



Deploying RSA Keys Within a PKI

Last Updated: July 18, 2012

This module explains how to set up and deploy Rivest, Shamir, and Adelman (RSA) keys within a public key infrastructure (PKI). An RSA key pair (a public and a private key) is required before you can obtain a certificate for your router; that is, the end host must generate a pair of RSA keys and exchange the public key with the certification authority (CA) to obtain a certificate and enroll in a PKI

- [Finding Feature Information, page 1](#)
- [Prerequisites for Configuring RSA Keys for a PKI, page 1](#)
- [Information About RSA Keys Configuration, page 2](#)
- [How to Set Up and Deploy RSA Keys Within a PKI, page 4](#)
- [Configuration Examples for RSA Key Pair Deployment, page 13](#)
- [Where to Go Next, page 18](#)
- [Additional References, page 18](#)
- [Feature Information for Deploying RSA Keys Within a PKI, page 19](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring RSA Keys for a PKI

- Before setting up and deploying RSA keys for a PKI, you should be familiar with the module “Cisco IOS XE PKI Overview: Understanding and Planning a PKI.”
- All commands that begin as “crypto ca” have been changed to begin as “crypto pki.” Although the router will still accept crypto ca commands, all output will be read back as crypto pki.



Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About RSA Keys Configuration

- [RSA Keys Overview, page 2](#)
- [Reasons to Store Multiple RSA Keys on a Router, page 3](#)
- [Benefits of Exportable RSA Keys, page 3](#)
- [Passphrase Protection While Importing and Exporting RSA Keys, page 3](#)

RSA Keys Overview

An RSA key pair consists of a public key and a private key. When setting up your PKI, you must include the public key in the certificate enrollment request. After the certificate has been granted, the public key will be included in the certificate so that peers can use it to encrypt data that is sent to the router. The private key is kept on the router and used both to decrypt the data sent by peers and to digitally sign transactions when negotiating with peers.

RSA key pairs contain a key modulus value. The modulus determines the size of the RSA key. The larger the modulus, the more secure the RSA key. However, keys with large modulus values take longer to generate, and encryption and decryption operations take longer with larger keys.

If you want a modulus value between 512 and 1024, enter an integer value that is a multiple of 64. If you want a value higher than 1024, enter 1536 or 2048. If you enter a value greater than 512, key generation may take a minute or longer.



Note

Peer *public* RSA key modulus values up to 4096 bits are automatically supported. The largest private RSA key modulus is 4096 bits. Therefore, the largest RSA private key a router may generate or import is 4096 bits. However, RFC 2409 restricts the private key size to 2048 bits or less for RSA encryption. The recommended modulus value for a CA is 2048 bits; the recommended modulus value for a client is 1024 bits.

- [Usage RSA Keys Versus General-Purpose RSA Keys, page 2](#)

Usage RSA Keys Versus General-Purpose RSA Keys

There are two mutually exclusive types of RSA key pairs--usage keys and general-purpose keys. When you generate RSA key pairs (via the **crypto key generate rsa** command), you will be prompted to select either usage keys or general-purpose keys.

Usage RSA Keys

Usage keys consist of two RSA key pairs--one RSA key pair is generated and used for encryption and one RSA key pair is generated and used for signatures. With usage keys, each key is not unnecessarily exposed. (Without usage keys, one key is used for both authentication methods, increasing the exposure of that key.)

General-Purpose RSA Keys

General-purpose keys consist of only one RSA key pair that used for both encryption and signatures. General-purpose key pairs are used more frequently than usage key pairs.

Reasons to Store Multiple RSA Keys on a Router

Configuring multiple RSA key pairs allows the Cisco IOS software to maintain a different key pair for each CA with which it is dealing or the software can maintain multiple key pairs and certificates with the same CA. As a result, the Cisco IOS software can match policy requirements for each CA without compromising the requirements specified by the other CAs, such as key length, key lifetime, and general-purpose versus usage keys.

Named key pairs (which are specified via the **label** *key-label* option) allow you to have multiple RSA key pairs, enabling the Cisco IOS software to maintain a different key pair for each identity certificate.

Benefits of Exportable RSA Keys



Caution

Exportable RSA keys should be carefully evaluated before use because using exportable RSA keys introduces the risk that these keys might be exposed. Any existing RSA keys are not exportable. New keys are generated as nonexportable by default. It is not possible to convert an existing nonexportable key to an exportable key.

Users can share the private RSA key pair of a router with standby routers, therefore transferring the security credentials between networking devices. The key pair that is shared between two routers will allow one router to immediately and transparently take over the functionality of the other router. If the main router were to fail, the standby router could be dropped into the network to replace the failed router without the need to regenerate keys, reenroll with the CA, or manually redistribute keys.

Exporting and importing an RSA key pair also enables users to place the same RSA key pair on multiple routers so that all management stations using Secure Shell (SSH) can be configured with a single public RSA key.

Exportable RSA Keys in PEM-Formatted Files

Using privacy-enhanced mail (PEM)-formatted files to import or export RSA keys can be helpful for customers who are using secure socket layer (SSL) or secure shell (SSH) applications to manually generate RSA key pairs and import the keys back into their PKI applications. PEM-formatted files allow customers to directly use existing RSA key pairs on their Cisco IOS XE routers instead of generating new keys.

Passphrase Protection While Importing and Exporting RSA Keys

You have to include a passphrase to encrypt the PKCS12 file or the PEM file that will be exported, and when the PKCS12 or PEM file is imported, the same passphrase has to be entered to decrypt it. Encrypting the PKCS12 or PEM file when it is being exported, deleted, or imported protects the file from unauthorized access and use while it is being transported or stored on an external device.

The passphrase can be any phrase that is at least eight characters in length; it can include spaces and punctuation, excluding the question mark (?), which has special meaning to the Cisco IOS parser.

How to Convert an Exportable RSA Key Pair to a Nonexportable RSA Key Pair

Passphrase protection protects the external PKCS12 or PEM file from unauthorized access and use. To prevent an RSA key pair from being exported, it must be labeled “nonexportable.” To convert an exportable RSA key pair into a nonexportable key pair, the key pair must be exported and then reimported without specifying the “exportable” keyword.

How to Set Up and Deploy RSA Keys Within a PKI

- [Generating an RSA Key Pair, page 4](#)
- [Generating and Storing Multiple RSA Key Pairs, page 5](#)
- [Exporting and Importing RSA Keys, page 6](#)
- [Encrypting and Locking Private Keys on a Router, page 10](#)
- [Removing RSA Key Pair Settings, page 12](#)

Generating an RSA Key Pair

Perform this task to manually generate an RSA key pair.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `crypto key generate rsa {general-keys | usage-keys} [label key-label] [modulus modulus-size] [exportable]`
4. `exit`
5. `show crypto key mypubkey rsa`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><code>enable</code></p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p><code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
Step 3	<p><code>crypto key generate rsa {general-keys usage-keys} [label key-label] [modulus modulus-size] [exportable]</code></p> <p>Example:</p> <pre>Router(config)# crypto key generate rsa general- keys modulus 360</pre>	<p>Generates RSA key pairs.</p> <ul style="list-style-type: none"> • If a <i>key-label</i> argument is not specified, the default value, which is the fully qualified domain name (FQDN) of the router, is used.

Command or Action	Purpose
Step 4 <code>exit</code> Example: <code>Router(config)# exit</code>	Exits global configuration mode.
Step 5 <code>show crypto key mypubkey rsa</code> Example: <code>Router# show crypto key mypubkey rsa</code>	(Optional) Displays the RSA public keys of your router. This step allows you to verify that the RSA key pair has been successfully generated.

- [What to Do Next, page 5](#)

What to Do Next

After you have successfully generated an RSA key pair, you can proceed to any of the additional tasks in this module to generate additional RSA key pairs, perform export and import of RSA key pairs, or configure additional security parameters for the RSA key pair (such as encrypting or locking the private key).

Generating and Storing Multiple RSA Key Pairs

Perform this task to configure the router to generate and store multiple RSA key pairs and associate the key pairs with a trustpoint.

A trustpoint (also known as a CA) manages certificate requests and issues certificates to participating network devices. These services provide centralized key management for the participating devices and are explicitly trusted by the receiver to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

You must have already generated an RSA key pair as shown in the task “[Generating an RSA Key Pair, page 4](#).”

SUMMARY STEPS

1. `crypto pki trustpoint name`
2. `rsakeypair key-label [key-size [encryption-key-size]]`
3. `exit`
4. `exit`
5. `show crypto key mypubkey rsa`

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 <code>crypto pki trustpoint <i>name</i></code></p> <p>Example:</p> <pre>Router(config)# crypto pki trustpoint fancy-ca</pre>	Creates a trustpoint and enters ca-trustpoint configuration mode.
<p>Step 2 <code>rsakeypair <i>key-label</i> [<i>key-size</i> [<i>encryption-key-size</i>]]</code></p> <p>Example:</p> <pre>Router(ca-trustpoint)# rsakeypair fancy-keys</pre>	<p>Specifies the key pair that is to be used with the trustpoint.</p> <ul style="list-style-type: none"> Specify the <i>key-size</i> argument for generating the key and specify the <i>encryption-key-size</i> argument to request separate encryption, signature keys, and certificates.
<p>Step 3 <code>exit</code></p> <p>Example:</p> <pre>Router (ca-trustpoint)# exit</pre>	Exits ca-trustpoint configuration mode.
<p>Step 4 <code>exit</code></p> <p>Example:</p> <pre>Router(config)# exit</pre>	Exits global configuration mode.
<p>Step 5 <code>show crypto key mypubkey rsa</code></p> <p>Example:</p> <pre>Router# show crypto key mypubkey rsa</pre>	<p>(Optional) Displays the RSA public keys of your router.</p> <p>This step allows you to verify that the RSA key pair has been successfully generated.</p>

Exporting and Importing RSA Keys

This section contains the following tasks that can be used for exporting and importing RSA keys. Whether you are using PKCS12 files or PEM files, exportable RSA keys allow you to use existing RSA keys on Cisco IOS routers instead of having to generate new RSA keys if the main router were to fail.

- [Exporting and Importing RSA Keys in PKCS12 Files, page 6](#)
- [Exporting and Importing RSA Keys in PEM-Formatted Files, page 8](#)

Exporting and Importing RSA Keys in PKCS12 Files

Exporting and importing RSA key pairs enables users to transfer security credentials between devices. The key pair that is shared between two devices allows one device to immediately and transparently take over the functionality of the other router.

You must generate an RSA key pair and mark it “exportable” as specified in the task “[Generating an RSA Key Pair](#), page 4.”



Note

- When you import a PKCS12 file that was generated by a third-party application, the PKCS12 file must include a CA certificate.
- If you have to reexport an RSA key pair after you have already exported the key pair and imported the key pair to a target router, you must specify the **exportable** keyword when you are importing the RSA key pair.
- The largest RSA key a router may import is 2048-bits.

>

SUMMARY STEPS

1. **crypto pki trustpoint** *name*
2. **rsa****keypair** *key-label* [*key-size* [*encryption-key-size*]]
3. **exit**
4. **crypto pki export** *trustpointname* **pkcs12** *destination-url* *passphrase*
5. **crypto pki import** *trustpointname* **pkcs12** *source-url* *passphrase*
6. **exit**
7. **show crypto key mypubkey rsa**

DETAILED STEPS

Command or Action	Purpose
Step 1 crypto pki trustpoint <i>name</i> Example: Router(config)# crypto pki trustpoint my-ca	Creates the trustpoint name that is to be associated with the RSA key pair and enters ca-trustpoint configuration mode.
Step 2 rsa keypair <i>key-label</i> [<i>key-size</i> [<i>encryption-key-size</i>]] Example: Router(ca-trustpoint)# rsa keypair my-keys	Specifies the key pair that is to be used with the trustpoint.
Step 3 exit Example: Router(ca-trustpoint)# exit	Exits ca-trustpoint configuration mode.

Command or Action	Purpose
<p>Step 4 <code>crypto pki export trustpointname pkcs12 destination-url passphrase</code></p> <p>Example:</p> <pre>Router(config)# crypto pki export my-ca pkcs12 tftp://tftpserver/my-keys PASSWORD</pre>	<p>Exports the RSA keys via the trustpoint name.</p> <p>Note You can export the trustpoint using any of the following file system types: flash, FTP, null, NVRAM, remote file copying (RCP), SCP, system, TFTP, Webflash, Xmodem, or Ymodem.</p>
<p>Step 5 <code>crypto pki import trustpointname pkcs12 source-url passphrase</code></p> <p>Example:</p> <pre>Router(config)# crypto pki import my-ca pkcs12 tftp://tftpserver/my-keys PASSWORD</pre>	<p>Imports the RSA keys to the target router.</p>
<p>Step 6 <code>exit</code></p> <p>Example:</p> <pre>Router(config)# exit</pre>	<p>Exits global configuration mode.</p>
<p>Step 7 <code>show crypto key mypubkey rsa</code></p> <p>Example:</p> <pre>Router# show crypto key mypubkey rsa</pre>	<p>(Optional) Displays the RSA public keys of your router.</p>

Exporting and Importing RSA Keys in PEM-Formatted Files

Perform this task to export or import RSA key pairs in PEM files.

You must generate an RSA key pair and mark it “exportable” as specified in the task “[Generating an RSA Key Pair, page 4.](#)”



Note

- The largest RSA key a router may import is 2048 bits.

>

SUMMARY STEPS

1. **crypto key generate rsa** {usage-keys | general-keys} label *key-label* [exportable
2. **crypto key export rsa** *key-label* pem {terminal | url *url*} {3des| des} *passphrase*
3. **crypto key import rsa** *key-label* pem [usage-keys] {terminal | url *url*} [exportable] *passphrase*
4. **exit**
5. **show crypto key mypubkey rsa**

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 crypto key generate rsa {usage-keys general-keys} label <i>key-label</i> [exportable</p> <p>Example:</p> <pre>Router(config)# crypto key generate rsa general-keys label mykey exportable</pre>	<p>Generates RSA key pairs.</p> <p>To use PEM files, the RSA key pair must be labeled exportable.</p>
<p>Step 2 crypto key export rsa <i>key-label</i> pem {terminal url <i>url</i>} {3des des} <i>passphrase</i></p> <p>Example:</p> <pre>Router(config)# crypto key export rsa mycs pem url nvram: 3des PASSWORD</pre>	<p>Exports the generated RSA key pair.</p> <p>Tip Be sure to keep the PEM file safe. For example, you may want to store it on another backup router.</p>
<p>Step 3 crypto key import rsa <i>key-label</i> pem [usage-keys] {terminal url <i>url</i>} [exportable] <i>passphrase</i></p> <p>Example:</p> <pre>Router(config)# crypto key import rsa mycs2 pem url nvram: PASSWORD</pre>	<p>Imports the generated RSA key pair.</p> <p>Note If you do not want the key to be exportable from your CA, import it back to the CA after it has been exported as a nonexportable key pair. Thus, the key cannot be taken off again.</p>
<p>Step 4 exit</p> <p>Example:</p> <pre>Router(config)# exit</pre>	<p>Exits global configuration mode.</p>
<p>Step 5 show crypto key mypubkey rsa</p> <p>Example:</p> <pre>Router# show crypto key mypubkey rsa</pre>	<p>(Optional) Displays the RSA public keys of your router.</p>

Encrypting and Locking Private Keys on a Router

Digital signatures are used to authenticate one device to another device. To use digital signatures, private information (the private key) must be stored on the device that is providing the signature. The stored private information may aid an attacker who steals the hardware device that contains the private key; for example, a thief might be able to use the stolen router to initiate a secure connection to another site by using the RSA private keys stored in the router.



Note

RSA keys are lost during password recovery operations. If you lose your password, the RSA keys will be deleted when you perform the password recovery operation. (This function prevents an attacker from performing password recovery and then using the keys.)

To protect the private RSA key from an attacker, a user can encrypt the private key that is stored in NVRAM via a passphrase. Users can also “lock” the private key, which blocks new connection attempts from a running router and protects the key in the router if the router is stolen by an attempted attacker.

Perform this task to encrypt and lock the private key that is saved to NVRAM.

Before encrypting or locking a private key, you should perform the following tasks:

- Generate an RSA key pair as shown in the task “[Generating an RSA Key Pair, page 4.](#)”
- Optionally, you can authenticate and enroll each router with the CA server.



Note

The RSA keys must be unlocked while enrolling the CA. The keys can be locked while authenticating the router with the CA because the private key of the router is not used during authentication.



Note

Interaction with Applications

An encrypted key is not effective after the router boots up until you manually unlock the key (via the **crypto key unlock rsa** command). Depending on which key pairs are encrypted, this functionality may adversely affect applications such as IP security (IPsec), SSH, and SSL; that is, management of the router over a secure channel may not be possible until the necessary key pair is unlocked.

>

SUMMARY STEPS

1. **crypto key encrypt [write] rsa [name key-name] passphrase *passphrase***
2. **exit**
3. **show crypto key mypubkey rsa**
4. **crypto key lock rsa name *key-name*] passphrase *passphrase***
5. **show crypto key mypubkey rsa**
6. **crypto key unlock rsa [name *key-name*] passphrase *passphrase***
7. **configure terminal**
8. **crypto key decrypt [write] rsa [name *key-name*] passphrase *passphrase***

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 <code>crypto key encrypt [write] rsa [name key-name] passphrase <i>passphrase</i></code></p> <p>Example:</p> <pre>Router(config)# crypto key encrypt write rsa name pki.company.com passphrase password</pre>	<p>Encrypts the RSA keys.</p> <p>After this command is issued, the router can continue to use the key; the key remains unlocked.</p> <p>Note If the write keyword is not issued, the configuration must be manually written to NVRAM; otherwise, the encrypted key will be lost next time the router is reloaded.</p>
<p>Step 2 <code>exit</code></p> <p>Example:</p> <pre>Router(config)# exit</pre>	<p>Exits global configuration mode.</p>
<p>Step 3 <code>show crypto key mypubkey rsa</code></p> <p>Example:</p> <pre>Router# show crypto key mypubkey rsa</pre>	<p>(Optional) Shows that the private key is encrypted (protected) and unlocked.</p> <p>Note You can also use this command to verify that applications such as Internet Key Exchange (IKE) and SSH are properly working after the key has been encrypted.</p>
<p>Step 4 <code>crypto key lock rsa name <i>key-name</i>] passphrase <i>passphrase</i></code></p> <p>Example:</p> <pre>Router# crypto key lock rsa name pki.company.com passphrase password</pre>	<p>(Optional) Locks the encrypted private key on a running router.</p> <p>Note After the key is locked, it cannot be used to authenticate the router to a peer device. This behavior disables any IPSec or SSL connections that use the locked key. Any existing IPSec tunnels created on the basis of the locked key will be closed. If all RSA keys are locked, SSH will automatically be disabled.</p>
<p>Step 5 <code>show crypto key mypubkey rsa</code></p> <p>Example:</p> <pre>Router# show crypto key mypubkey rsa</pre>	<p>(Optional) Shows that the private key is protected and locked.</p> <p>The output will also show failed connection attempts via applications such as IKE, SSH, and SSL.</p>
<p>Step 6 <code>crypto key unlock rsa [name <i>key-name</i>] passphrase <i>passphrase</i></code></p> <p>Example:</p> <pre>Router# crypto key unlock rsa name pki.company.com passphrase password</pre>	<p>(Optional) Unlocks the private key.</p> <p>Note After this command is issued, you can continue to establish IKE tunnels.</p>

Command or Action	Purpose
Step 7 <code>configure terminal</code> Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 8 <code>crypto key decrypt [write] rsa [name <i>key-name</i>] passphrase <i>passphrase</i></code> Example: <pre>Router(config)# crypto key decrypt write rsa name pki.company.com passphrase password</pre>	(Optional) Deletes the encrypted key and leaves only the unencrypted key. Note The write keyword immediately saves the unencrypted key to NVRAM. If the write keyword is not issued, the configuration must be manually written to NVRAM; otherwise, the key will remain encrypted the next time the router is reloaded.

Removing RSA Key Pair Settings

You might want to remove an RSA key pair for one of the following reasons:

- During manual PKI operations and maintenance, old RSA keys can be removed and replaced with new keys.
- An existing CA is replaced and the new CA requires newly generated keys; for example, the required key size might have changed in an organization so that you would have to delete the old 1024-bit keys and generate new 2048-bit keys.

Perform this task to remove all RSA keys or the specified RSA key pair that has been generated by your router.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `crypto key zeroize rsa key-pair-label`
4. `exit`
5. `show crypto key mypubkey rsa`

DETAILED STEPS

Command or Action	Purpose
Step 1 <code>enable</code> Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

Command or Action	Purpose
Step 2 <code>configure terminal</code> Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3 <code>crypto key zeroize rsa key-pair-label</code> Example: <pre>Router(config)# crypto key zeroize rsa fancy-keys</pre>	Deletes RSA key pairs from your router. <ul style="list-style-type: none"> If the <i>key-pair-label</i> argument is not specified, all RSA keys that have been generated by your router will be deleted.
Step 4 <code>exit</code> Example: <pre>Router(config)# exit</pre>	Exits global configuration mode.
Step 5 <code>show crypto key mypubkey rsa</code> Example: <pre>Router# show crypto key mypubkey rsa</pre>	(Optional) Displays the RSA public keys of your router. This step allows you to verify that the RSA key pair has been successfully generated.

Configuration Examples for RSA Key Pair Deployment

Generating and Specifying RSA Keys Example

The following example is a sample trustpoint configuration that shows how to generate and specify the RSA key pair “exampleCAkeys”:

```
crypto key generate rsa general-purpose exampleCAkeys
crypto ca trustpoint exampleCAkeys
  enroll url http://exampleCAkeys/certsrv/mscep/mscep.dll
  rsakeypair exampleCAkeys 1024 1024
```

Exporting and Importing RSA Keys Examples

This section contains the following configuration examples:

Exporting and Importing RSA Keys in PKCS12 Files Example

In the following example, an RSA key pair “mynewkp” is generated on Router A, and a trustpoint name “mynewtp” is created and associated with the RSA key pair. The trustpoint is exported to a TFTP server, so

that it can be imported on Router B. By importing the trustpoint “mynewtp” to Router B, the user has imported the RSA key pair “mynewkp” to Router B.

Router A

```
crypto key generate rsa general label mykeys exportable
! The name for the keys will be:mynewkp
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.
How many bits in the modulus [512]:
% Generating 512 bit RSA keys ...[OK]
!
crypto pki trustpoint mynewtp
  rsakeypair mykeys
  exit
crypto pki export mytp pkcs12 flash:myexport password mypassword123
Destination filename [myexport]?
Writing pkcs12 file to tftp:/mytftpserver/myexport
CRYPTO_PKI:Exported PKCS12 file successfully.
Verifying checksum... OK (0x3307)
!
July 8 17:30:09 GMT:%CRYPTO-6-PKCS12EXPORT_SUCCESS:PKCS #12 Successfully Exported.
```

Router B

```
crypto pki import mynewtp pkcs12 flash:myexport password mypassword123
Source filename [myexport]?
CRYPTO_PKI:Imported PKCS12 file successfully.
!
July 8 18:07:50 GMT:%CRYPTO-6-PKCS12IMPORT_SUCCESS:PKCS #12 Successfully Imported.
```

Exporting and Importing and RSA Keys in PEM Files Example

The following example shows the generation, exportation, and importation for the RSA key pair "mytp", and verifies its status:

```
! Generate the key pair
!
Router(config)# crypto key generate rsa general-purpose label mytp exportable

The name for the keys will be: mytp
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys ...[OK]
!
! Archive the key pair to a remote location, and use a good password.
!
Router(config)# crypto pki export mytp pem url nvram:mytp 3des password mypassword123

% Key name:mytp
Usage:General Purpose Key
Exporting public key...
Destination filename [mytp.pub]?
Writing file to nvram:mytp.pub
Exporting private key...
Destination filename [mytp.prv]?
Writing file to nvram:mytp.prv
!
! Import the key as a different name.
!
Router(config)# crypto pki import mytp2 pem url nvram:mytp2 password mypassword123

% Importing public key or certificate PEM file...
Source filename [mytp2.pub]?
```

```

Reading file from nvram:mytp2.pub
% Importing private key PEM file...
Source filename [mytp2.prv]?
Reading file from nvram:mytp2.prv% Key pair import succeeded.
!
! After the key has been imported, it is no longer exportable.
!
! Verify the status of the key.
!
Router# show crypto key mypubkey rsa

% Key pair was generated at:18:04:56 GMT Jun 6 2011
Key name:mycs
Usage:General Purpose Key
Key is exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00E65253
9C30C12E 295AB73F B1DF9FAD 86F88192 7D4FA4D2 8BA7FB49 9045BAB9 373A31CB
A6B1B8F4 329F2E7E 8A50997E AADBCFAA 23C29E19 C45F4F05 DBB2FA51 4B7E9F79
A1095115 759D6BC3 5DFB5D7F BCF655BF 6317DB12 A8287795 7D8DC6A3 D31B2486
C9C96D2C 2F70B50D 3B4CDDAE F661041A 445AE11D 002EEF08 F2A627A0 5B020301 0001
% Key pair was generated at:18:17:25 GMT Jun 6 2011
Key name:mycs2
Usage:General Purpose Key
Key is not exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00E65253
9C30C12E 295AB73F B1DF9FAD 86F88192 7D4FA4D2 8BA7FB49 9045BAB9 373A31CB
A6B1B8F4 329F2E7E 8A50997E AADBCFAA 23C29E19 C45F4F05 DBB2FA51 4B7E9F79
A1095115 759D6BC3 5DFB5D7F BCF655BF 6317DB12 A8287795 7D8DC6A3 D31B2486
C9C96D2C 2F70B50D 3B4CDDAE F661041A 445AE11D 002EEF08 F2A627A0 5B020301 0001

```

Exporting Router RSA Key Pairs and Certificates from PEM Files Example

The following example shows how to generate and export the RSA key pair “aaa” and certificates of the router in PEM files that are associated with the trustpoint “mycs.” This example also shows PEM-formatted files, which include PEM boundaries before and after the base64-encoded data, that are used by other SSL and SSH applications.

```

Router(config)# crypto key generate rsa general-keys label aaa exportable

The name for the keys will be:aaa
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose
Keys. Choosing a key modulus greater than 512 may take a few minutes.
!
How many bits in the modulus [512]:
% Generating 512 bit RSA keys ...[OK]
!
Router(config)# crypto pki trustpoint mycs

Router(ca-trustpoint)# enrollment url http://mycs

Router(ca-trustpoint)#
rsakeypair aaa

Router(ca-trustpoint)# exit

Router(config)# crypto pki authenticate mycs

Certificate has the following attributes:
Fingerprint:C21514AC 12815946 09F635ED FBB6CF31
% Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.
!
Router(config)# crypto pki enroll mycs

%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this password to the CA
Administrator in order to revoke your certificate.

```

```

For security reasons your password will not be saved in the configuration.
Please make a note of it.
Password:
Re-enter password:
% The fully-qualified domain name in the certificate will be: Router
% The subject name in the certificate will be:host.company.com
% Include the router serial number in the subject name? [yes/no]: n
% Include an IP address in the subject name? [no]: n
Request certificate from CA? [yes/no]: y
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto ca certificate' command will also show the fingerprint.
Router(config)# Fingerprint:8DA777BC 08477073 A5BE2403 812DD157
00:29:11:%CRYPTO-6-CERTRET:Certificate received from Certificate Authority
Router(config)# crypto ca export aaa pem terminal 3des password

% CA certificate:
-----BEGIN CERTIFICATE-----
MIICAzCCAa2gAwIBAgIBATANBgkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJVUzES
<snip>
waDeNOSI3WlDa0AWq5DkVBkxwgn0TqIJXJOCttjHnWHK1LMcMVGn
-----END CERTIFICATE-----
% Key name:aaa
Usage:General Purpose Key
-----BEGIN RSA PRIVATE KEY-----
Proc-Type:4, ENCRYPTED
DEK-Info:DES-EDE3-CBC, ED6B210B626BC81A
Urguv0jnjwOgowWVUQ2XR5nbzzYHI2vGLunpH/IxIsJuNjRVjbaAUUpGk7VnPCT87
<snip>
kLC0txzEv7JHc72gMku9uUlrLSnFH5slzAtoC0czfU4=
-----END RSA PRIVATE KEY-----
% Certificate:
-----BEGIN CERTIFICATE-----
MIICTjCCAfIgaAwIBAgICIQUwDQYJKoZIhvcNAQEFBQAwTjELMAkGA1UEBhMCVVMx
<snip>
6xlBaIisuMxnHmr89KkKkYlU6
-----END CERTIFICATE-----

```

Importing Router RSA Key Pairs and Certificate from PEM Files Example

The following example shows how to import the RSA key pairs and certificate to the trustpoint “ggg” from PEM files via TFTP:

```

Router(config)# crypto pki import ggg pem url tftp://10.1.1.2/username/msca password

% Importing CA certificate...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.ca]?
Reading file from tftp://10.1.1.2/username/msca.ca
Loading username/msca.ca from 10.1.1.2 (via FastEthernet0):!
[OK - 1082 bytes]
% Importing private key PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.prv]?
Reading file from tftp://10.1.1.2/username/msca.prv
Loading username/msca.prv from 10.1.1.2 (via FastEthernet0):!
[OK - 573 bytes]
% Importing certificate PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.crt]?
Reading file from tftp://10.1.1.2/username/msca.crt
Loading username/msca.crt from 10.1.1.2 (via FastEthernet0):!
[OK - 1289 bytes]
% PEM files import succeeded.
Router(config)#

```

Encrypting and Locking Private Keys on a Router Examples

This section contains the following configuration examples:

Configuring and Verifying an Encrypted Key Example

The following example shows how to encrypt the RSA key “pki-123.company.com.” Thereafter, the **show crypto key mypubkey rsa** command is issued to verify that the RSA key is encrypted (protected) and unlocked.

```
Router(config)# crypto key encrypt rsa name pki-123.company.com passphrase password
Router(config)# exit
Router# show crypto key mypubkey rsa

% Key pair was generated at:00:15:32 GMT Jun 25 2003

Key name:pki-123.company.com
Usage:General Purpose Key
*** The key is protected and UNLOCKED. ***

Key is not exportable.

Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E0CC9A 1D23B52C
CD00910C ABD392AE BA6D0E3F FC47A0EF 8AFEE340 0EC1E62B D40E7DCC
23C4D09E
03018B98 E0C07B42 3CFD1A32 2A3A13C0 1FF919C5 8DE9565F 1F020301 0001

% Key pair was generated at:00:15:33 GMT Jun 25 2003

Key name:pki-123.company.com.server
Usage:Encryption Key
Key is exportable.

Key Data:
307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00D3491E 2A21D383
854D7DA8 58AFBDAC 4E11A7DD E6C40AC6 66473A9F 0C845120 7C0C6EC8 1FFF5757
3A41CE04 FDCB40A4 B9C68B4F BC7D624B 470339A3 DE739D3E F7DDB549 91CD4DA4
DF190D26 7033958C 8A61787B D40D28B8 29BCD0ED 4E6275C0 6D020301 0001

Router#
```

Configuring and Verifying a Locked Key Example

The following example shows how to lock the key “pki-123.company.com.” Thereafter, the **show crypto key mypubkey rsa** command is issued to verify that the key is protected (encrypted) and locked.

```
Router# crypto key lock rsa name pki-123.company.com passphrase password
!
Router# show crypto key mypubkey rsa

% Key pair was generated at:20:29:41 GMT Jun 20 2003
Key name:pki-123.company.com
Usage:General Purpose Key
*** The key is protected and LOCKED. ***
Key is exportable.
Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00D7808D C5FF14AC
```

0D2B55AC 5D199F2F 7CB4B355 C555E07B 6D0DECBE 4519B1F0 75B12D6F 902D6E9F
 B6FDAD8D 654EF851 5701D5D7 EDA047ED 9A2A619D 5639DF18 EB020301 0001

Where to Go Next

After you have generated an RSA key pair, you should set up the trustpoint. If you have already set up the trustpoint, you should authenticate and enroll the routers in a PKI. For information on enrollment, see the module “Configuring Certificate Enrollment for a PKI.”

Additional References

The following sections provide references related to configuring RSA keys for a PKI.

- [Related Documents, page 18](#)
- [Technical Assistance, page 18](#)

Related Documents

Related Topic	Document Title
Overview of PKI, including RSA keys, certificate enrollment, and CAs	Cisco IOS XE PKI Overview: Understanding and Planning a PKI
PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples	<i>Cisco IOS Security Command Reference</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Feature Information for Deploying RSA Keys Within a PKI

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 Feature Information for Deploying RSA Keys Within a PKI

Feature Name	Releases	Feature Information
Cisco IOS XE 4096-Bit Peer Public Key Support	Cisco IOS XE Release 2.4	<p>This feature introduces Cisco IOS XE 4096-bit peer public key support.</p> <p>The following section provides information about this feature:</p> <ul style="list-style-type: none"> • RSA Keys Overview
Exporting and Importing RSA Keys	Cisco IOS XE Release 2.1	<p>This feature allows you to transfer security credentials between devices by exporting and importing RSA keys. The key pair that is shared between two devices will allow one device to immediately and transparently take over the functionality of the other router.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Benefits of Exportable RSA Keys • Exporting and Importing RSA Keys in PKCS12 Files <p>The following commands were introduced or modified by this feature: crypto ca export pkcs12, crypto ca import pkcs12, crypto key generate rsa (IKE)</p>

Feature Name	Releases	Feature Information
Import of RSA Key Pair and Certificates in PEM Format	Cisco IOS XE Release 2.1	<p>This feature allows customers to use PEM-formatted files to import or export RSA key pairs. PEM-formatted files allow customers to directly use existing RSA key pairs on their Cisco IOS XE routers instead of generating new keys.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Benefits of Exportable RSA Keys • Exporting_and_Importing_RSA_Keys_in_PEM-Formatted_Files <p>The following commands were introduced by this feature: crypto ca export pem, crypto ca import pem, crypto key export pem, crypto key import pem</p>
Multiple RSA Key Pair Support	Cisco IOS XE Release 2.1	<p>This feature allows a user to configure a router to have multiple RSA key pairs. Thus, the Cisco IOS XE software can maintain a different key pair for each identity certificate.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Reasons_to_Store_Multiple_RSA_Keys_on_a_Router_1055232. • Generating_and_Storing_Multiple_RSA_Key_Pairs <p>The following commands were introduced or modified by this feature: crypto key generate rsa, crypto key zeroize rsa, rsakeypair</p>

Feature Name	Releases	Feature Information
Protected Private Key Storage	Cisco IOS XE Release 2.1	<p>This feature allows a user to encrypt and lock the RSA private keys that are used on a Cisco IOS XE router, thereby, preventing unauthorized use of the private keys.</p> <p>The following section provides information about this feature:</p> <ul style="list-style-type: none"> • <code>Encrypting_and_Locking_Private_Keys_on_a_Router</code> <p>The following commands were introduced or modified by this feature : crypto key decrypt rsa, crypto key encrypt rsa, crypto key lock rsa, crypto key unlock rsa, show crypto key mypubkey rsa</p>

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

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.

© 2012 Cisco Systems, Inc. All rights reserved.