

Configuring Keychain Management

This chapter describes how to configure keychain management on a Cisco NX-OS device.

This chapter includes the following sections:

- About keychain management, on page 1
- Prerequisites for keychain management, on page 2
- Guidelines and limitations for keychain management, on page 2
- Default settings for keychain management, on page 3
- Configuring keychain management, on page 3
- Determining active key lifetimes, on page 11
- Keychain management configuration verification commands, on page 11
- Configuration example for keychain management, on page 11
- Where to go next, on page 12
- Additional references for keychain management, on page 12

About keychain management

Keychain management allows you to create and maintain keychains, which are sequences of keys (sometimes called shared secrets). You can use keychains with features that secure communications with other devices by using key-based authentication. The device allows you to configure multiple keychains.

Some routing protocols that support key-based authentication can use a keychain to implement a hitless key rollover for authentication. For more information, see the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*.

Lifetime of a Key

To maintain stable communications, each device that uses a protocol that is secured by key-based authentication must be able to store and use more than one key for a feature at the same time. Based on the send and accept lifetimes of a key, keychain management provides a secure mechanism to handle key rollover. The device uses the lifetimes of keys to determine which keys in a keychain are active.

Each key in a keychain has two lifetimes, as follows:

- Accept lifetime

  The time interval within which the device accepts the key during a key exchange with another device.

- Send lifetime

The time interval within which the device sends the key during a key exchange with another device.

You define the send and accept lifetimes of a key using the following parameters:

- Start-time

  The absolute time that the lifetime begins.

- End-time

  The end time can be defined in one of the following ways:

  - The absolute time that the lifetime ends
  - The number of seconds after the start time that the lifetime ends
  - Infinite lifetime (no end-time)

During a key send lifetime, the device sends routing update packets with the key. The device does not accept communication from other devices when the key sent is not within the accept lifetime of the key on the device.

We recommend that you configure key lifetimes that overlap within every keychain. This practice avoids failure of neighbor authentication due to the absence of active keys.

Prerequisites for keychain management

Keychain management has no prerequisites.

Guidelines and limitations for keychain management

Use this reference to review the guidelines and limitations for keychain management.

Keychain management has the following configuration guidelines and limitations:

- Changing the system clock impacts when the keys are active.
- A keychain's configuration type must match the type it has been linked to within the client protocol. If an attempt is made to mismatch these types, a syslog message is generated to notify the user.

  For example: It is not supported if a keychain named **keychain_abc** is configured as a Macsec keychain but is associated as a Classic keychain with OSPF. Similarly, the case where the keychain is first associated with the client (a process known as forward-referencing) and then configured as a different keychain type, is also not supported.

- It is highly recommended for user to specify the passwordtype and password when programmatically(restconf/Netconf and so on) configuring a neighbor/template’s password. When either one of the property is missing in the programmatic call, BGP will use already available(or default) value of the missing property to configure the neighbor/template’s password.

  If the user has to configure with a property missing then the user has to follow the same sequence of steps in both peer routers.

Default settings for keychain management

Use this reference to review the default settings for keychain management.

Table 1: Default keychain management parameters

Parameters	Default
Key chains	No keychain exists by default.
Keys	No keys are created by default when you create a new keychain.
Accept lifetime	Always valid.
Send lifetime	Always valid.
Key-string entry encryption	Unencrypted.

Configuring keychain management

Explains configuring keychain management.

Create a keychain

You can create a keychain on the device. A new keychain contains no keys.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **key chain** *name*

Example:

```
switch(config)#key chain bgp-keys
switch(config-keychain)#
```

Creates the keychain and enters keychain configuration mode.

Step 3 (Optional) **show key chain** *name*

Example:

```
switch#show key chain bgp-keys
```

Displays the keychain configuration.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Remove a keychain

You can remove a keychain on the device.



Note Removing a keychain removes any keys within the keychain.

Before you begin

If you are removing a keychain, ensure that no feature uses it. If a feature is configured to use a keychain that you remove, that feature is likely to fail to communicate with other devices.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **no key chain** *name*

Example:

```
switch(config)#no key chain bgp-keys
```

Removes the keychain and any keys that the keychain contains.

Step 3 (Optional) **show key chain** *name*

Example:

```
switch#show key chain bgp-keys
```

Confirms that the keychain no longer exists in running configuration.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure a primary key and enable the AES password encryption feature

Secure device authentication by using AES password encryption (Type-6).

You can configure a primary key for Type-6 encryption and enable the Advanced Encryption Standard (AES) password encryption feature.

Beginning with Cisco NX-OS Release 10.3(3)F, Type-6 encryption is supported for RPM legacy keychain.

Follow these steps to configure a primary key and enable the AES password encryption feature:

Procedure

Step 1 Configures a primary key (`Master Key`) to be used with the AES password encryption feature using the [**no**] **key config-key ascii** [<new_key> old <old_master_key>] command.

Example:

```
switch#key config-key ascii
New Master Key:
Retype Master Key:

{"actionLSubj": {"attributes": {"dn": "sys/action/lsubj-[sys/passwdenc]"}
"children": [{"smartcardPasswdEncryptMasterKeyConfigLTask": {"attributes": {"adminSt":
"start",
"dn": "sys/action/lsubj-[sys/passwdenc]/smartcardPasswdEncryptMasterKeyConfigLTask",
"key": "ciscociscociscocisco",
"oldKey": "test1test1test1test1",
"delete": "no",
"freq": "one-shot"}}}]}}
```

The primary key can contain between 16 and 32 alphanumeric characters. You can use the **no** form of this command to delete the primary key at any time.

If you enable the AES password encryption feature before configuring a primary key, a message appears stating that password encryption will not take place unless a primary key is configured. If a primary key is already configured, you are prompted to enter the current primary key before entering a new primary key.

Note
Starting with Cisco NX-OS Release 10.3(2)F, you can configure primary key using DME payload and non-interactive mode.

Step 2 Enter the global configuration mode. **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Step 3 Enable or disable the AES password encryption feature using the [**no**] **feature password encryption aes** command.

Example:

```
switch(config)#feature password encryption aes
```

.

Step 4 Convert existing plain or weakly encrypted passwords to Type-6 encrypted passwords using the **encryption re-encrypt obfuscated** command.

Example:

```
switch(config)#encryption re-encrypt obfuscated
```

Step 5 (Optional) Display the configuration status of the AES password encryption feature and the primary key using the **show encryption service stat** command.

Example:

```
switch#show encryption service stat
```

Step 6 Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Note
This command is necessary to synchronize the primary key in the running configuration and the startup configuration.

Related Topics

AES password encryption and primary encryption keys
AES password encryption and primary encryption keys
Configure text for a key, on page 6
Configure accept and send lifetimes for a key, on page 8

Configure text for a key

You can configure the text for a key. The text is the shared secret. The device stores the text in a secure format.

For MACsec and RPM legacy keychain, the text is encrypted and stored in Type-6 format if AES password encryption feature is enabled and primary key configured otherwise it will be stored in Type-7 encrypted format.

By default, accept and send lifetimes for a key are infinite, which means that the key is always valid. After you configure the text for a key, configure the accept and send lifetimes for the key.

Before you begin

Determine the text for the key. You can enter the text as unencrypted text or in the encrypted form that Cisco NX-OS uses to display key text when you use the **show key chain** command. Using the encrypted form is particularly helpful if you are creating key text to match a key as shown in the **show key chain** command output from another device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **key chain** *name*

Example:

```
switch(config)#key chain bgp-keys
switch(config-keychain)#
```

Enters keychain configuration mode for the keychain that you specified.

Step 3 **key** *key-ID*

Example:

```
switch(config-keychain)#key 13
switch(config-keychain-key)#
```

Enters key configuration mode for the key that you specified. The *key-ID* argument must be a whole number between 0 and 65535.

Step 4 **key-string** [*encryption-type*] *text-string*

Example:

```
switch(config-keychain-key)#key-string 0 AS3cureStr1ng
```

Configures the text string for the key. The *text-string* argument is alphanumeric, case-sensitive, and supports special characters.

The *encryption-type* argument can be one of the following values:

- 0—The *text-string* argument that you enter is unencrypted text. This is the default.
- 6—Beginning with Cisco NX-OS Release 10.3(3)F, the Cisco proprietary (Type-6 encrypted) method is supported on Cisco Nexus 9000 Series platform switches.
- 7—The *text-string* argument that you enter is encrypted. The encryption method is a Cisco proprietary method. This option is useful when you are entering a text string based on the encrypted output of a **show key chain** command that you ran on another Cisco NX-OS device.

The **key-string** command has limitations on using the following special characters in the *text-string*:

Table 2: Configure text for a key

Special Character	Description	Comments
|	Vertical bar or pipe	Unsupported
>	Greater than	Unsupported
\	Backslash	Unsupported
(	Left parenthesis	Unsupported
'	Apostrophe	Unsupported
"	Quotation mark	Unsupported
?	Question mark	Supported. I a question m

For more information on the special characters usage in commands, see Understanding the Command-Line Interface section.

Step 5 (Optional) **show key chain** *name* [**mode decrypt**]

Example:

```
switch#show key chain bgp-keys
```

Shows the keychain configuration, including the key text configuration. The mode decrypt option, which can be used by a device administrator only, displays the keys in cleartext.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Configure a primary key and enable the AES password encryption feature

Configure accept and send lifetimes for a key

You can configure the accept lifetime and send lifetime for a key. By default, accept and send lifetimes for a key are infinite, which means that the key is always valid.

Note We recommend that you configure the keys in a keychain to have overlapping lifetimes. This practice prevents loss of key-secured communication due to moments where no key is active.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **key chain** *name*

Example:

```
switch(config)#key chain bgp-keys
switch(config-keychain)#
```

Enters keychain configuration mode for the keychain that you specified.

Step 3 **key** *key-ID*

Example:

```
switch(config-keychain)#key 13
switch(config-keychain-key)#
```

Enters key configuration mode for the key that you specified.

Step 4 **accept-lifetime** [**local**] *start-time* [**duration** *duration-value* | **infinite** | *end-time*]

Example:

```
switch(config-keychain-key)#accept-lifetime 00:00:00 Jun 13 2013 23:59:59 Sep 12 2013
```

Configures an accept lifetime for the key. By default, the device treats the *start-time* and *end-time* arguments as UTC. If you specify the **local** keyword, the device treats these times as local times.

The *start-time* argument is the time of day and date that the key becomes active.

Specify the end of the lifetime with one of the following options:

- **duration** *duration-value* —The length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years).
- **infinite**—The accept lifetime of the key never expires.
- *end-time* —The *end-time* argument is the time of day and date that the key becomes inactive.

Step 5 **send-lifetime** [**local**] *start-time* [**duration** *duration-value* | **infinite** | *end-time*]

Example:

```
switch(config-keychain-key)#send-lifetime 00:00:00 Jun 13 2013 23:59:59 Aug 12 2013
```

Configures a send lifetime for the key. By default, the device treats the *start-time* and *end-time* arguments as UTC. If you specify the **local** keyword, the device treats these times as local times.

The *start-time* argument is the time of day and date that the key becomes active.

You can specify the end of the send lifetime with one of the following options:

- **duration** *duration-value* —The length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years).
- **infinite**—The send lifetime of the key never expires.
- *end-time* —The *end-time* argument is the time of day and date that the key becomes inactive.

Step 6 (Optional) **show key chain** *name* [**mode decrypt**]

Example:

```
switch#show key chain bgp-keys
```

Shows the keychain configuration, including the key text configuration. The mode decrypt option, which can be used by a device administrator only, displays the keys in cleartext.

Step 7 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Configure a primary key and enable the AES password encryption feature

Configure a key for OSPFv2 cryptographic authentication

You can configure message digest 5 (MD5) or hash-based message authentication code secure hash algorithm (HMAC-SHA) authentication for OSPFv2.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **key chain** *name*

Example:

```
switch(config)#key chain bgp-keys
switch(config-keychain)#
```

Enters keychain configuration mode for the keychain that you specified.

Step 3 **key** *key-ID*

Example:

```
switch(config-keychain)#key 13
switch(config-keychain-key)#
```

Enters key configuration mode for the key that you specified. The *key-ID* argument must be a whole number between 0 and 65535.

Note
For OSPFv2, the key identifier in the key key-id command supports values from 0 to 255 only.

Step 4 [**no**] **cryptographic-algorithm** {**HMAC-SHA-1** | **HMAC-SHA-256** | **HMAC-SHA-384** | **HMAC-SHA-512** | **MD5**}

Example:

```
switch(config-keychain-key)#cryptographic-algorithm HMAC-SHA-1
```

Configures the OSPFv2 cryptographic algorithm to be used for the specified key. You can configure only one cryptographic algorithm per key.

Step 5 (Optional) **show key chain** *name*

Example:

```
switch#show key chain bgp-keys
```

Shows the keychain configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Determining active key lifetimes

Use this reference to determine which keys within a key chain have active accept or send lifetimes.

Table 3: Active key lifetime command

Command	Purpose
show key chain	Displays the key chains configured on the device.

Keychain management configuration verification commands

Use this reference to verify the Keychain management configuration.

To verify keychain management configuration information, use the following command:

Table 4: Keychain management show commands

Command	Purpose
show key chain *name*	Displays the keychains configured on the device.

Configuration example for keychain management

This example shows how to configure a keychain named "ospf-keys". Each key text string is encrypted. The keys are configured to use MD5 as their cryptographic algorithm. Each key has longer accept lifetimes than send lifetimes, resulting in overlap between a pair of keys. In this example, there is configured overlap between key 1 and key 2, as well as key 2 and key 3. This prevents a period of time in which there are no active keys, helping to avoid a disruption in communication of the underlying protocol:

```
key chain ospf-keys
  key 1
    key-string 7 070c285f4d0658544541
    accept-lifetime local 00:00:00 May 13 2024  12:00:00 Sep 14 2024
    send-lifetime local 00:00:00 May 13 2024  00:00:00 Sep 14 2024
    cryptographic-algorithm MD5
  key 2
    key-string 7 070c285f4d0658574446
    accept-lifetime local 00:00:00 Sep 13 2024  12:00:00 Jan 15 2025
    send-lifetime local 10:00:00 Sep 13 2024  12:00:00 Jan 15 2025
    cryptographic-algorithm MD5
  key 3
    key-string 7 070c285fad0622474941
    accept-lifetime local 00:00:00 Jan 15 2025  12:00:00 Jun 15 2025
```

```
    send-lifetime local 10:00:00 Jan 15 2025  12:00:00 Jun 15 2025
    cryptographic-algorithm MD5
```

This example shows how to configure a keychain named "bgp-keys" with Type-6 encryption. This encryption mode is available when feature password encryption aes is enabled:

Configuration example for keychain management command examples

```
key chain bgp-keys
  key 1
    key-string 6
JDYkbN6ZTz3Hqrv5ZWliyxqlYiQXYc0wWpOnK7epMGoHK6qVJPeJtSYAGhQ9V+QKG4ZrcWeuunTtAA==
    accept-lifetime local 00:00:00 May 13 2024  12:00:00 Sep 14 2024
    send-lifetime local 00:00:00 May 13 2024  00:00:00 Sep 14 2024
  key 2
    key-string 6
JDYkO6Di45Bu1ikPja/r8VJNoSTa4I4QMxtzzG3DQzal9G9LJA6F1WNGX8GRgn95SPuf4naoTZCtAA==
    accept-lifetime local 00:00:00 Sep 13 2024  12:00:00 Jan 15 2025
    send-lifetime local 10:00:00 Sep 13 2024  12:00:00 Jan 15 2025
  key 3
    key-string 6
JDYk8DJ15ZdOQ/O7vnj2M92lRiR2x8VrL0Muj/30TNlIK5f+JMFEHoWy0Rfuy827G/H10w2it7eVAA==
    accept-lifetime local 00:00:00 Jan 15 2025  12:00:00 Jun 15 2025
    send-lifetime local 10:00:00 Jan 15 2025  12:00:00 Jun 15 2025
```

Where to go next

For information about routing features that use keychains, see the

Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.

Additional references for keychain management

Use this reference to find additional references for keychain management.

Related Documents

Table 5: Related documents

Related Topic	Document Title
Border Gateway Protocol	*Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*
OSPFv2	*Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*

Standards

Table 6: Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—