crypto key generate rsa

- crypto key generate rsa, page 2
crypto key generate rsa

To generate Rivest, Shamir, and Adelman (RSA) key pairs, use the `crypto key generate rsa` command in global configuration mode.

```
crypto key generate rsa [general-keys| usage-keys| signature| encryption] [label key-label] [exportable] [modulus modulus-size] [storage devicename :] [redundancy] [on devicename :]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>general-keys</td>
<td>(Optional) Specifies that a general-purpose key pair will be generated, which is the default.</td>
</tr>
<tr>
<td>usage-keys</td>
<td>(Optional) Specifies that two RSA special-usage key pairs, one encryption pair and one signature pair, will be generated.</td>
</tr>
<tr>
<td>signature</td>
<td>(Optional) Specifies that the RSA public key generated will be a signature special usage key.</td>
</tr>
<tr>
<td>encryption</td>
<td>(Optional) Specifies that the RSA public key generated will be an encryption special usage key.</td>
</tr>
</tbody>
</table>
| label  key-label | (Optional) Specifies the name that is used for an RSA key pair when they are being exported.  
If a key label is not specified, the fully qualified domain name (FQDN) of the router is used. |
| exportable     | (Optional) Specifies that the RSA key pair can be exported to another Cisco device, such as a router. |
| modulus  modulus-size | (Optional) Specifies the IP size of the key modulus.  
By default, the modulus of a certification authority (CA) key is 1024 bits. The recommended modulus for a CA key is 2048 bits. The range of a CA key modulus is from 350 to 4096 bits.  
**Note** Effective with Cisco IOS XE Release 2.4 and Cisco IOS Release 15.1(1)T, the maximum key size was expanded to 4096 bits for private key operations. The maximum for private key operations prior to these releases was 2048 bits. |
| storage  devicename  : | (Optional) Specifies the key storage location. The name of the storage device is followed by a colon (:). |
| redundancy     | (Optional) Specifies that the key should be synchronized to the standby CA. |
on `devicename`:

(Optional) Specifies that the RSA key pair will be created on the specified device, including a Universal Serial Bus (USB) token, local disk, or NVRAM. The name of the device is followed by a colon (:).

Keys created on a USB token must be 2048 bits or less.

**Command Default**

RSA key pairs do not exist.

**Command Modes**

Global configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.3</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>12.2(8)T</td>
<td>The <code>key-label</code> argument was added.</td>
</tr>
<tr>
<td>12.2(15)T</td>
<td>The <code>exportable</code> keyword was added.</td>
</tr>
<tr>
<td>12.2(18)SXD</td>
<td>This command was integrated into Cisco IOS Release 12.2(18)SXD.</td>
</tr>
<tr>
<td>12.4(4)T</td>
<td>The <code>storage</code> keyword and <code>devicename:</code> argument were added.</td>
</tr>
<tr>
<td>12.2(33)SRA</td>
<td>This command was integrated into Cisco IOS Release 12.2(33)SRA.</td>
</tr>
<tr>
<td>12.4(11)T</td>
<td>The <code>storage</code> keyword and <code>devicename:</code> argument were implemented on the Cisco 7200VXR NPE-G2 platform. The <code>signature</code>, <code>encryption</code> and <code>on</code> keywords and <code>devicename:</code> argument were added.</td>
</tr>
<tr>
<td>12.4(24)T</td>
<td>Support for IPv6 Secure Neighbor Discovery (SeND) was added.</td>
</tr>
<tr>
<td>XE 2.4</td>
<td>The maximum RSA key size was expanded from 2048 to 4096 bits for private key operations.</td>
</tr>
<tr>
<td>15.0(1)M</td>
<td>This command was modified. The <code>redundancy</code> keyword was introduced.</td>
</tr>
<tr>
<td>15.1(1)T</td>
<td>This command was modified. The range value for the <code>modulus</code> keyword value is extended from 360 to 2048 bits to 360 to 4096 bits.</td>
</tr>
<tr>
<td>15.2(2)SA2</td>
<td>This command was implemented on the Cisco ME 2600X Series Ethernet Access Switches.</td>
</tr>
</tbody>
</table>
Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Use this command to generate RSA key pairs for your Cisco device (such as a router). RSA keys are generated in pairs—one public RSA key and one private RSA key.

If your router already has RSA keys when you issue this command, you will be warned and prompted to replace the existing keys with new keys.

Before issuing this command, ensure that your router has a hostname and IP domain name configured (with the `hostname` and `ip domain-name` commands). You will be unable to complete the `crypto key generate rsa` command without a hostname and IP domain name. (This situation is not true when you generate only a named key pair.)

Secure Shell (SSH) may generate an additional RSA key pair if you generate a key pair on a router having no RSA keys. The additional key pair is used only by SSH and will have a name such as `{router_FQDN}.server`. For example, if a router name is "router1.cisco.com," the key name is "router1.cisco.com.server."

This command is not saved in the router configuration; however, the RSA keys generated by this command are saved in the private configuration in NVRAM (which is never displayed to the user or backed up to another device) the next time the configuration is written to NVRAM.

If the configuration is not saved to NVRAM, the generated keys are lost on the next reload of the router.

There are two mutually exclusive types of RSA key pairs: special-usage keys and general-purpose keys. When you generate RSA key pairs, you will be prompted to select either special-usage keys or general-purpose keys.

**Special-Usage Keys**

If you generate special-usage keys, two pairs of RSA keys will be generated. One pair will be used with any Internet Key Exchange (IKE) policy that specifies RSA signatures as the authentication method, and the other pair will be used with any IKE policy that specifies RSA encrypted keys as the authentication method.

A CA is used only with IKE policies specifying RSA signatures, not with IKE policies specifying RSA-encrypted nonces. (However, you could specify more than one IKE policy and have RSA signatures specified in one policy and RSA-encrypted nonces in another policy.)

If you plan to have both types of RSA authentication methods in your IKE policies, you may prefer to generate special-usage keys. With special-usage keys, each key is not unnecessarily exposed. (Without special-usage keys, one key is used for both authentication methods, increasing the exposure of that key.)

**General-Purpose Keys**

---

crypto key generate rsa

---

crypto key generate rsa
If you generate general-purpose keys, only one pair of RSA keys will be generated. This pair will be used with IKE policies specifying either RSA signatures or RSA encrypted keys. Therefore, a general-purpose key pair might get used more frequently than a special-usage key pair.

Named Key Pairs

If you generate a named key pair using the `key-label` argument, you must also specify the `usage-keys` keyword or the `general-keys` keyword. Named key pairs allow you to have multiple RSA key pairs, enabling the Cisco IOS software to maintain a different key pair for each identity certificate.

Modulus Length

When you generate RSA keys, you will be prompted to enter a modulus length. The longer the modulus, the stronger the security. However, a longer modulus takes longer to generate (see the table below for sample times) and takes longer to use.

<table>
<thead>
<tr>
<th>Router</th>
<th>360 bits</th>
<th>512 bits</th>
<th>1024 bits</th>
<th>2048 bits (maximum)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco 2500</td>
<td>11 seconds</td>
<td>20 seconds</td>
<td>4 minutes, 38 seconds</td>
<td>More than 1 hour</td>
</tr>
<tr>
<td>Cisco 4700</td>
<td>Less than 1 second</td>
<td>1 second</td>
<td>4 seconds</td>
<td>50 seconds</td>
</tr>
</tbody>
</table>

Cisco IOS software does not support a modulus greater than 4096 bits. A length of less than 512 bits is normally not recommended. In certain situations, the shorter modulus may not function properly with IKE, so we recommend using a minimum modulus of 2048 bits.

As of Cisco IOS Release 12.4(11)T, 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 for a CA is 2048 bits; the recommended modulus for a client is 2048 bits.

Note

Additional limitations may apply when RSA keys are generated by cryptographic hardware. For example, when RSA keys are generated by the Cisco VPN Services Port Adapter (VSPA), the RSA key modulus must be a minimum of 384 bits and must be a multiple of 64.

Specifying a Storage Location for RSA Keys

When you issue the `crypto key generate rsa` command with the `storage devicename` keyword and argument, the RSA keys will be stored on the specified device. This location will supersede any `crypto key storage` command settings.

Specifying a Device for RSA Key Generation

As of Cisco IOS Release 12.4(11)T and later releases, you may specify the device where RSA keys are generated. Devices supported include NVRAM, local disks, and USB tokens. If your router has a USB token configured and available, the USB token can be used as cryptographic device in addition to a storage device. Using a USB token as a cryptographic device allows RSA operations such as key generation, signing, and authentication of credentials to be performed on the token. The private key never leaves the USB token and is not exportable. The public key is exportable.
RSA keys may be generated on a configured and available USB token, by the use of the on devicename : keyword and argument. Keys that reside on a USB token are saved to persistent token storage when they are generated. The number of keys that can be generated on a USB token is limited by the space available. If you attempt to generate keys on a USB token and it is full you will receive the following message:

% Error in generating keys:no available resources

Key deletion will remove the keys stored on the token from persistent storage immediately. (Keys that do not reside on a token are saved to or deleted from nontoken storage locations when the copy or similar command is issued.)


**Specifying RSA Key Redundancy Generation on a Device**

You can specify redundancy for existing keys only if they are exportable.

**Examples**

The following example generates a general-usage 1024-bit RSA key pair on a USB token with the label “ms2” with crypto engine debugging messages shown:

```
Router(config)# crypto key generate rsa label ms2 modulus 2048 on usbtoken0:
The name for the keys will be: ms2
% The key modulus size is 2048 bits
% Generating 1024 bit RSA keys, keys will be on-token, non-exportable...
Jan 7 02:44:10.467: CryptoEngine0: CRYPTO_ISA_RSA_CREATE_PUBKEY(hw)(ipsec)
Jan 7 02:44:10.467: CryptoEngine0: CRYPTO_ISA_RSA_PUB_DECRYPT(hw)(ipsec)
```

Now, the on-token keys labeled “ms2” may be used for enrollment.

The following example generates special-usage RSA keys:

```
Router(config)# crypto key generate rsa usage-keys
The name for the keys will be: myrouter.example.com
Choose the size of the key modulus in the range of 360 to 2048 for your Signature Keys. Choosing a key modulus greater than 512 may take a few minutes. How many bits in the modulus[512]? <return>
Generating RSA keys.... [OK].
Choose the size of the key modulus in the range of 360 to 2048 for your Encryption Keys. Choosing a key modulus greater than 512 may take a few minutes. How many bits in the modulus[512]? <return>
Generating RSA keys.... [OK].
```

The following example generates general-purpose RSA keys:

```
Router(config)# crypto key generate rsa general-keys
The name for the keys will be: myrouter.example.com
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]? <return>
Generating RSA keys.... [OK].
```

**Note**

You cannot generate both special-usage and general-purpose keys; you can generate only one or the other.
The following example generates the general-purpose RSA key pair "exampleCAkeys":

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

The following example specifies the RSA key storage location of "usbtoken0:" for "tokenkey1":

crypto key generate rsa general-keys label tokenkey1 storage usbtoken0:

The following example specifies the redundancy keyword:

Router(config)# crypto key generate rsa label MYKEYS redundancy
The name for the keys will be: MYKEYS

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, keys will be non-exportable with redundancy...[OK]

<table>
<thead>
<tr>
<th>Related Commands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>copy</td>
<td>Copies any file from a source to a destination, use the copy command in privileged EXEC mode.</td>
</tr>
<tr>
<td>crypto key storage</td>
<td>Sets the default storage location for RSA key pairs.</td>
</tr>
<tr>
<td>debug crypto engine</td>
<td>Displays debug messages about crypto engines.</td>
</tr>
<tr>
<td>hostname</td>
<td>Specifies or modifies the hostname for the network server.</td>
</tr>
<tr>
<td>ip domain-name</td>
<td>Defines a default domain name to complete unqualified hostnames (names without a dotted-decimal domain name).</td>
</tr>
<tr>
<td>show crypto key mypubkey rsa</td>
<td>Displays the RSA public keys of your router.</td>
</tr>
<tr>
<td>show crypto pki certificates</td>
<td>Displays information about your PKI certificate, certification authority, and any registration authority certificates.</td>
</tr>
</tbody>
</table>
crypto key generate rsa

crypto key generate rsa