# Provision Third-party Key Packages

Key packages are required for onboarding owner keys, customer consent tokens, and owner RPMs on the router.



## Key packages

A key package is a Cryptographic Message Syntax (CMS [RFC5652]) that must be digitally signed with private keys of the customer's Ownership Certificate (OC). Key packages can be used to

- add one or more keys to the router
- delete one or more keys from the router, and
- revoke one or more keys from the router.

| Feature Name | Release Information | Feature Description |
|---|---|---|
| Key package enhancements | Release 26.1.1 | This feature introduces you to the version 3 key package. With the version 3 key package, you can create, validate, sign a key package before the key package is provisioned on the router.<br><br>Unlike the reserved customer consent token name, CUS-CT, used in version 1 and version 2 key packages, you can now use any name for the customer consent token that you include in the key package.<br><br>However, to enable the customer consent token workflow, execute the **consent-token customer** command after the key package installation. |

A key package provides a secure mechanism to install owner or third-party public keys (GPG or X.509) on the router.

### Key package versions and IOS XR releases

You can create key packages using the GitHub script for different key package versions. These key packages can be installed on the router based on the supported IOS XR releases.

***Table 1: Key package versions and IOS XR releases***

| Key package version | IOS XR release | Purpose |
|---|---|---|
| key-pkg-ver1 | Cisco IOS XR Release 7.6.x, Cisco IOS XR Release 7.7.x | With `key-pkg-ver1`, you can perform the following operations:<br>• Create a single key package with signed keys.<br>• Remove any existing key package.<br>• Verify a key package.<br><br>The key request in the key package supports ADD, DELETE, or REVOKE action. |

| Key package version | IOS XR release | Purpose |
|---|---|---|
| key-pkg-ver2 | Cisco IOS XR Release 7.8.x – Cisco IOS XR Release 25.4.x | With `key-pkg-ver2`, you can perform the following operations:<br>• Create a super key package with two or more key packages or a single key package containing signed keys,<br>• Remove any existing key package or super key package.<br>• Verify a key package or super key package.<br>The key request in the key package supports ADD, DELETE, or REVOKE action. |
| key-pkg-ver3 | Cisco IOS XR Release 26.1.1 and later | With `key-pkg-ver3`, you can perform the following operations:<br>• Create a single key package JSON file containing signed keys.<br>• Validate the key package.<br>• Sign the key package.<br>Key requests do not include action (add, delete, revoke). |

# Prerequisites for key packages

Ensure that the following prerequisites are met before installing key packages on the router:

- Device ownership should be established.
- The router must be up and running.
- Any uncommitted install operations should not be in progress.

# Restrictions for key packages

- Key packages to add, remove, revoke or unrevoke customer or Cisco keys must have a timestamp.
- Key packages must be processed based on the timestamp order. Key packages with a timestamp earlier than that of a processed key package are rejected.
- A key package to remove a key fails if there are any installed packages signed with this key.

# Install key packages on the router

This procedure allows you to add or install a customer or third-party key package on the router.

Key package versions 1 and 2 support ADD, DELETE, and REVOKE actions in key requests.

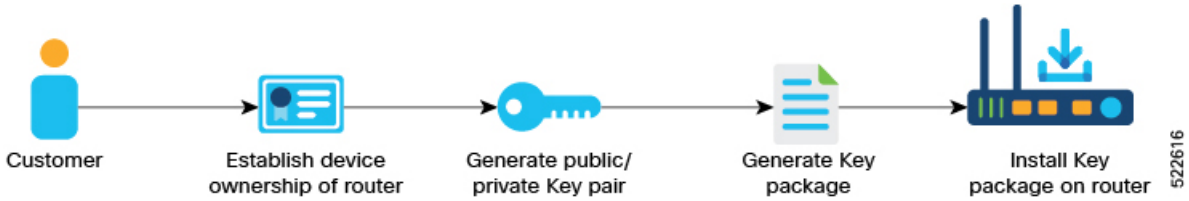
Starting from Cisco IOS XR Release 26.1.1, the ADD, DELETE, and REVOKE actions should not be included anymore in key requests to install, remove, and revoke keys respectively. Instead, if you want to add a new key, include the new key with other installed keys in the new key request. Similarly, to remove an installed key, exclude this key and include other installed keys in the new key request.

For example, if keys K1 and K2 are installed, and if you want to install key K3 on the router, include keys K1, K2, and K3 in the new key request. If you want to remove key K2 from a list of keys K1, K2, and K3, include keys K1 and K3 only in the new key request.

You can also use your keys to sign the consent token response.

To unrevoke a key by removing a key from the list of revoked keys using the **clear device-ownership** command. For more information, see Clear device ownership.

***Figure 1: Workflow for installing a key package on the router***




### Before you begin

Ensure that your device ownership is established.

## Procedure

**Step 1** On a Linux machine, use the standard openssl commands to generate an RSA/GPG key-pair.

**Note**
Generate your own public-private key-pair (typically this key pair is a GPG key, but it could also be an X509 certificate).

**Step 2** Generate version 1, version 2, or version 3 key package using the script at https://github.com/ios-xr/key-package-scripts.

**Step 3** Install the key package on the router.

Copy the key package to router and execute the **platform security key-package customer install** *key-package-file* command to install the key package.

```
RP/0/RP0/CPU0:router# platform security key-package customer install
disk0:/testing2/key-pkg/key_add.kpkg
Mon Jun 14 16:09:28.238 UTC

Key package successfully validated
```

```
Config file successfully parsed.
Successfully added key cust-ct.der to TPM
Successfully processed all keys.
RP/0/RP0/CPU0:router#
```

**Step 4** Verify that the key package is installed.

```
RP/0/RP0/CPU0:router# show platform security key-package customer allowed-list location
0/RP0/CPU0

Mon Jun 14 16:10:01.440 UTC
-----------------------------------------------
Node - node0_RP0_CPU0
-----------------------------------------------

Key Name: D3CUS-CT1
Key:
MIIC7TCCAdUCAQIwDQYJKoZIhvcNAQELBQAwOzELMAkGA1UEBhMCVVMxDDAKBgNV
BAoMA3h6eTEMMAoGA1UECwwDYWJjMRAwDgYDVQQDDAdST09ULUNOMB4XDTIxMDYx
NDE1MjkwOVoXDTI0MDMxMDE1MjkwOVowPjELMAkGA1UEBhMCVVMxDDAKBgNVBAoM
A3h5ejEMMAoGA1UECwwDYWJjMRMwEQYDVQQDDApDVVNULUNULUNOMIIBIjANBgkq
hkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAyOT2SGTuJcQlAHCsQn4gcoZGK+po1A6g
LPV5AzOBcY0pfXV5eXoxf6S8qbmQP4l4v5MjsHzFTOuouMmijpGYFJv7TORwJ2Xw
weJ5aKbqsYTQlSQSUZ1XxG7AOdHMshVRzy7vIA7LLQJnD0j1F1U2FoRi5NhhY12L
wmYA4aPj1o+LoubAfjF1BVl3vE8rfI0mzsXODJIks+oeJbsq4HmyMbOAzLVdeucp
7bu3S8kDlc1ph4zqm81BkDZgV1++2CoCBWROt9dRZrp+ENw1GEHcXgS659iZpUmj
juG1n0W3Y6br8SE+EqqhMqkAfSbO8vaG02qYtTUNJ5gkMcTljCfDAQIDAQABMA0G
CSqGSIb3DQEBCwUAA4IBAQCDeJ5ov2gG3rj5ttpfibxiakpzl706W9crjIePJka6
CWS7Y3nxt02+PGsBByEcBPV7aU8oH2GfKN4jNZHDChfzGN7rtfRE2CG+ttvTxJLC
Ba+LjzKFSveKgPRG/gAAkZY0hRmTe7FkgmKB4UCi+u0XP3U5VlT5XRP3LGVoX0fC
rY4/GBKkG5eOF+VGD4iyPfOHjrwduO/K2DqDXyUfa1PXZDzatpnin07ShkCJQoT+
u6C1SotJ8mtrFJpePDUsa5W3O2oPROFHd4sGCivt40AbpaWECK+KLpKC+DoqN+46
tMV79rpQ0mtXo/XfY4UGir4weH9g/e2fct4g+Y2E/BD+


Key Name: D3CUS-CTX
Key:
PNM:APNAM,KNM:AKNAM,
RP/0/RP0/CPU0:router#
```

**Note**
Key packages can be also included in a GISO and installed on the router from the GISO. For more information, see Install signed owner RPMs using GISO.

# Handling key rotation

## Summary

Key rotation is a process that allows you to install new keys on the router and delete installed keys from the router using a single key package or mutiple key packages based on the key package version. `key-pkg-ver-1` and `key-pkg-ver-2` allow you to add and remove keys using multiple key packages. If you use `key-pkg-ver-3`, you can use a single key package that include the keys to be installed on the router and keys to be removed from the router.

### Workflow

These are the stages in the key rotation process:

1. If you are using `key-pkg-ver-1` and `key-pkg-ver-2`, create two key packages, one to remove an existing key, and the other to install a new key.

   If you are using `key-pkg-ver-3`, create a single key package that includes new keys to be installed on the router and omits existing keys that should be removed from the router.

2. Install the key package or key packages created in stage 1 using the **platform security key-package customer install** <*pkg-name*>.*kpkg* command.