA private key, 1024 bit long modulus
...................++++++
....++++++
e is 65537 (0x10001)
```

Step 6 Create the **client.csr** file. This is the owner. The common name must be different from the issuer home.

Example:

```
OpenSSL> req -new -key client.key -out client.csr
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 (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:cae
Common Name (eg, YOUR name) []:
Email Address []:
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Step 7 Set the duration the certificate will be valid. Keep track of this date.

Example:

```
OpenSSL> x509 -req -days 365 -in client.csr -CA rt.cert -CAkey rt.key -CAcreateserial -out client.cert
Loading 'screen' into random state - done
Signature ok
subject=/C=AU/ST=wi/L=hudson/O=cisco/OU=cae/CN=mikef/emailAddress=mikef@cisco.com
Getting CA Private Key
```

Step 8 Create the pkcs12 certificate.

Example:

```
OpenSSL> pkcs12 -export -in client.cert -inkey client.key -out client.p12
Loading 'screen' into random state - done
Enter Export Password:
Verifying - Enter Export Password:
OpenSSL> genrsa -out server.key 1024
Loading 'screen' into random state - done
Generating RSA private key, 1024 bit long modulus
..++++++
..................++++++
e is 65537 (0x10001)
```

Step 9 Create the new server key. This is the owner. The common name must be different from the issuer home.

Example:

```
OpenSSL> req -new -key server.key -out server.csr
```

```
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 (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:
Email Address []:
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Step 10 Set the duration the certificate will be valid. Keep track of this date.

Example:

```
OpenSSL> x509 -req -days 365 -in server.csr -CA rt.cert -CAkey rt.key -CAcreateserial -out server.cert
Loading 'screen' into random state - done
Signature ok
subject=/C=AU/ST=wi/L=town/O=cisco/OU=tac/CN=bill/emailAddress=bill@cisco.com
Getting CA Private Key
```

Step 11 Create the pkcs12 certificate for serverpub.

Example:

```
OpenSSL> pkcs12 -export -in server.cert -inkey server.key -nokeys -out serverpub.p12
Loading 'screen' into random state - done
Enter Export Password:
Verifying - Enter Export Password:
```

Step 12 Create the pkcs12 certificate again for the server.

Example:

```
OpenSSL> pkcs12 -export -in server.cert -inkey server.key -out server.p12
Loading 'screen' into random state - done
Enter Export Password:
Verifying - Enter Export Password:
OpenSSL>
```

Creating JKS Files Using the Java Keytool

To create the JKS files needed by the DCNM-SAN using the JAVA Keytool, do the following:

Step 1 Copy **client.p12** and **serverpub.p12** that are found in the OpenSSL **/bin** directory to the DCNM-SAN Java directory tool directory **C:\Program Files\Java\jre1.5.0_11\bin**.

Step 2 From a DOS window in the Java **/bin** directory, create the JKS files needed by the SME KMC.

Example:

```
Import client PKCS12 keystore to JKS
keytool -importkeystore -srckeystore client.p12 -srcstoretype PKCS12 -destkeystore sme_rkm_client.jks
 -deststoretype JKS
Import server PKCS12 keystore to JKS
keytool -importkeystore -srckeystore serverpub.p12 -srcstoretype PKCS12 -destkeystore sme_rkm_trust.jks
 -deststoretype JKS
```

Place these keystore files in the mds9000/conf/cert directory and restart DCNM-SAN.

Placing Certificates in RKM

To place certificates in the RKM, follow these steps:

Step 1 After generating all certificates, copy the **rt.p12** file to the **C:\rkm-2.1.2-trial\certs\rt** directory.

Step 2 Copy the **server.p12** file to the **C:\rkm-2.1.2-trial\certs\server** directory.

Step 3 Restart the RKM.

Migrating From Cisco KMC to RKM

You can use RKM at the time of SME installation, or you can choose to deploy SME with the integrated Cisco KMC later. If RKM is deployed after Cisco KMC has been used alone, you need to perform an explicit key migration procedure before using RKM with SME.

This section describes the procedure for migrating encryption keys, wrap keys, and encryption policy information from Cisco KMC to RKM.

Note The migration procedure will differ when Cisco KMC uses the PostgresSQL database or the Oracle Express database for the key catalog. These differences are documented wherever applicable.

Note In Cisco MDS 9000 NX-OS Software Release 4.1(1c) and later, the keys are restored in the same state (active or deactivated) as before the migration.

Feature History for RKM

The below table lists the release history for this feature.

Table 1: Feature History for RKM

Feature Name	Releases	Feature Information
Software change	5.2(1)	In Release 5.2(1), Fabric Manager is changed to DCNM for SAN (DCNM-SAN).
	4.1(1c)	In Release 4.1(1b) and later, the MDS SAN-OS software is changed to MDS NX-OS software. The earlier releases are unchanged and all references are retained.
RKM migration procedure	4.1(1c)	Procedure to migrate from Cisco KMC to RKM is explained.