Implementing Certification Authority Interoperability

Certification authority interoperability

Certification authority (CA) interoperability is a security capability on Cisco NCS 1004 that

- supports IPSec, SSL, and SSH protocols.
- provides manageability and scalability for IPSec deployments.
- allows devices to obtain and use digital certificates issued by a CA.

CA interoperability permits devices and CAs to communicate so that a device can obtain and use digital certificates from the CA. Although IPSec can be implemented in a network without the use of a CA, using a CA provides manageability and scalability for IPSec.

Note IPSec is not currently supported on Cisco NCS 1004.

Prerequisites for CA interoperability

Ensure that the user permissions and the certification authority (CA) requirements are in place on Cisco NCS 1004 before you configure CA interoperability.

The prerequisites for implementing CA interoperability on Cisco NCS 1004 include the following requirements:

- You must be in a user group that is associated with a task group that includes the proper task IDs. The command reference guides include the task IDs that are required for each command. If you suspect that user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

- You must have a CA available to your network before you configure this interoperability feature. The CA must support the Cisco Systems PKI protocol and the Simple Certificate Enrollment Protocol (SCEP), which was formerly called Certificate Enrollment Protocol (CEP).

Maximum supported CA key size for interoperability on Cisco NCS 1004

The following restriction applies when configuring CA interoperability on Cisco NCS 1004:

- The restriction for CA interoperability on Cisco NCS 1004 is that the software does not support CA server public keys that are greater than 2048 bits.

Enroll RSA key pairs for device certificates

Secure the Cisco NCS 1004 device by obtaining signed certificates from the certification authority for each RSA key pair.

You must obtain a signed certificate from the CA for each of the RSA key pairs on the Cisco NCS 1004 device. If you generated general-purpose RSA keys, the Cisco NCS 1004 device has only one RSA key pair and needs only one certificate. If you previously generated special usage RSA keys, the Cisco NCS 1004 device has two RSA key pairs and needs two certificates.

Procedure

Step 1 Request certificates for all of your RSA key pairs using the command **crypto ca enroll** *ca-name*.

- This command causes Cisco NCS 1004 to request as many certificates as there are RSA key pairs. Run this command only once, even if you have special usage RSA key pairs.
- This command requires you to create a challenge password that is not saved with the configuration. This password is required if your certificate must be revoked, so you must remember this password.
- A certificate may be issued immediately, or Cisco NCS 1004 sends a certificate request every minute until the enrollment retry period is reached and a timeout occurs. If a timeout occurs, contact your system administrator to get your request approved, and then enter this command again.

Example:

```
RP/0/RP0/CPU0:ios# crypto ca enroll myca
```

Step 2 (Optional) Display information about the CA certificate using the command **show crypto ca certificates**.

Example:

```
RP/0/RP0/CPU0:ios# show crypto ca certificates
```

The Cisco NCS 1004 device receives signed certificates for each RSA key pair from the CA. If the request times out, contact your administrator and retry the enrollment.

Configure certificate enrollment using cut-and-paste

Set up the trustpoint CA that Cisco NCS 1004 uses, and enable manual certificate enrollment through terminal-based cut-and-paste.

Manual cut-and-paste certificate enrollment lets you import the CA certificate and your own certificate by pasting them at the terminal, instead of using a network-based enrollment protocol.

Procedure

Step 1 Enter the XR Config mode.

Example:

```
RP/0/RP0/CPU0:ios# configure
```

Step 2 Declare the CA that Cisco NCS 1004 must use, and enter trustpoint configuration mode. Use the *ca-name* argument to specify the name of the CA.

Example:

```
RP/0/RP0/CPU0:ios(config)# crypto ca trustpoint myca
```

Step 3 Specify manual cut-and-paste certificate enrollment.

Example:

```
RP/0/RP0/CPU0:ios(config-trustp)# enrollment terminal
```

Step 4 Save the configuration changes by using the **commit** or **end** command.

Step 5 Authenticate the CA by obtaining the certificate of the CA. Use the *ca-name* argument to specify the name of the CA. Use the same name that you entered in step 2.

Example:

```
RP/0/RP0/CPU0:ios# crypto ca authenticate myca
```

Step 6 Obtain the certificates for Cisco NCS 1004 from the CA. Use the *ca-name* argument to specify the name of the CA. Use the same name that you entered in step 2.

Example:

```
RP/0/RP0/CPU0:ios# crypto ca enroll myca
```

Step 7 Import a certificate manually at the terminal. Use the *ca-name* argument to specify the name of the CA. Use the same name that you entered in step 2.

Note
You must enter the **crypto ca import** command twice if usage keys (signature and encryption keys) are used. The first time the command is entered, one of the certificates is pasted into Cisco NCS 1004; the second time the command is entered, the other certificate is pasted into Cisco NCS 1004. The order in which the certificates are pasted is not significant.

Example:

```
RP/0/RP0/CPU0:ios# crypto ca import myca certificate
```

Step 8 Display information about your certificate and the CA certificate.

Example:

```
RP/0/RP0/CPU0:ios# show crypto ca certificates
```

The following example shows how to configure CA interoperability. Comments are included within the configuration to explain various commands.

```
RP/0/RP0/CPU0:ios# configure
RP/0/RP0/CPU0:ios(config)# hostname mydevice
RP/0/RP0/CPU0:ios(config)# domain name mydomain.com
RP/0/RP0/CPU0:ios(config)# end

Uncommitted changes found, commit them? [yes]:yes

RP/0/RP0/CPU0:ios(config)# crypto key generate rsa mykey

The name for the keys will be:mykey
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keypair
Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus [1024]:
Generating RSA keys ...
Done w/ crypto generate keypair
[OK]

! The following commands declare a CA and configure a trusted point.

RP/0/RP0/CPU0:ios# configure
RP/0/RP0/CPU0:ios(config)# crypto ca trustpoint myca
RP/0/RP0/CPU0:ios(config)# enrollment url http://xyz-ultra5
RP/0/RP0/CPU0:ios(config)# enrollment retry count 25
RP/0/RP0/CPU0:ios(config)# enrollment retry period 2
RP/0/RP0/CPU0:ios(config)# rsakeypair mykey
RP/0/RP0/CPU0:ios(config)# end

Uncommitted changes found, commit them? [yes]:yes

! The following command authenticates the CA to your device.

RP/0/RP0/CPU0:ios(config)# crypto ca authenticate myca

! The following command requests certificates for all of your RSA key pairs.

RP/0/RP0/CPU0:ios(config)# crypto ca enroll myca

! The following command displays information about your certificate and the CA certificate.

RP/0/RP0/CPU0:ios# show crypto ca certificates
```

Cisco NCS 1004 is successfully configured to use the specified trustpoint CA and manual certificate enrollment via cut-and-paste. Certificates are authenticated, imported, and validated.

Public key pair capabilities in XR Config mode

A public key pair capability in XR Config mode is a security mechanism on Cisco NCS 1004 that

- supports the generation of crypto key pairs in both FIPS (Federal Information Processing Standard) and non-FIPS modes,
- enables selection of multiple key types and key sizes, and
- controls overwriting, deleting, and viewing keys based on XR Config and XR EXEC mode distinctions.

Supported key types for non-FIPS and FIPS modes

The following table lists the key types and key sizes that are supported in non-FIPS and FIPS modes.

Table 1: Supported Key Types for non-FIPS and FIPS modes

Key Types	Non-FIPS mode	FIPS mode
RSA	Supported for all key sizes from 512 to 4096	Supported for key sizes 2048, 3072, and 4096
DSA	Supported for key sizes 512, 768, and 1024	Supported for key size 2048
ECDSA	Supported for key sizes nistp256, nistp384, and nistp512	Supported for key sizes nistp256, nistp384, and nistp512
ED25519	Supported	Not supported

Guidelines and restrictions

When you generate crypto key pairs in XR Config mode, the following guidelines and restrictions apply:

- This feature does not support the generation of **system-root-key** and **system-enroll-key**.
- The key pairs that are generated in XR Config mode overwrite any previously generated key pairs in XR EXEC mode.
- Cisco NCS 1004 does not support overwriting key pairs that are generated in XR Config mode from XR EXEC mode.
- When you run the **no** form of the **crypto key generate** command in XR Config mode, only the keys that are generated in XR Config mode are deleted.
- Cisco NCS 1004 does not support deleting key pairs that are generated in XR Config mode from XR EXEC mode.
- When you run the **crypto key generate** command in XR EXEC mode, the keys that are generated in XR Config mode are not overwritten or deleted.
- The **show crypto key mypubkey** command displays the keys that are generated in XR EXEC mode first, followed by the keys that are generated in XR Config mode.

Configuration exampl

This example shows the creation of key pairs in XR Config mode:

```
RP/0/RP0/CPU0:ios# conf t
RP/0/RP0/CPU0:ios(config)# crypto key generate dsa 512
RP/0/RP0/CPU0:ios(config)# crypto key generate rsa user1 general-keys 2048
RP/0/RP0/CPU0:ios(config)# crypto key generate rsa user2 usage-keys 2048
RP/0/RP0/CPU0:ios(config)# crypto key generate rsa 2048
RP/0/RP0/CPU0:ios(config)# crypto key generate ecdsa nistp256
RP/0/RP0/CPU0:ios(config)# crypto key generate ecdsa nistp384
RP/0/RP0/CPU0:ios(config)# crypto key generate ecdsa nistp521
RP/0/RP0/CPU0:ios(config)# crypto key generate ed25519
RP/0/RP0/CPU0:ios(config)# commit
```

Use the **no** form of the command in XR Config mode to delete any of the key pairs.

System logs and error messages

Cisco NCS 1004 generates the following system logs on successful creation of key pairs:

```
cepki[287]: %SECURITY-CEPKI-6-KEY_INFO : crypto key DSA generated, label:the_default,
modBits:1024
cepki[287]: %SECURITY-CEPKI-6-KEY_INFO : crypto key ECDSA_NISTP256 generated,
label:the_default, modBits:256
```

Cisco NCS 1004 generates the following system logs on deletion of key pairs:

```
cepki[287]: %SECURITY-CEPKI-6-KEY_INFO : crypto key RSA zeroized, label:user1
cepki[287]: %SECURITY-CEPKI-6-KEY_INFO : crypto key DSA zeroized, label:the_default
```

Cisco NCS 1004 generates the following error messages if you try to overwrite key pairs that are generated in XR Config mode from XR EXEC mode:

```
RP/0/RP0/CPU0:ios# conf t
RP/0/RP0/CPU0:ios(config)# crypto key generate ed25519
RP/0/RP0/CPU0:ios(config)# commit
RP/0/RP0/CPU0:ios(config)# crypto key generate ed25519
Cannot execute the command : Operation not permitted
ce_cmd[68727]: %SECURITY-CEPKI-6-ERR_2 : Cannot      execute the command :   Operation not
 permitted
ce_cmd[68736]: %SECURITY-CEPKI-6-ERR : Key is added as part of config mode, key deletion
is not allowed , delete key from config mode
```

Cisco NCS 1004 generates the following error messages if you try to delete key pairs that are generated in XR Config mode from XR EXEC mode:

```
RP/0/RP0/CPU0:ios# conf t
RP/0/RP0/CPU0:ios(config)# crypto key generate ed25519
RP/0/RP0/CPU0:ios(config)# commit
RP/0/RP0/CPU0:ios(config)# crypto key zeroize ed25519
Cannot execute the command : Operation not permitted
ce_cmd[68736]: %SECURITY-CEPKI-6-ERR_2 : Cannot execute the command :   Operation not
permitted
```

Viewing the generated key pairs

You can view the key pairs that are generated in XR Config mode, listed under **Public keys from config sysdb** in the following command output:

```
RP/0/RP0/CPU0:ios# show crypto key mypubkey ecdsa
Key label: the_default
Type     : ECDSA General Curve Nistp256
Degree   : 256
Created  : 11:49:22 IST Wed Apr 21 2021
Data     :
 04D6D132 2253ABD0 81449E3F 9D5CEA3A 1107950A 829E9090 8960FBD5 ABA039B7
 24A4E217 7EA47475 91C60AC7 013DBC2E EA8434D9 0BD5B0FC 694913AE 0098A4F5
 77

Key label: the_default
Type     : ECDSA General Curve Nistp521
Degree   : 521
Created  : 22:44:22 IST Thu Mar 18 2021
Data     :
 04017798 4369F493 8D0E57D1 1975FC46 CDC03A78 03A9F90E B38CA504 17DB9A64
 D1DEA6A6 D23E7E20 4D8D4D31 C7878BDB BF5EEE40 1978A889 70C5D703 BB033B77
 0FFD9201 366A9AC8 35E69BB3 97FF4E91 6B498510 39425971 C5E43858 83286088
 A6A7BF92 0EA2B416 BD4E81CE DCEB65F1 15CC75B5 91204E89 3339A168 2382CAB6
```

```
 40170131 8F

-----------------------------
 Public keys from config sysdb:
 -----------------------------
 Key label: the_default
Type     : ECDSA General Curve Nistp384
Degree   : 384
Created  : 11:51:52 IST Wed Apr 21 2021
Data     :
 045F7C14 1A88C27E 9CED3FF1 7FEDFA03 B49575FA 7AD88370 BC9C7D7F F99C8917
 33620916 758BDEFC 7187E33A 2D3CCD33 14FF3267 9855A5E9 E3BD166C CE838462
 40742231 6198EE12 3E189F42 22A8149A 8E7B186D 88E728D4 7F47D565 53441061
 79
```

How to implement CA interoperability

This process enables Cisco NCS 1004 devices to obtain digital certificates from a certification authority (CA) by ensuring device identity, key generation, and trustpoint configuration before requesting certificates.

Summary

The key components involved in the process are:

- Network administrator: Configures the device identity, generates RSA key pairs, and declares the trustpoint that points to the CA.
- Cisco NCS 1004: Stores the keys and trustpoint configuration, and uses them to interact with the CA.
- Certification authority (CA): Issues the digital certificate that the device uses to establish trust with peers.

The process configures the prerequisites that the device needs before it can authenticate the CA and request certificates.

Workflow

The process involves the following stages:

1. Configure the hostname and IP domain name of Cisco NCS 1004 so that the device can build the fully qualified domain name (FQDN) that is used by IPSec keys and certificates.

   For details, see Configure hostname and IP domain name.

2. Generate an RSA key pair on Cisco NCS 1004. The device uses RSA keys to sign and encrypt IKE key management messages and to obtain a certificate from the CA.

   For details, see Generate RSA key pair.

3. Import the public key into Cisco NCS 1004 so that the device can authenticate the user.

   For details, see Import public key to Cisco NCS 1004.

4. Declare the CA and configure the trusted point so that Cisco NCS 1004 can verify certificates issued to peers.

   For details, see Declare certification authority and configure trusted point.

Result

Cisco NCS 1004 is ready to authenticate the CA and request its own certificates.

What's next

After you complete this process, authenticate the CA on Cisco NCS 1004. For details, see Authenticate certification authority.

Configure the hostname and IP domain name on Cisco NCS 1004

Use this task to configure the hostname and IP domain name of Cisco NCS 1004.

You must configure the hostname and IP domain name of Cisco NCS 1004 if they are not already configured. The hostname and IP domain name are required because Cisco NCS 1004 assigns a fully qualified domain name (FQDN) to the keys and certificates that are used by IPSec, and the FQDN is based on the hostname and IP domain name that you assign to the Cisco NCS 1004 device. For example, a certificate that is named *ncs1k.example.com* is based on the Cisco NCS 1004 hostname of *ncs1k* and a device IP domain name of *example.com*.

Procedure

Step 1 Enter the XR Config mode.

Example:

```
RP/0/RP0/CPU0:ios#configure
```

Step 2 Configure the hostname of the Cisco NCS 1004 device.

Example:

```
RP/0/RP0/CPU0:ios(config)# hostname myhost
```

Step 3 Configure the IP domain name of Cisco NCS 1004.

Example:

```
RP/0/RP0/CPU0:ios(config)# domain name mydomain.com
```

Step 4 Save the configuration changes by using the **commit** or **end** command.

Example:

```
RP/0/RP0/CPU0:ios(config)# commit
RP/0/RP0/CPU0:ios(config)# end
```

The hostname and IP domain name are configured, enabling Cisco NCS 1004 to generate the FQDN required for IPSec keys and certificates.

Generate an RSA key pair on Cisco NCS 1004

Enable secure key management and certificate enrollment for Cisco NCS 1004 devices by generating RSA key pairs.

RSA key pairs are required to sign and encrypt IKE key management messages and to obtain certificates from a certification authority. Cisco NCS 1004 automatically generates RSA keys at boot, but you should manually generate keys if none are present. Details for both RSA and DSA keys appear in the running configuration.

Note

- RSA keys are auto-generated when Cisco NCS 1004 boots. Generate the key pair only if the RSA key pair is missing on the device.
- The details of RSA and DSA keys are displayed in the running configuration.

Procedure

Step 1 Generate the RSA key pairs using the command **crypto key generate rsa** [**usage-keys** | **general-keys**] [*keypair-label*]

- Use the **usage-keys** keyword to specify special usage keys, or use the **general-keys** keyword to specify general-purpose RSA keys.
- The *keypair-label* argument is the RSA key pair label that names the RSA key pairs.
- You can also configure this command from XR Config mode. For details about generating key pairs in XR Config mode, see Public key pair generation in XR Config mode.

To delete the RSA keys, use the **no** form of the command: **no crypto key generate rsa**.

Example:

```
RP/0/RP0/CPU0:ios# crypto key generate rsa general-keys
```

Step 2 (Optional) Delete all RSA keys from Cisco NCS 1004.

crypto key zeroize rsa [*keypair-label*]

- Run the **crypto key zeroize** command only in EXEC mode.
- Delete all RSA keys from Cisco NCS 1004 if you suspect that the keys are compromised and must no longer be used.
- To remove a specific RSA key pair, use the *keypair-label* argument.
- You can also delete key pairs with the **no** form of the command from XR Config mode. For details about deleting key pairs in XR Config mode, see Public key pair generation in XR Config mode.

Example:

```
RP/0/RP0/CPU0:ios# crypto key zeroize rsa key1
```

Step 3 (Optional) Display the RSA public keys for Cisco NCS 1004.

show crypto key mypubkey rsa

The **show running-config** command also displays the RSA keys. The keys in the following example are in OpenSSL format.

Note
Only those keys that are generated in XR Config mode are visible in the running configuration.

Example:

```
RP/0/RP0/CPU0:ios# show crypto key mypubkey rsa
Fri Mar 27 14:00:20.954 IST
Key label: system-root-key
Type : RSA General purpose
Size : 2048
Created : 01:13:10 IST Thu Feb 06 2020
Data :
 30820122 300D0609 2A864886 F70D0101 01050003 82010F00 3082010A 02820101
 00A93DE0 1E485EE3 0E7F0964 C48361D1 B6014BE7 A303D8D6 F7790E92 88E69C4B
 B97B7A9C D1B277E3 1569093C 82BD3258 7F67FB49 94860ECD 34498F1F 59B45757
 F32C8E8F 7CEE23EC C36A43D1 9F85C0D9 B96A14DD DD3BBD4C A1FB0888 EED210A7
 39D9A403 7ACE0F6E 39107226 CA621AD8 6E8102CA 9761B86F D33F2871 9DD16559
 AFCB4729 EFCEDBAF 83DF76E4 9A439844 EE3B1180 4022F575 99E11A2C E25BB23D
 9DD74C81 4E5C1345 D9E3CC79 1B98B1AA 6C06F004 22B901EC 36C099FE 10DE2622
 EB7CE618 9A555769 12D94C90 D9BEE5EA A664E7F6 4DF8D8D4 FE7EAB07 1EF4FEAB
 22D9E55F 62BA66A0 72153CEC 81F2639F B5F2B5C5 25E10364 19387C6B E8DB8990
 11020301 0001
Key label: system-enroll-key
Type : RSA General purpose
Size : 2048
Created : 01:13:16 IST Thu Feb 06 2020
Data :
 30820122 300D0609 2A864886 F70D0101 01050003 82010F00 3082010A 02820101
 009DBC14 C83604E4 EB3D3CF8 5BA7FDDB 80F7E85B 427332D8 BBF80148 F0A9C281
 49F87D5C 0CEBA532 EBE797C5 7F174C69 0735D13A 493670CB 63B04A12 4BCA7134
 EE0031E9 047CAA1E 802030C5 6071E8C2 F8ECE002 CC3B54E7 5FD24E5C 61B7B7B0
 68FA2EFA 0B83799F 77AE4621 435D9DFF 1D713108 37B614D3 255020F9 09CD32E8
 82B07CD7 01A53896 6DD92B5D 5119597C 98D394E9 DBD1ABAF 6DE949FE 4A8BF1E7
 851EB3F4 60B1114A 1456723E 063E50C4 2D410906 BDB7590B F1D58480 F3FA911A
 6C9CD02A 58E68D04 E94C098F 0F0E81DB 76B40C55 64603499 2AC0547A D652412A
 BCBBF69F 76B351EE 9B2DF79D E490C0F6 92D1BB97 B905F33B FAB53C20 DDE2BB22
 C7020301 0001
```

RSA key pairs are generated on Cisco NCS 1004, enabling secure communication and certificate management. You can delete or display keys as needed for maintenance or troubleshooting.

Import a public key to Cisco NCS 1004

Import a public key to Cisco NCS 1004 to enable user authentication prior to digital certificate requests.a public key to Cisco NCS 1004.

Use this task when you need Cisco NCS 1004 to authenticate a user with a public key before requesting a certificate from the certification authority.

Procedure

Step 1 Import the RSA public key using the command **crypto key import authentication rsa** [**usage-keys** | **general-keys**] [*keypair-label*].

- Use the **usage-keys** keyword to specify special usage keys, or use the **general-keys** keyword to specify general-purpose RSA keys.
- The *keypair-label* argument is the RSA key pair label that names the RSA key pairs.

Example:

```
RP/0/RP0/CPU0:ios# crypto key import authentication rsa general-keys
```

Step 2 (Optional) Display the RSA public keys for Cisco NCS 1004 using the **show crypto key mypubkey rsa** command.

The **show running-config** command also displays the RSA keys. The keys in the following example are in OpenSSL format.

Note
Only those keys that are generated in XR Config mode are visible in the running configuration.

Example:

```
RP/0/RP0/CPU0:ios# show crypto key mypubkey rsa
Fri Mar 27 14:00:20.954 IST
Key label: system-root-key
Type : RSA General purpose
Size : 2048
Created : 01:13:10 IST Thu Feb 06 2020
Data :
 30820122 300D0609 2A864886 F70D0101 01050003 82010F00 3082010A 02820101
 00A93DE0 1E485EE3 0E7F0964 C48361D1 B6014BE7 A303D8D6 F7790E92 88E69C4B
 B97B7A9C D1B277E3 1569093C 82BD3258 7F67FB49 94860ECD 34498F1F 59B45757
 F32C8E8F 7CEE23EC C36A43D1 9F85C0D9 B96A14DD DD3BBD4C A1FB0888 EED210A7
 39D9A403 7ACE0F6E 39107226 CA621AD8 6E8102CA 9761B86F D33F2871 9DD16559
 AFCB4729 EFCEDBAF 83DF76E4 9A439844 EE3B1180 4022F575 99E11A2C E25BB23D
 9DD74C81 4E5C1345 D9E3CC79 1B98B1AA 6C06F004 22B901EC 36C099FE 10DE2622
 EB7CE618 9A555769 12D94C90 D9BEE5EA A664E7F6 4DF8D8D4 FE7EAB07 1EF4FEAB
 22D9E55F 62BA66A0 72153CEC 81F2639F B5F2B5C5 25E10364 19387C6B E8DB8990
 11020301 0001
Key label: system-enroll-key
Type : RSA General purpose
Size : 2048
Created : 01:13:16 IST Thu Feb 06 2020
Data :
 30820122 300D0609 2A864886 F70D0101 01050003 82010F00 3082010A 02820101
 009DBC14 C83604E4 EB3D3CF8 5BA7FDDB 80F7E85B 427332D8 BBF80148 F0A9C281
 49F87D5C 0CEBA532 EBE797C5 7F174C69 0735D13A 493670CB 63B04A12 4BCA7134
 EE0031E9 047CAA1E 802030C5 6071E8C2 F8ECE002 CC3B54E7 5FD24E5C 61B7B7B0
 68FA2EFA 0B83799F 77AE4621 435D9DFF 1D713108 37B614D3 255020F9 09CD32E8
 82B07CD7 01A53896 6DD92B5D 5119597C 98D394E9 DBD1ABAF 6DE949FE 4A8BF1E7
 851EB3F4 60B1114A 1456723E 063E50C4 2D410906 BDB7590B F1D58480 F3FA911A
 6C9CD02A 58E68D04 E94C098F 0F0E81DB 76B40C55 64603499 2AC0547A D652412A
 BCBBF69F 76B351EE 9B2DF79D E490C0F6 92D1BB97 B905F33B FAB53C20 DDE2BB22
 C7020301 0001
```

The RSA public key is imported to Cisco NCS 1004, enabling the device to authenticate the user before requesting a digital certificate.

Declare a certification authority and configure a trusted point

Set up the certificate infrastructure by declaring a certification authority and configuring a trusted point on Cisco NCS 1004, allowing secure verification of certificates issued to peers.

Configuring a trusted point that references the CA enables Cisco NCS 1004 to request and verify certificates issued to peer devices for secure communication.

Procedure

Step 1 Enter the XR Config mode.

Example:

```
RP/0/RP0/CPU0:ios# configure
```

Step 2 Declare the CA, configure a trusted point with a selected name so that Cisco NCS 1004 can verify the certificates that are issued to peers, and enter trustpoint configuration mode.

Example:

```
RP/0/RP0/CPU0:ios(config)# crypto ca trustpoint myca
```

Step 3 Specify the URL of the CA. The URL must include any non-standard cgi-bin script location.

enrollment url *CA-URL*

Example:

```
RP/0/RP0/CPU0:ios(config-trustp)# enrollment url http://ca.domain.com/certsrv/mscep/mscep.dll
```

Step 4 (Optional) Specify the location of the LDAP server if your CA system supports the LDAP protocol.

query url *LDAP-URL*

Example:

```
RP/0/RP0/CPU0:ios(config-trustp)# query url ldap://my-ldap.domain.com
```

Step 5 (Optional) Specify a retry period usnig the command **enrollment retry period** *minutes*.

- After Cisco NCS 1004 requests a certificate, the device waits to receive a certificate from the CA. If the device does not receive a certificate within the retry period, it sends another certificate request.
- The range is from 1 to 60 minutes. The default is 1 minute.

Example:

```
RP/0/RP0/CPU0:ios(config-trustp)# enrollment retry period 2
```

Step 6 (Optional) Specify how many times Cisco NCS 1004 continues to send unsuccessful certificate requests before giving up. The range is from 1 to 100.

Example:

```
RP/0/RP0/CPU0:ios(config-trustp)# enrollment retry count 10
```

Step 7 (Optional) Use the **rsakeypair** *keypair-label* command to specify a named RSA key pair that is generated by using the **crypto key generate rsa** command for this trustpoint. If you do not set this key pair, the trustpoint uses the default RSA key in the current configuration.

Example:

```
RP/0/RP0/CPU0:ios(config-trustp)# rsakeypair mykey
```

Step 8 Save the configuration changes by using the **commit** or **end** command.

Authenticate certification authority

Use this task to authenticate the CA to your Cisco NCS 1004 device.

The Cisco NCS 1004 device must authenticate the CA by obtaining the self-signed certificate of the CA, which contains the public key of the CA. Because the certificate of the CA is self-signed (the CA signs its own certificate), manually authenticate the public key of the CA by contacting the CA administrator to compare the fingerprint of the CA certificate.

Procedure

Step 1 Authenticate the CA to your Cisco NCS 1004 device by obtaining a CA certificate that contains the public key for the CA.

crypto ca authenticate *ca-name*

Example:

```
RP/0/RP0/CPU0:ios# crypto ca authenticate myca
```

Step 2 (Optional) Display information about the CA certificate.

show crypto ca certificates

Example:

```
RP/0/RP0/CPU0:ios# show crypto ca certificates
```

Multi-tier certificate authority for trustpoint authentication

Multi-tier certificate authority for trustpoint authentication is a security capability on Cisco NCS 1004 that lets a complete CA hierarchy, from the Root CA to the subordinate CA that issues the certificate, be imported as part of a single terminal-based authentication request.

- Supports a maximum of 8 tiers, that is, a chain of CAs with one Root CA and seven subordinate CAs, for trustpoint authentication.
- Lets you import the complete CA hierarchy in a single authentication request.
- Provides flexibility and security in network topologies that use a multi-tier CA hierarchy for enrollment.

Feature history for multi-tier certificate authority for trustpoint authentication

Table 2: Feature History Table

Feature Name	Release Information	Description
Multi-Tier Certificate Authority for Trustpoint Authentication	Cisco IOS XR Release 7.10.1	Apart from the Root certificate authority (CA), you can now use a subordinate CA to issue certificates and authenticate your network devices. This feature is beneficial when you have an existing CA hierarchy where it is not the Root CA but the subordinate CA that issues the leaf or certificates. In earlier releases, you could associate only a single CA, not a multi-tier CA, to a trustpoint. And, you could use only the Root CA certificate to enroll the certificates. This feature modifies the show crypto ca certificates command to display the **Trusted Certificate Chain** field.

Need for multi-tier certificate authority for trustpoint authentication

During terminal-based enrollment of a CA trustpoint, Cisco IOS XR network devices accepted only Root CA certificates. Some network topologies use a multi-tier CA hierarchy for enrollment because it provides more flexibility and security. From Cisco IOS XR Release 7.10.1 and later, as part of terminal-based authentication, you can import a complete CA hierarchy (from the Root CA to the subordinate CA that issues the certificate) as part of a single authentication request. With this feature, you can provide a certificate chain that includes the Root CA and intermediate subordinate CAs as part of the terminal-based enrollment process. This feature is useful if you have an existing multi-tier CA hierarchy where the Root CA does not issue any certificates directly, and if you want only subordinate CAs to issue certificates to authenticate your network devices.

Using multi-tier CA for trustpoint authentication

The **crypto ca authenticate** command is used to authenticate a trustpoint with a multi-tier CA hierarchy. You must use only Privacy Enhanced Mail (PEM)-encoded certificates for trustpoint authentication that uses multi-tier CAs. The enrollment process is the same as enrollment that uses a single-tier CA, except that Cisco NCS 1004 displays a console message that prompts you to use only PEM-encoded certificates.

Prerequisite for using multi-tier CA for trustpoint authentication

You must generate a key pair, import a public key, and configure a trustpoint on Cisco NCS 1004 as detailed in the previous sections.

Configuration example for multi-tier CA trustpoint authentication

The following example shows the trustpoint authentication for a multi-tier CA on Cisco NCS 1004.

```
RP/0/RP0/CPU0:ios#crypto ca authenticate test-ca
Mon Feb  6 08:17:48.943 UTC


Enter the base 64/PEM encoded certificate/certificates.
Please note: for multiple certificates use only PEM
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
MIIF5TCCA82gAwIBAgICEAEwDQYJKoZIhvcNAQELBQAwXTELMAkGA1UEBhMCSU4x
CzAJBgNVBAgMAktBMQwwCgYDVQQHDANCR0wxDTALBgNVBAoMBENTQ08xDTALBgNV
.
.
.
/4UzeeX6ll0gGJVbDwGeIZTH00artqxHquKQ2P7eXQ1pg0PRNRqWN90SvT5yE33N
eHgbtvdHg1K6K6IAj/NGnd7xUrA1TQ4bdmouCNkgbXM/G9DwgkOOvZ8KYRP9JW57
LYIv2ZcRS2vdnZRD9JPGVig2EgcfVPtj+Q==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIF9TCCA92gAwIBAgIUD6AGesleqedhorkrJ9HWjz1RQzswDQYJKoZIhvcNAQEL
BQAwXTELMAkGA1UEBhMCSU4xCzAJBgNVBAgMAktBMQwwCgYDVQQHDANCR0wxDTAL
.
.
.
+6rMWd6BmfSy2PT3Qz5AjO2+3N1dd67qRRrX7skklkX4JXY42n5/l9PQtSp0wTBh
uy5yUAagynu0z07GczE7E9V+tJHRmNTbnd8pxLk4lTwqtiCIXwQLZA75SkwCS5wh
fn7OrV7uFjMaggNkvj0kSSOkWxqJ+j/KqMAA2zQMUV+qdvT6i+ZV44U=
-----END CERTIFICATE-----
Serial Number  : 10:01
  Subject:
        CN=SUB_CA_CERT,OU=SPBU,O=CSCO,L=BGL,ST=KA,C=IN
  Issued By      :
        CN=TWO-LEVEL-CA,OU=SPBU,O=CSCO,L=BGL,ST=KA,C=IN
  Validity Start : 12:31:40 UTC Sun Jun 14 2020
  Validity End   : 12:31:40 UTC Wed Jun 12 2030

  CRL Distribution Point
        http://10.105.236.78/crl_akshath_two_level_ca/crl.der
  SHA1 Fingerprint:
         D8E0C11ECED96F67FDBC800DB6A126676A76BD62
  Serial Number  : 0F:A0:06:7A:C9:5E:A9:E7:61:A2:B9:2B:27:D1:D6:8F:3D:51:43:3B
  Subject:
        CN=TWO-LEVEL-CA,OU=SPBU,O=CSCO,L=BGL,ST=KA,C=IN
  Issued By      :
        CN=TWO-LEVEL-CA,OU=SPBU,O=CSCO,L=BGL,ST=KA,C=IN
  Validity Start : 13:12:32 UTC Sun Jun 07 2020
  Validity End   : 13:12:32 UTC Sat Jun 02 2040

  CRL Distribution Point
        http://10.105.236.78/crl_akshath_two_level_ca/crl.der
  SHA1 Fingerprint:
         08E71248FB7578614442E713AC87C461D173952F

CA Certificate validated using issuer certificate.
RP/0/RP0/CPU0:ios#
```

Verification of multi-tier CA trustpoint authentication

Use the **show crypto ca certificates test-ca** command to view the CA certificate chain. The command output displays the **Trusted Certificate Chain** field if one or more subordinate CAs are involved in the hierarchy.

```
RP/0/RP0/CPU0:ios#show crypto ca certificates test-ca
Mon Feb  6 09:03:53.019 UTC

Trustpoint       : test-ca
==================================================
CA certificate
  Serial Number  : 10:01
  Subject:
            CN=SUB_CA_CERT,OU=SPBU,O=CSCO,L=BGL,ST=KA,C=IN
  Issued By      :
            CN=TWO-LEVEL-CA,OU=SPBU,O=CSCO,L=BGL,ST=KA,C=IN
  Validity Start : 12:31:40 UTC Sun Jun 14 2020
  Validity End   : 12:31:40 UTC Wed Jun 12 2030

  CRL Distribution Point
            http://10.105.236.78/crl_akshath_two_level_ca/crl.der
  SHA1 Fingerprint:
             D8E0C11ECED96F67FDBC800DB6A126676A76BD62
Trusted Certificate Chain
  Serial Number  : 0F:A0:06:7A:C9:5E:A9:E7:61:A2:B9:2B:27:D1:D6:8F:3D:51:43:3B
  Subject:
            CN=TWO-LEVEL-CA,OU=SPBU,O=CSCO,L=BGL,ST=KA,C=IN
  Issued By      :
            CN=TWO-LEVEL-CA,OU=SPBU,O=CSCO,L=BGL,ST=KA,C=IN
  Validity Start : 13:12:32 UTC Sun Jun 07 2020
  Validity End   : 13:12:32 UTC Sat Jun 02 2040

  CRL Distribution Point
            http://10.105.236.78/crl_akshath_two_level_ca/crl.der
  SHA1 Fingerprint:
             08E71248FB7578614442E713AC87C461D173952F
certificate
  Key usage      : General Purpose
  Status         : Available
  Serial Number  : 28:E5
  Subject:
            CN=test
  Issued By      :
            CN=SUB_CA_CERT,OU=SPBU,O=CSCO,L=BGL,ST=KA,C=IN
  Validity Start : 08:49:54 UTC Mon Feb 06 2023
  Validity End   : 08:49:54 UTC Wed Mar 08 2023
  SHA1 Fingerprint:
             6C8644FA67D9CEBC7C5665C35838265F578835AB
Associated Trustpoint: test-ca
```